User's Manual
Installation and Operation Guidelines

TeleBoss™ 850 Pollable Remote Access Unit

Version 2.05.980_STD

Asentria Corporation
1200 North 96th Street
Seattle, Washington, 98103
U.S.A.
Tel: 206.344.8800
Fax: 206.344.2116
www.asentria.com

TeleBoss™ 850 Pollable Remote Access Unit
Installation and Operation Guidelines

For firmware version 2.05.980 _STD
Manual Release Date: May 17, 2010
Manual Revision: A

Changes In This Version of the User Manual

- Added a section about accessing the unit and configuring a network connection using the OmniDiscover program.
- Changes all references to DIP Switch positions from ON/OFF to UP/DOWN to avoid possible confusion.
- Added the **`net.ftp.autodel`** Setting Key to FTP Settings to enable the FTP Auto-delete function.
- Added new options to the Network Settings/VPN Settings menu.
- Added an SSH to Telnet Bridging option to the CPE Settings menu.
- Added a Setting Key so modem handshaking can be manually set to one of three settings.
- Added the File Access Pointer option to the User Profile menu.
- Added the option in the User Profiles/Set Pass-through Pointer To menu to configure a user to pass-through to a CPE device when connected via SSH.
- Added support for EventSensor Reporting.
- Added Event Message Settings sub-menu to Alarm/Event Definitions menu so a user can customize the message sent in various event alarm actions.
- Moved Callout Attempts, Callout Delay, Action Schedule, and Reminder Interval from Action Definitions menu. to the Action Settings sub-menu.
- Added Joinable Pass-through as an option in the General Settings menu.
- 'Store Relay Activity' option in the Audit Log Settings menu is changed to 'Store Output Activity'.
- Added a table of Status Keys – a quick way to view read-only settings on the unit.
- Added a section concerning SNMP to the Securing a TeleBoss 850 chapter.
- Revised and simplified the section describing SSH and SFTP.
- Added a section explaining the SSH to Telnet Bridging feature.
- Further defined the function of the Button Unlock feature.
- Added a note about the NetPoll Feature and how it may affect security if enabled.
- Added a section about 'How to Secure Telnet' to Telnet/TCP Connections chapter.
- Added a section about VPN's including VPN On-Demand.
- Added support for a Serial Break on a Pass-through connection.
- Revised the Default Router section to explain the DYNAMIC routing feature.
- Added 'Stop if any/all actions OK' and 'Continue' actions to the Action List menu.
- Added a note about how to Clear Event Actions that are still in the event queue.
- Added SMS Messaging as a type of alarm notice now supported.
- Revised section on Analog Voltage Sensor setup to include both Voltage and Current Sensors.
- Added a section describing Type2 EventSensors
- Added a section describing the difference between type of Relay cards available in the T850.
- Added a section to describe ES-T and ES-TH Type II SensorJack Sensor calibration technique.
- Added a section describing how to upload/download/delete scripts using Setting Key commands.
- Added support for a number of new OmniLua Scripting Functions.
- Added the **LOGOFF** command to the User Interface Commands table.
- Added the **DOALARM** and **DOSMS** commands to the System Commands table.
- Added a table for Data Release Commands in the Command Reference chapter.
- Added a section displaying the output of several Usage Commands.
- Added notes to the Wireless Modem/Installation section regarding the correct screws to remove when uninstalling the modem card, and an antenna recommendation.

Conventions used in this manual

- Commands are printed in this format: **COMMANDS** (Arial font, caps, bold, black) although commands used in the unit are not case-sensitive.
- Setting Keys are printed in this format: **`setting.key`** (Courier New font, bold, blue) but any key values displayed are in normal type.
- Red type indicates a safety or security warning.
- Hyperlinks to other sections in the manual are displayed in Arial font, blue, underline.
- Screen shots of menus are all taken from the command line interface. Web interface shots are not displayed in the manual at this time.
- Some settings can only be changed with a Setting Key (no command line menu or web interface options). These are noted throughout Setup Menu section of the manual by **Setting Key:** **`<name of key>`** with a description of the key and allowable values.

Table of Contents

Quick Start

What's Included

This chapter is a brief guide to help get your TeleBoss 850 (T850) up and running quickly.

Hardware Needed

- Asentria TeleBoss 850
- 15VDC power adaptor (Included if AC power option)
- DC power source (if DC power option)
- Computer with serial port and terminal emulation software, and/or network access.
- Ethernet cable
- RJ45 M-M unshielded serial cable and RJ45/DB9 straight thru adapter (Included)
- A PC running any type of SNMP trap management software, if T850 will be sending SNMP traps as event actions.

Information Needed

- IP address(es) to assign to the T850
- Subnet mask
- Default router IP or gateway router IP address if on a WAN (Optional)
- IP address of a PC running any type of SNMP trap management software, if T850 will be sending SNMP traps as event actions.

Connecting

Cables and Power

1. Connect the RJ45 cable (with optional adapter, if needed) to the serial port I/O2 of the T850 and the COM1 of a PC or laptop running any terminal emulator.
2. Connect the attached ground wire securely to an appropriate earth ground (this is essential).
3. Connect an Ethernet cable, if available, into the RJ-45 jack labeled ETH1.
4. Connect the power supply to the unit (see Power Requirements section).

Power Requirements

The T850 is configured with one of two types of power connectors: AC or DC.

If configured for AC, the unit uses a barrel connector for connecting to the 15VDC power adapter shipped with the unit.

If configured for DC, the unit is configured with a 4-pin Molex connector for use with a DC power source. The unit is shipped with the cables and instructions for direct connection to a DC power source. The instructions are shown below, in case they are missing from the box.

» **Note:** This instruction sheet describes connection of the provided –48V wiring harness kit to the source power supply. This unit should be assembled and installed by a qualified technician who can ensure the power source is an isolated, SELV (Safety Extra Low Voltage) circuit. There are two versions of the harness using different wiring colors as shown below.

» **Note:** Because the T850 is generally considered to be "permanently connected“, safety standards require that an appropriate disconnect device shall be provided as part of the building installation. The -48VDC input should be protected by an external 2A Slow Blow Fuse conforming to CSA/UL 248-14, IEC 60127-4/2, at the power supply or within the building circuitry as appropriate. The input DC power current limiting fuse circuit is provided for by the end user, and is required for unit operation in compliance with safety agency approvals.

One example of a compliant fuse for the -48V input is a Littelfuse 239P series, 2 amp fuse with a 250 VDC minimum voltage rating and interrupt rating 10,000 amps at 125 VAC, 0.7 to 0.8 power factor and 100 amps at 125VAC, 0.7-0.8 power factor.

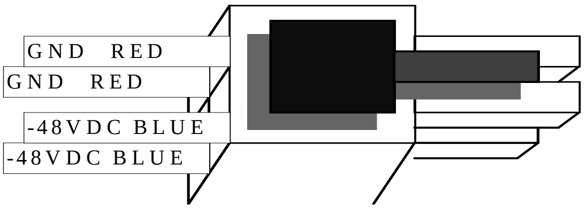
CONTENTS:

Please inventory the package contents and ensure you have the following items pertaining to the -48VDC Power Option:

1. A cable harness consisting of 2 red and 2 blue wires connected to a white nylon “molex” connector.
2. A bare white nylon housing.
3. 5 crimp-on contacts.

-48VDC CONNECTION:

The -48VDC power supply option has 4 input connections. This gives the user the ability to connect this unit to an auxiliary -48VDC power source. Note: The dark area on the diagram represents the latching mechanism on the housing.



DANGER! FIRE HAZARD!
DO NOT LEAVE AN UNCONNECTED WIRE EXPOSED!
DO NOT CONNECT THE UNIT TO ANY OTHER EQUIPMENT UNTIL YOU KNOW THE UNIT POWERS UP CORRECTLY!

Option A: Connect the supplied harness assembly to your -48VDC voltage source:

1. Ensure the unit is not connected to any peripheral equipment.
   » **NOTE:** Peripheral Equipment connections may cause a short circuit of your -48V supply if the power connections are reversed! Do not connect peripheral equipment connections until you know the unit is operational by observing the front panel Power LED.
2. Strip the ends of the wires.
3. Using wire nuts (not supplied), connect the stripped wires to the power source. The red wires connect to ground or the most Positive connection on the voltage source. The blue wires connect to -48VDC or the most Negative connection on the voltage source.

Option B: Use the supplied kit to make a wire harness:

1. You will need a crimping tool that crimps standard Molex type 18-24 AWG Mini-Fit Terminals (Molex Part Number: 39-00-0060, Engineering Series 5556).
2. Crimp the supplied terminals to your cable connections.
3. Insert the crimped terminals into the supplied white nylon housing. Orient the housing so the latching mechanism is up and you are looking into the large end of the housing. See diagram above. Insert the 2 Ground or Most Positive leads into the upper and lower compartments on the left side of the connector, e.g. the same positions as the red wires on the supplied harness assembly. Insert the 2 -48VDC or Most Negative leads into the upper and lower compartments on the right side of the connector, e.g. the same positions as the blue leads on the supplied harness assembly.
4. Connect the completed assembly into the power input receptacle at the rear of the unit.

Accessing the Command Line via a Serial Connection

1. Connect to I/O 2 with a serial terminal emulation program at 19200 baud, 8N1.
2. Enter **STATUS** or **?** and press <Enter>. You will be presented with a status screen similar to the following.

```
TeleBoss 850 2.05.980 STD  -  Status
Site Name       : 850-850000163
Serial Number   : 850000163        Eth 1    : STATIC
Date            : THU 04/29/2010   IP Addr  : 0.0.0.0
Time            : 08:02:48         MAC Addr : 00:10:A3:60:02:3E
Memory          : 32768K           Eth 2    : STATIC
% Full Alarm    : OFF              IP Addr  : 0.0.0.0
No-Data Alarm 1 : OFF              MAC Addr : 00:10:A3:60:02:3F
No-Data Alarm 2 : OFF              Modem    : Yes
Duplex          : FULL
------------------------------------------------------------------------
Port   Baud/Etc.  Recs     Bytes    Full Wrap Name
IO1  : 19200,8N1  00000000 00000000   0% OFF  I/O 1
IO2  : 19200,8N1  00000000 00000000   0% OFF  I/O 2

COMPLETE
>
```

When the status screen appears, the unit is successfully connected and ready for use.

Accessing the Command Line via the Asentria OmniDiscover program

1. From the Asentria website (http://www.asentria.com/docsandsoftware/productManuals.aspx), or the Documentation and Utilities CD, download the OmniDiscover program. This program will allow you to locate devices on your network (ie: the T850) with Asentria MAC addresses, and allow you to assign the network settings directly over the network, thus eliminating the need for the serial port connection as described above.

2. Open the OmniDiscover program. It will immediately display all Asentria devices on the network. Right clicking on the line for this unit displays three options: Setup, Telnet and Web.

   Setup opens another window where the IP Address, Subnet Mask, and Gateway (router) can be configured (see below). Press "OK" and these will be assigned to the unit and displayed in the previous window. (Select this option to configure the network settings for the first time.)

   Telnet opens a connection to the device using your default Telnet client.

   Web opens an HTTP connection to the device using your default browser, if the device supports and is configured to allow a web connection.

3. Once the network settings have been assigned, the T850 command line can be accessed via any Telnet client or HTTP web connection.

Contact Asentria Technical Support for any questions or assistance with OmniDiscover.

Network Setup

via OmniDiscover connection:

1. See the description of how to use OmniDiscover as described above.

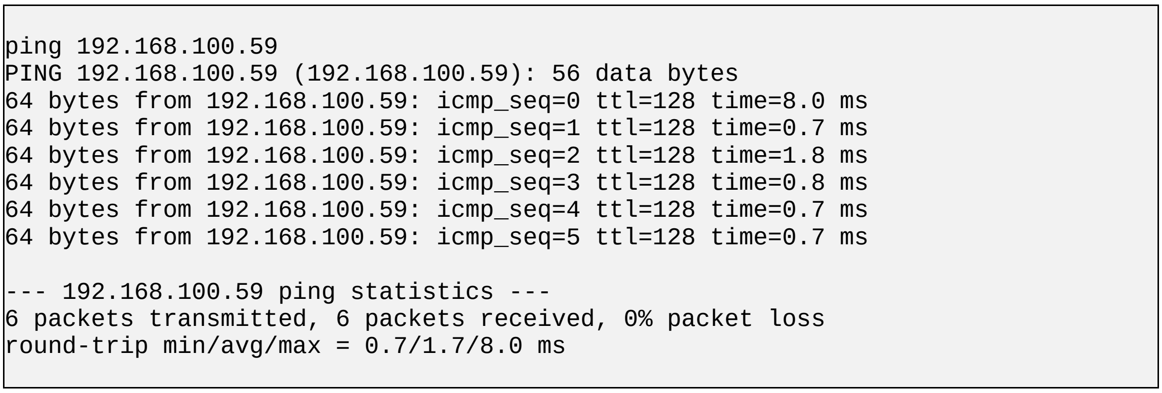
via serial connection:

1. Access the Main Setup Menu by typing **SETUP** and pressing <Enter>.
2. Select the Network Settings branch.
3. Select A) Ethernet Settings and select the Ethernet interface that corresponds to the one on the back panel that you plugged your network cable into.

4. Enter an IP address, subnet mask and--if necessary--a router address.
5. Toggle NAT on/off as desired.
6. If using this Ethernet interface for a VLAN connection, select this option to configure any of six VLAN connections. See the VLANs section in the Features chapter for details on how to configure.
7. Press <ESC> to go back one level in the menu tree, or <CTRL + C> to exit the Main Setup Menu and return to the command prompt.

Testing Network Connectivity

1. Verify that the network router is available to the unit by typing the command **PING <*IP_address*>**. A router is always a good candidate to test pings. The following screenshot is an example of a successful ping test.



```
ping 192.168.100.59
PING 192.168.100.59 (192.168.100.59): 56 data bytes
64 bytes from 192.168.100.59: icmp_seq=0 ttl=128 time=8.0 ms
64 bytes from 192.168.100.59: icmp_seq=1 ttl=128 time=0.7 ms
64 bytes from 192.168.100.59: icmp_seq=2 ttl=128 time=1.8 ms
64 bytes from 192.168.100.59: icmp_seq=3 ttl=128 time=0.8 ms
64 bytes from 192.168.100.59: icmp_seq=4 ttl=128 time=0.7 ms
64 bytes from 192.168.100.59: icmp_seq=5 ttl=128 time=0.7 ms

--- 192.168.100.59 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0.7/1.7/8.0 ms
```

2. Press <CTRL + C> to stop the ping testing. If <CTRL + C> is not pressed, the unit will continue pinging attempts indefinitely.
3. If there is an error message or no response from the router, first check the network settings and connection, then consult your System Administrator or Asentria Technical Support.
4. Using a Telnet client, connect to the IP address assigned to the unit.

SNMP Trap Setup

If you will be using your T850 to send SNMP traps, this section will help you ensure it is set up correctly.

Setup

1. Configure the network settings as described in the previous section.
2. Select the Network Settings then SNMP Settings sub-menu.
3. Verify the SNMP Community name is correct for your network.
4. Switch to the Action Definitions menu and enter the Hostname or IP address of the computer to receive the traps into the field, “Hostname/IP Address 1”.
5. Press <CTRL + C> to exit the Setup menu and return to the command prompt.
6. On the computer that will be receiving the SNMP traps, start your preferred SNMP trap manager.

Testing SNMP Traps

1. Using a Telnet client, connect to the IP address assigned to the unit.
2. Enter the command **DOTRAP** from the command prompt.
3. Verify that the trap manager receives the test trap.
4. If there is an error message or no response from the router, first check the network settings and connection, then consult your System Administrator or Asentria Technical Support.

What is a TeleBoss 850

The Basics

Fig 1: TeleBoss 850-0

» **Note:** Functionally, the TeleBoss 850-0 is identical to the T850-2 and T850-6 products, except that it does not have expansion bays on the back panel for insertion of different I/O Expansion Cards. Therefore, references in this manual to Expansion Cards (wireless modems, GPS) functions do not apply to the T850-0. However, the T850-0 does feature an on-board SensorJack port for use with Asentria Type2 EventSensors, many of which feature the same I/O as some of the optional Expansion Cards, so references to contact closures, analog sensors, and relay outputs are all applicable to the T850-0.

Fig 2: TeleBoss 850 (T850-2 on top, T850-6 on bottom)

The T850 is a powerful remote device management system which can collect and forward text records such as those used by Call Accounting and Telemanagement billing applications. These records are collected by the T850 from PBX serial ports, and in some cases directly over a TCP/IP connection. The T850 can also make a passthrough connection directly to devices connected on one of its serial ports, and can also connect you via web or Telnet to other devices on the same remote network as the T850. The T850 provides versatile alarm management of text-based alarms as well as interfaces with environmental monitoring equipment and contact closure alarms at your remote site. The T850 is a powerful remote network management solution for Call Accounting systems, Service Bureaus, and end users who need to collect PBX data as well as get remote access, and collect alarms from equipment at remote sites.

On-board I/O provides serial, Ethernet, and dialup connectivity. The T850-2 (11-inch) and T850-6 (17-inch) models provide two or six expansion slots respectively to allow addition of various communications and monitoring interfaces (Expansion Cards).

Communication Methods

The T850 has a diverse selection of communication methods available for different applications. The following methods can be used to either access the command processor or provide a passthrough connection to devices attached to the serial ports. All methods of connecting to the unit can be secured via password for protection of data and hardware.

- RS-232 serial
- Telnet
- Standard modem serial
- Security callback modem serial
- SSH
- HTTP Server

Data may be retrieved from or through the T850 by any of the following methods:

- Serial or modem connection to command processor (using Line or Zmodem) or pass-through
- Inline Mode (data in I/O1, data out I/O2)
- Telnet to command processor or passthrough
- Telnet real-time sockets
- FTP or SFTP push (automatic delivery to FTP or SFTP server)
- FTP or SFTP get (manual retrieval from FTP or SFTP server)

Alarms generated or detected within the T850 can be delivered through any of the following means:

- Modem callout
- SNMP trap
- Email
- Dialup pager
- Script actions
- SMS Messages
- Asentria Alarms
- Relays (if configured with optional relay Expansion Card)

Data Storage

Basic data storage in the T850 is accomplished in a database of four files – FILE1, FILE2, EVENTS, and AUDIT. FILE1 and FILE2 are typically associated with Serial Port I/O 1 and Serial Port I/O 2 respectively, although either serial port can store to either FILE1 and FILE2, or both. Data collected via IP Record Collection (IPRC) is also stored to either FILE1 or FILE2. EVENTS and AUDIT are log files generated from the Event Log Settings and Audit Log Settings menus per the parameters set there. The number of records stored in each these four files can be displayed using the **DIR** command on any connection to the command processor, including FTP.

The T850 also features three “auxillary” files for storage of data to be used in scripting functions, named AUX1, AUX2, and AUX3. These three files are not displayed with the **DIR** command, although data collected via serial port or IPRC can be stored to any of these three auxillary files in addition to FILE1 and FILE2. Refer to the Scripting chapter for more information on processing data stored in AUX1, AUX2 and AUX3.

Remote Access

The T850 can provide an administrator transparent access to devices connected to the serial ports of the unit via serial, modem, and Telnet pass-through connections. This sort of access can be used to configure, maintain, or manipulate devices that would normally have no remote access.

Serial Monitoring (Data Events)

The T850 can be used to monitor incoming data for user-defined strings and then report the event via several avenues. The T850 allows for up to 1000 different data events. Each data event contains independent actions, counters, and other unique settings. Data events triggered within the T850 can be logged to an Event Log. This file can be viewed through the Event Log section of the Setup menu, via the **TYPE EVENTS** command, or via FTP or the web interface.

Environmental Monitoring

Through the use of external EventSensor modules and/or internal Expansion Cards, a variety of environmental sensor monitoring and alarming capabilities are available in the T850. Each individual sensor can be configured with independent actions, counters and other unique settings. Sensor events triggered within the T850 can be logged to an Event Log. This file can be viewed through the Event Log section on the Setup menu, via the **TYPE EVENTS** command, or via FTP or the web interface.

Event Notification

Actions generated or detected within the T850 can be delivered through any of the following means:

- Modem callout
- SNMP trap
- Email
- Dialup pager
- Script actions
- SMS Messages
- Asentria Alarms
- Relays (if configured with optional relay Expansion Card)

Audit Log

The T850 has the capability to log many types of administrative events, from serial port handshaking alarms to login attempts. These Audit Log entries are stored in a file and can be viewed through the Audit Log section of the Setup menu, via the **TYPE AUDIT** command, or via FTP or the web interface.

Integration with SitePath

Using the T850 in conjunction with Asentria's SitePath Remote Management System, you can create secure and controlled IP access to remote servers and appliances co-located on the same remote network as the T850. SitePath uses an integrated SSL or IPSEC VPN implementation which simplifies otherwise complex VPN setup down to a few easy steps, allowing users to access remote devices via the SitePath VPN Gateway. The T850 plus SitePath provide IP routing to authorized remote network addresses, and prevents unauthorized access to any other addresses on the remote LAN.

Parts Identification

Features and Accessories

Standard Equipment
The base T850 comes with the following standard on-board equipment:

- AC or DC Power Input
- 32MB logging database for CDR or other text records
- 2 – RJ45 DTE serial I/O ports
- 1 – 9 pin Mini DIN SensorJack port for connection of Type2 EventSensors
- 2 – 10/100Mb Ethernet interfaces with support for six 802.1Q VLAN interfaces on each.
- 1 – MMC memory I/O slot
- 0, 2 or 6 – Expansion Card slots
- Internal lithium coin-cell type battery backup*/**

***** Battery backup preserves clock operation when power is not present. Data records and settings are stored in non-volatile memory and therefore do not require battery backup.

** **CAUTION: THERE IS A RISK OF EXPLOSION IF THE BATTERY IS REPLACED BY AN INCORRECT TYPE. DISPOSE OF USED BATTERIES ACCORDING TO THE INSTRUCTIONS.** The instructions are that lithium batteries can be recycled, and you should contact a recycling orgainization in your area for details.

In addition to the above components, the standard unit is shipped the following accessories:

- This product manual on the Documentation and Software CD
- RJ45 M-M unshielded serial cable and RJ45/DB9 straight thru adapter for each serial port ordered
- RJ45 Ethernet cable for each Ethernet port
- Power supply adapter (for AC units), or wiring harness and Molex plug (for DC units)

Options
Each of the following components is optional and may be installed on a T850:

- Additional RJ45 DTE serial I/O ports in sets of 4 to total 6, 10, 14, 18, 22, or 26 ports
- 64MB logging database for CDR or other text records
- Internal 33.6K baud, or wireless modem
- Run-time battery (enables the unit to function for a period of time without power, if enabled).
- Expansion Cards configured as wireless modem, ADSL modem, contact closures, analog sensors, and relays.

The T850 may come with any of the following accessories as well, depending on the configuration or order:

- Modem cable for internal modem
- Antenna for wireless modem Expansion Card
- Serial cables and RJ45/DB9 adapters for 4-port Serial Expansion Cards

LEDs, Ports, DIP Switches and Buttons

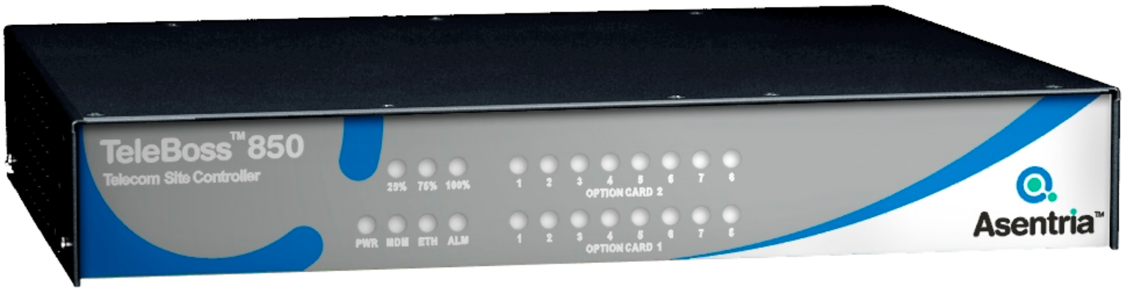


Fig 3: Front panel (T850-2)

LEDs – Front Panel

Power
The Power LED is green and has two operational states. During the boot up cycle, it will blink once every second until the boot sequence is complete. During normal operation, it is steady on with a blink every 5 seconds.

MDM (Modem)
The MDM LED lights solid green whenever the modem is connected and blinks when the modem is dialing out.

ETH (Ethernet)
The Link LED lights solid green whenever an active Telnet or FTP connection is made to the unit.

ALM (Alarm)
This LED is reserved for future use.

25% - 75% - 100%
The T850 has three LEDs to indicate file full status. A blinking percentage full LED indicates the database has less than the amount indicated by that LED, but more than the previous. A solid lit LED indicates the database percentage is at or over the value for that LED.

Expansion Card *n*
Each optional Expansion Card has eight LEDs associated with it that may or may not be used.

LEDs – Back Panel

Each RJ45 port on the back panel has two LEDs associated with it – one on the Right of the port, one on the Left of the port.

Ethernet Ports (ETH1 and ETH2)

- Right – Lights solid red when an Ethernet cable is connected to the port and an active Ethernet network. The LED is off when the cable is disconnected from the network, or the Ethernet Port.
- Left – Flashes yellow/green when network data (TCP packets) is being transmitted or received across the port. When no data is actually being transmitted/received, this LED is off.

I/O Port 1 & 2 (and any additional 4-I/O Port cards that may be installed)

- Right – Lights solid green when a correctly configured cable from another device is connected to it. Otherwise this LED remains off. As the I/O Port receives or transmits data, this LED will flash red.
- Left – Lights solid green when power is applied to the T850, regardless of whether a cable is connected to the I/O Port or not.

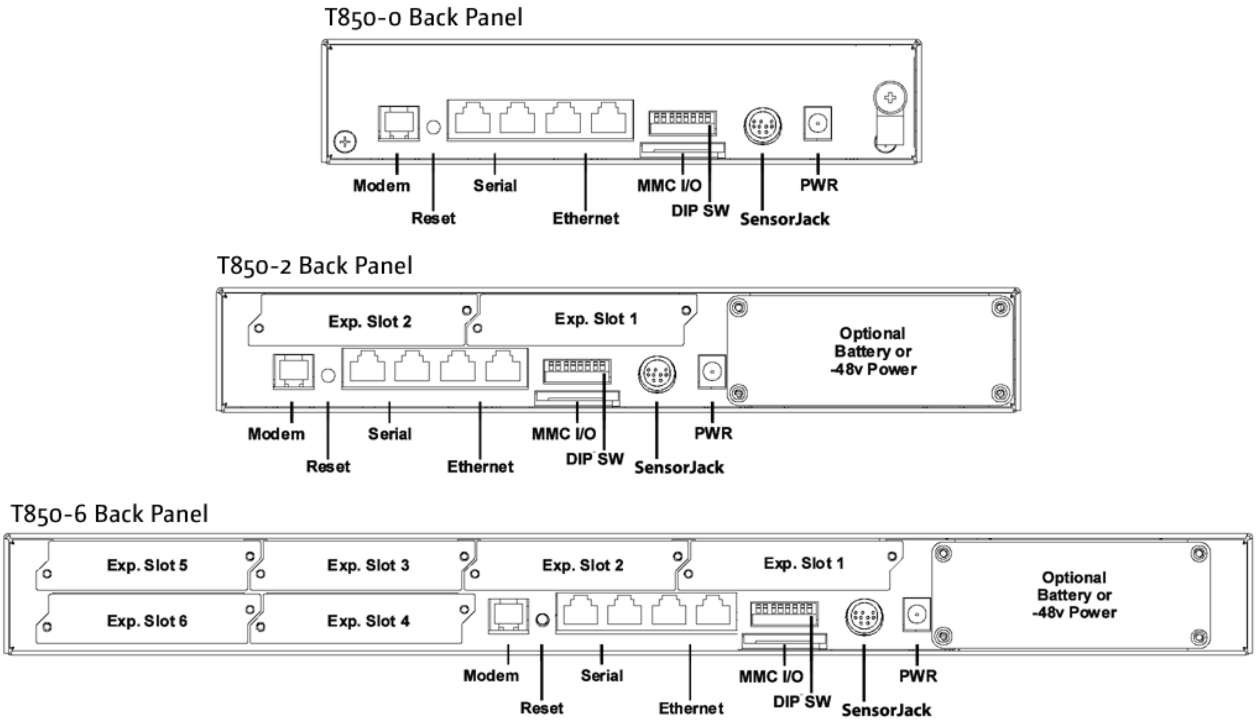


Fig 4: T850 Back Panel configurations

The above drawings show all three models of the T850 which have the newer 9-pin Mini DIN SensorJack port for connecting Asentria Type2 EventSensors. You may have an older model T850 which does not have the SensorJack Port, in which case you can use Serial Port I/O 1 set to ES Bus Mode with a Serial to ES Bus Adapter to connect Asentria Type1 EventSensors.

Ports

Memory I/O

The slot labeled Memory I/O can be used for the optional external Temperature Sensor, which is a small MMC card. Eventually T850's may also be upgraded using a MultiMedia Card (MMC) in this slot.

Sensor

The SensorJack port is a 9-pin Mini DIN connector for use with Asentria Type2 EventSensors.

Ethernet

The Ethernet 10/100Mb interfaces are standard RJ45. Either of these standard connectors will connect the T850 to an Ethernet hub or switch. Refer to the Telnet/TCP Connections section in the Features chapter for further information regarding a number of different types of Telnet connection options. LEDs by each Ethernet connection on the back panel flicker when packets are being transmitted/received on that port.

Serial Ports

Each of the two (or more) serial ports is configured as a DTE port using an RJ45 connector. This is the standard recommended pinout for EIA/TIA-561 for 8 pin RJ45 connector:

PIN1	=RI	=RING INDICATOR, INPUT to the T850
PIN2	=DCD	=CARRIER DETECT, INPUT to the T850
PIN3	=DTR	=DATA TERMINAL READY, OUTPUT from the T850
PIN4	=SIGNAL GROUND	
PIN5	=RXD	=RECEIVED DATA, INPUT to the T850
PIN6	=TXD	=TRANSMITTED DATA, OUTPUT from the T850
PIN7	=CTS	=CLEAR TO SEND, INPUT to the T850
PIN8	=RTS	=REQUEST TO SEND, OUTPUT from the T850

The DB9 female cable end which mates with the serial port connectors of connected devices will often have a pair of screw-down cable screws. These cable screws should be used to assure a solid connection of the cable with the device.

Default settings for the serial ports are 19200-baud, 8-bit word length, no parity, and one stop bit (19200, 8N1). Use the internal setup menu to adjust these settings.

Internal Modem
If a dialup POTS modem is installed, an RJ11 (typical U.S. phone) connector is used. A POTS (analog) dialup phone line is inserted into this connector. The modem installed within this unit is FCC certified. For further information, consult the Internal Modem Guidelines appendix or the serial number label on the bottom of the T850.

* Expansion Card Slots
Optional Expansion Cards can be installed to expand the capabilities of the T830-2 and T830-6. Contact Asentria Sales (sales@asentria.com) for more information on Expansion Cards.

DIP Switches
The bank of 8 DIP switches on the back panel of the T850 are used to control the baud and parity settings of I/O2, to set the operational mode for I/O2, and to put the unit into "boot load mode" where it can be forced to load a new application (firmware image). The following table shows how to set the various DIP switches to obtain certain settings:

I/O 2 Baud	**SW1**	**SW2**
2400	DOWN	DOWN
9600	UP	DOWN
19200	DOWN	UP
115200	UP	UP
I/O 2 Word, Parity	**SW3**	
8N1	DOWN	
7E1	UP	
I/O 2 Mode	**SW4**	
Command Mode	DOWN	
Data Mode	UP	
Boot Load Mode	**SW8**	**SW1 thru SW7**
No Forced App Reload (Default)	DOWN	X (don't care)
Forced Application Reload	UP	UP

» **Note:** Boot Load Mode can only be set by flipping ALL DIP switches to the UP position. This is not a setting that can be configured via internal menu settings, or Setting Keys.

» **Note:** For settings that can be set either via DIP switch, internal menu settings, or Setting Keys, the T850 always pays attention to the last setting, regardless of how it was done. So if the internal setting for I/O 2 Port Mode is Command, and someone flips SW4 to the UP position, the Mode is immediately set to Data.

Buttons
The only button on the T850 is the Reset button located on the back panel next to the left of serial port I/O 2. The Reset button can be used for two different functions:

1) To reset the T850 – press the Reset button for approximately 1 second and T850 will be begin the reboot process as described in the Power Up Sequence section on the next page.

2) To activate the Button Unlock feature which resets the username and password back to default.

Getting Connected

Power Up Sequence

On startup, the T850 goes through the following boot sequence in approximately 55 seconds:

1) The power LED flashes once each second for 30 seconds.
2) The LED for Expansion Card 1 go through a 15 second flashing sequence.
3) All LED's then go off for approximately 5 seconds.
4) Power, Modem (if installed) and Ethernet LEDs light for 5 seconds, then Modem and Ethernet go off.
5) Power LED will blink once every 5 seconds as a "heartbeat" while the T850 is powered on.

Default Passwords

The T850 uses a very flexible system for managing users, passwords, and access rights. By default, the following fifth user profiles are enabled. Note that if a password is defined without a user name, the profile is defined just by the user name. All of the default profiles are password-only. All passwords are masked: ********

The default settings are configured to low security for your convenience in setup. It is highly recommended that you change these passwords and record them in a secure location.

User	Password	Login To
User1	**SMDR**	**Command processor**
User2	**SUPER**	**Command processor**
User3	**ACCESS1**	**Passthrough/File 1**
User4	**ACCESS2**	**Passthrough/File 2**
User 11	**Username: admin** **Password: password**	**Command processor**

The Status Screen

The T850 Status screen is the unit's one-stop informational source. Most of the information that a user would need to know about the unit is available through this display. This section outlines this data and highlights why it is useful.

```
TeleBoss 850 2.05.980 STD  -  Status
Site Name       : 850-850000163
Serial Number   : 850000163        Eth 1    : STATIC
Date            : THU 04/29/2010   IP Addr  : 0.0.0.0
Time            : 08:02:48         MAC Addr : 00:10:A3:60:02:3E
Memory          : 32768K           Eth 2    : STATIC
% Full Alarm    : OFF              IP Addr  : 0.0.0.0
No-Data Alarm 1 : OFF              MAC Addr : 00:10:A3:60:02:3F
No-Data Alarm 2 : OFF              Modem    : Yes
Duplex          : FULL
-------------------------------------------------------------------------
Port   Baud/Etc.  Recs     Bytes    Full Wrap Name
IO1  : 19200,8N1  00000000 00000000   0% OFF  I/O 1
IO2  : 19200,8N1  00000000 00000000   0% OFF  I/O 2

COMPLETE
```

TeleBoss 850 indicates that this product is the T850, followed by **2.05.980 STD,** the currently loaded firmware version.

Site Name is the identifier assigned to each T850 by the end user in the General Settings menu.

Serial Number is the factory-assigned, unique serial number for this T850.

Date and **Time** display the current date and time.

Memory indicates the amount of flash memory configured for storage of data.

% Full Alarm / No Data Alarm *n* indicates the current ON/OFF status of the % Full Alarm, and No Data Alarms 1 and 2, respectively.

Duplex controls the echo settings for the command processor. Full duplex causes the T850 to echo all characters sent to the remote device. Half duplex turns off character echo.

Eth 1 and **Eth 2** displays STATIC, DHCP, or VLAN, depending on which mode each of the two Ethernet interfaces is configured for.

IP Add and **MAC Add** immediately following Eth 1 and Eth 2 are the network IP address assigned to each Ethernet card, and that cards MAC address. The MAC address of both Ethernet cards can also be found on the unit's serial number label.

Modem indicates whether the optional internal modem is installed.

The default serial port names of **I/O 1, 2, etc** are displayed for each of the installed serial ports along with the following information:

Baud Rate/Etc. displays the baud, word length, parity, and stop bit settings for each installed serial port.

Recs shows the number of carriage return-delimited records stored within the file associated with each serial port.

Bytes displays the amount of storage allocated for the above records.

Full is a rough percentage indicator of how much data is stored in a particular file.

Wrap indicates the ON/OFF status of whether file wrapping is enabled on a particular port. When ON, a unit that is 100% full will overwrite the oldest buffered records with new ones.

Name displays the target name, which is an optional name given to the device connected to the port. This target name is used in event notifications and can be configured in the Serial Settings menu for each port.

Setup Menu

Overview

This section displays screen shots and descriptions taken from the command prompt menu system. However, the menu structure and options are the same as the web interface.

The Setup menu contains all of the configuration options available on the T850. It is organized in a logical tree structure with all settings classified under the following groups:

```
TeleBoss 850 - Main Setup Menu
A) Network Settings
B) Serial Settings
C) Modem Settings
D) Security Settings
E) Alarm/Event Definitions
F) Action Definitions
G) General Settings
H) Event Log Settings
I) Audit Log Settings
J) Scripting Settings

Enter your Selection:
```

Note: Some menus may not be available, depending on your hardware configuration.
Note: Passwords are case sensitive and are masked in all menus for security reasons.

Each section in this chapter will go over one of the above setup branches, outlining the options within.

Press either <ESC> or <Enter> to go back one level in the menu tree, or <CTRL + C> to exit any setup menu and return to the command prompt.

Since this product allows for multiple simultaneous command processors, two administrators could conceivably change the same option at the same time, but due to the multitasking nature of the T850, the changes are processed in the order received.

The T850 processes setup changes in real time. In other words, the unit begins to implement changes to its configuration as soon as they are entered. There is no need to exit a setup menu or reboot the unit to apply changes. The exception to this rule is IP-specific network settings. Changes to these settings are implemented only after all open Telnet command processors are closed.

Option Types

String entry
There are several different types of inputs employed within the Setup menu. The most common is the string type entry:

```
  A) Site Name                   [Test Site]
```

When selected, this setting will provide a prompt requesting a new value. You may press <Enter> or <ESC> to abort the option entry or press <SPACE> and <Enter> to delete the current value and leave it blank. Some numerical or required settings will not allow you to leave an option blank, so pay attention to the unit's response when attempting to delete a setting's value.

Toggle
The second most common option type is the toggle type option:

```
  A) Enable Web Interface      [OFF]
```

When selected, this option will not prompt for a new value. It will simply cycle to the next available option in its list. This is typically used for options with two or three choices. Most often it is in an ON/OFF form, but could be a series of options such as "NONE", "1", and "2".

Alarm actions (action list)
Alarm actions have their own unique method of entry. Refer to the Action List section in the Features chapter for more information.

Option list
The option list type is similar to the toggle type in that it provides a list of options to choose from:

```
TeleBoss 850 - Serial Port 2 Baud Rate
A) 300
B) 600
C) 1200
D) 2400
E) 4800
F) 9600
G) 19200
H) 38400
I) 57600
J) 115200
```

After selecting an option, you are immediately returned to the previous menu. The new value will be displayed to the right of the setting name, letter, or number.

Web Interface

The T850 has a built-in HTTP web server that can be used to configure the unit from anywhere the unit can be accessed on the network or Internet. Once you have enabled it through the network section of the Setup menu, simply connect to ***http://<IP address of T850>*** or ***https://<IP address of T850>*** to use Secure Sockets Layer (SSL). See Web Interface Settings menu for further description.

Upon connection, you will be greeted by a login screen. Log in with your Login ID (Username) and Password. These are the same credentials you would use to log into the command prompt. Once logged in, the General Status screen will be displayed with a menu bar across the top of the page that displays the same menu options as the command prompt menu system. From here, you can alter any setting in the same way you could via the prompt.

Main Setup Menu

```
TeleBoss 850 - Main Setup Menu
A) Network Settings
B) Serial Settings
C) Modem Settings
D) Security Settings
E) Alarm/Event Definitions
F) Action Definitions
G) General Settings
H) Event Log Settings
I) Audit Log Settings
J) Scripting Settings
```

Network Settings contains settings for network communication, SNMP, FTP, PPP, Email, and more.

Serial Settings contains settings pertaining to the function of each serial port.

Modem Settings contains modem init settings and modem-specific security options.

Security Settings contains all user profiles, RADIUS configuration, and general security settings.

Alarm / Event Definitions contains all of the settings that define events within the T850.

Action Definitions contains configurations for all of the actions possible when events are detected.

General Settings contains the site name, answer string, confirmation prompt, date/time, and other general settings.

Event Log Settings allows for configuration and displaying of the Events Log.

Audit Log Settings allows for configuration and displaying of the Audit Log.

Scripting Settings allows for configuration of scripts.

Network Settings

The Network Settings menu contains all of the options pertaining to network communication.

```
TeleBoss 850 - Network Settings
A) Ethernet Settings
B) Default Router                       [0.0.0.0]
C) Name Resolution Settings
D) Telnet Duplex                        [FULL]
E) Inactivity Timeout                   [0]
F) IP Record Collection Settings        [OFF]
G) Web Interface Settings               [ON]
H) EventSensor Reporting Settings
I) SNMP Settings
J) FTP Settings
K) PPP Settings
L) Email Settings
M) Real-Time Socket Settings
N) SNMP Trap Capture Settings
O) IP Address Restrictions
P) Static Route Settings
Q) DSL Settings
R) VPN Settings
S) CPE Settings
   Note: Changes to IP Address, Subnet Mask, or Router
         Address will not take effect until any open
         Telnet command processor sessions are ended.
```

Ethernet Settings displays the menu where you can configure each of the two Ethernet interfaces, and also any of the six VLAN interfaces that each supports.

Default Router displays the configured default router (gateway) for the unit. Refer to the Default Router section in the Features chapter for more information.

Name Resolution Settings allows you to configure the IP addresses of up to two Domain Name Servers (DNS).

Telnet Duplex controls the echo settings for Telnet. Full duplex causes the unit to echo all characters sent to the remote device. Half duplex turns off character echo. Default setting is Full.

Note: If Duplex is set to Half, set **`sys.terminal.mode`**`=OFF`. Otherwise, characters will continue to be echoed to the terminal screen.

Inactivity Timeout sets the number of minutes (0 - 255) before a network connection with no activity will be terminated. A setting of 0 means an inactive connection will not be terminated. Default setting is 0.

IP Record Collection Settings displays the IP Record Collection Settings menu where an IPRC protocol can be selected and configured to collect data from various types of IP-enabled switches.

Web Interface Settings displays the Web Interface Settings menu where you can toggle the web interface ON or OFF, set the session timeout (0 - 65535 minutes), and set the TCP port number for the web connection.

EventSensor Reporting Settings displays the EventSensor Reporting Settings menu where where the parameters of the EventSensor Reporting feature can be configured.

SNMP Settings displays the SNMP Settings menu where you can configure the SNMP community name, and spoofed PPP/Trap IP address.

FTP Settings displays the FTP Settings menu, where you can configure automatic FTP pushes of buffered data.

PPP Settings displays the PPP Settings menu, where you can configure settings for PPP Dialout, PPP Hosting, and IP Routing

Email Settings displays the Email settings menu, where you can configure the SMTP server address, Email domain name, and authentication parameters.

Real-Time Socket Settings displays the Real-Time Socket Settings menus where you can configure real-time socket settings for each file of buffered data. Real-Time Sockets are used to collect data on TCP port 2201 from a serial port in real-time, while buffering data if the network connection goes down.

SNMP Trap Capture Settings displays the SNMP Trap Capture Settings menu where you can toggle this feature on or off, and select which file to store the traps in.

IP Address Restrictions displays the IP Address Restrictions menu, where you can limit Ethernet and PPP communications to or from specific IP addresses.

Static Route Settings displays the Static Route Settings menu where you can configure static network routes.

DSL Settings displays the DSL Settings menu where settings are configured so the T850 can communicate using the optional **ADSL Modem**.

VPN Settings displays the VPN Settings menu where settings are configured so the T850 can communicate with the optional Asentria SitePath secure, unified administration portal software.

CPE Settings displays the Customer Premises Equipment (CPE) Settings menu where up to 64 different networked devices can be configured to communicate with the optional Asentria SitePath secure, unified, administration portal software.

Ethernet Settings

Ethernet Settings displays the following menu where each of the two installed Ethernet cards can be configured:

```
TeleBoss 850 - Ethernet Settings
A) Ethernet 1
B) Ethernet 2

Enter your Selection: a


TeleBoss 850 - Ethernet 1 Settings
A) Mode                                 [STATIC]
B) IP Address                           [0.0.0.0]
C) Subnet Mask                          [255.255.255.0]
D) Router Address                       [0.0.0.0]
E) NAT                                  [ON]
F) VLAN Settings
```

Security Note: If the T850 is going to be exposed to the Internet, use the other security features available within the unit to prevent unauthorized access to your network. The other security features are SSH, SFTP, Strong Passwords, Challenge and Responses. Also shutdown unsecure connections such as Telnet and FTP.

Mode toggles between STATIC, DHCP, or VLAN – whichever is appropriate for this Ethernet port. Default setting is STATIC.

IP Address is the network address assigned to this Ethernet card. Default setting is 0.0.0.0

Subnet Mask sets the network subnet mask provided by the network administrator. Default setting is 255.255.255.0

Router Address sets the router address provided by the network administrator. Default setting is 0.0.0.0

NAT is an ON/OFF toggle to enable Network Address Translation. Default setting is ON.

VLAN Settings displays the following sub-menu where any of six individual VLAN connections can be configured. Refer to the VLANs section in the Features chapter for a detailed explanation of VLANs.

```
TeleBoss 850 - VLAN Settings
A) VLAN 1
. . .
F) VLAN 6

Enter your Selection: a

TeleBoss 850 - VLAN 1 Settings
A) ID                                 [0]
B) Priority                           [0]
C) IP Address                         [0.0.0.0]
D) Subnet Mask                        [255.255.255.0]
E) Router Address                     [0.0.0.0]
```

Note: The T850 does not heed changes to network configurations while you are connected to a command processor via Telnet, web interface, or SSH. Changes, including population of the candidate default router set, are pended until all network-based command processor sessions have ended. Open FTP and RTS connections will fail if these settings are changed during an open connection.

Name Resolution Settings

```
TeleBoss 850 - Name Resolution Settings
A) DNS Server 1                       [0.0.0.0]
B) DNS Server 2                       [0.0.0.0]
C) DNS Mode                           [MANUAL]
```

DNS Server 1 / **2** are the IP addresses of Domain Name Servers that you may want to configure so that you can use host names rather than IP addresses in functions where name resolution may be needed, such as; Email server, RTS push hosts, action IP settings, network time servers, scripting tcp connections, etc. Default setting for each DNS Server is 0.0.0.0.

DNS Mode toggles between MANUAL, ETH1-DHCP, ETH2-DHCP, and DSL. Default setting is MANUAL.

IP Record Collection Settings

```
TeleBoss 850 - IP Record Collection (IPRC) Setup
A) IP Record Collection                 [OFF]
B) Store Collected Data In              [FILE1]
```

IP Record Collection selects and displays a configuration menu for each of the IPRC protocols that the T850 supports: Generic Server, Avaya Reliable Session Protocol, Alcatel OmniPCX, Cisco CallManager 4.x, Generic Client (supports Siemens HiPath 4000), Intecom Telari, Nortel BCM, Syslog, NEC NEAX2400, and Cisco CallManager 5.x. Default setting is OFF.

Store Collected Data In sets the data file in which to store records received via IPRC. Default setting is FILE1.

Note: Refer to the IPRC chapter for a detailed explanation of IPRC.

Web Interface Settings

```
TeleBoss 850 Web Interface Settings
A) Enable Web Interface                                 [ON]
B) Web Session Timeout                                  [30]
C) HTTP Connection Port                                 [80]
D) HTTPS Connection Port                                [443]
```

Enable Web Interface is an ON/OFF toggle to enable the T850's internal web server. Default setting is ON.

Web Session Timeout sets the number of minutes (0 - 65535 minutes) a connection may remain idle before expiring. A setting of 0 means the connection will never automatically expire. Default setting is 30.

HTTP Connection Port is the TCP port through which an HTTP connection is made. Default setting is 80.

HTTPS Connection Port is the TCP port through which an HTTPS connection is made. Default setting is 443.

Connect using ***HTTP://<IPaddress of T850>*** or ***HTTPS://<IPaddress of T850>*** to use Secure Sockets Layer (SSL). You will be greeted by a login screen. Log in with your Login ID (Username) and Password. These are the same credentials you would use to log into the command prompt. Once logged in, the Output Status screen will be displayed, with a menu bar across the top of the page that displays the same menu options as the command prompt menu system.

Note: If using SSL, the SSL certificate will show "localhost" as the name, which may cause a certificate security warning to pop up, depending on the browser being used. The certificate may then be permanently accepted so the warning doesn't appear each time.

EventSensor Reporting Settings

```
TeleBoss 850 EventSensor Reporting Settings
A) EventSensor Report To IP                             []
B) EventSensor Report To Port                           [4000]
C) Enable EventSensor Reporting Host                    [OFF]
D) EventSensor Reporting Host Port                      [4000]
```

Event Sensor Report To IP sets the IP address of the host unit a sensor connected to this T850 would report to.

Event Sensor Report To Port sets the TCP Port that a sensor connected to this T850 would use to report to a host Asentria device. Default setting is 4000.

Enable EventSensor Reporting Host is an ON/OFF toggle to enable this T850 to be a host for EventSensor reporting from another Asentria device. Default setting is OFF.

EventSensor Reporting Host Port sets the TCP Port that this T850 will use for receiving sensor reports from another Asentria device. Default setting is 4000.

For a further explanation of EventSensor Reporting, refer to the EventSensor Reporting section in the Features chapter

SNMP Settings

```
TeleBoss 850 - SNMP Settings
A) SNMP Community                        [public]
B) Trap Settings
C) Security Method                       [MD5-DES]
D) PPP/Trap IP Address Spoofing          [0.0.0.0]
```

SNMP Community sets the SNMP community name to use. Default setting is Public. (Max length is 23 chars)

Trap Settings displays a menu that allows you to configure various Notification settings.

Security Method toggles between MD5-DES and SHA-AES to controls whether MD5 and DES, or SHA-1 and AES, are used for authentication and privacy, respectively, for for SNMPv3 get/set/trap operations. Default setting is MD5-DES.

PPP/Trap IP Address Spoofing allows you to configure the IP address to be displayed in an SNMP trap sent over a PPP connection. If undefined, the T850 PPP IP is used. Default setting is 0.0.0.0

Trap Settings

```
TeleBoss 850 - Trap Settings
A) Notification Attempts (0=infinite)                [5]
B) Notification Timeout (seconds)                    [60]
C) Notification Cycles (0=infinite)                  [10]
D) Notification Snooze Period (minutes)              [60]
E) Notification Security Name                        []
F) Notification Security Password                    [********]
```

Notification Attempts sets the number of attempts (1 to 65535) of sending a notification (trap/inform) per cycle (that is, the initial attempt + retries). If this is 0 then there is 1 infinite cycle. Default setting is 5.

Notification Timeout sets the number of seconds (3 to 60) between two attempts to send an SNMP notification in the same cycle. Default setting is 60.

Notification Cycles sets the maximum number of cycles (0 to 60) to try per notification action, where one notification action corresponds to one "inform" keyword in an action list for an event. A cycle is a set of notification attempts delimited by a successful action delivery or snooze period. Default setting is 10.

Notification Snooze Period sets the time in minutes (1 to 1440) between two SNMP notification cycles for any one notification action. That is, if you have two events generate informs, each inform will have its own timeouts for retries and cycles, and its own snooze period. Default setting is 60.

Notification Security Name / Password sets the name and password used for authentication when sending SNMPv3 traps. (Max length for each is 31 chars)

Note: SNMP traps are *not* a guaranteed means of delivering notifications. Traps are a one-way IP network datagram and the device receiving traps does not acknowledge them. Therefore, if the trap does not reach its intended destination for whatever reason, the sending device has no way of recognizing this and resending the trap.

FTP Settings

```
TeleBoss 850 - FTP Settings
A) FTP Push Enable                      [OFF]
B) FTP Server Address                   []
C) Username                             [Default FTP Username]
D) Password                             [********]
E) Account                              []
F) Directory                            []
G) Minutes Between Push Attempts        [1440]
H) Select Files to Push
I) Remote File Names
```

FTP Push Enable toggles between OFF, REGULAR, and SECURE. Default setting is OFF.

> **Setting Key: `net.ftppush.sftpport`**
> By default, Secure FTP (SFTP) uses TCP Port 22. For security purposes, this can be changed to any TCP port between 1 - 65535 as directed by users network system administrator.

FTP Server Address is the IP address or host name of the FTP server to push to. (Max length 64 chars)

Username/Password defines the login credentials that are able to access the remote FTP server. (Max length Username is 126 chars) (Max length Password is 31 chars)

Account is a third login option used only on some FTP servers. Consult your network administrator to see if this is necessary. (Max length 126 chars)

Directory is the path used to transfer the file(s). The file(s) is transferred to the root login directory if this option is left blank. (Max length 253 chars)

Minutes Between Push Attempts sets the number of minutes (1 to 9999) between FTP push attempts. Default setting is 1440 minutes.

Select Files to Push displays the FTP File Selection menu where you can select which files are pushed by toggling ON or OFF. Default setting for all is ON, except for Audit Log, which is OFF.

```
TeleBoss 850 - FTP File Selection
A) Data File 1                          [ON]
B) Data File 2                          [ON]
C) Events File                          [ON]
D) Audit Log                            [OFF]
```

Remote File Names displays the FTP File Names menu where you can give each file a name other than the default name, and/or prepend a date, time, and/or unique sequence # to the file name.

```
TeleBoss 850 - FTP File Names
A) Include Date in Filename             [OFF]
B) Include Time in Filename             [OFF]
C) Include Sequence #s in Filename      [OFF]
D) Data File 1                          [FILE1]
E) Data File 2                          [FILE2]
F) Events File                          [EVENTS]
```

Include Date in Filename toggles OFF, ON, or YYYYMMDD. When set to ON, the date is formatted with a 2-digit year (YYMMDD). When set to YYYYMMDD, the date is formatted with a 4-digit year. Either of these options will cause the date of the file transfer to be appended to the name of each transferred file of data. Default setting is OFF.

Include Time in Filename is an ON/OFF toggle to enable the addition of the file transfer time to the name of each transferred file of data. Default setting is OFF.

Include Sequence #s in Filename is an ON/OFF toggle to enable the addition of a unique sequence number to the beginning of the name of each transferred file of data. This ensures that no two transfers will have the same file name. Default setting is OFF.

Data File *n* / Events File are text-entry fields where the name each data file will have on the remote server (not including any date, time, or sequence numbers) can be configured.

Note: There is no menu option to set "FTP Auto-delete", a setting that will cause all data in the file to be deleted when polled via the FTP 'get' function. To enable FTP Auto-delete, set **`net.ftp.autodel`**`=ON`.

Once FTP Push has been configured, entering the **PUSHTEST** command will test the connectivity to the FTP server and write a "log in" and "log out" entry to the Status File in the directory you configured. No data is pushed with this command. Connection data displayed on the terminal screen is useful if the connection fails.

An immediate push of data can be done using the **PUSHNOW** command.

PPP Settings

```
TeleBoss 850 - PPP Settings
A) PPP Dialout Settings
B) PPP Hosting Settings
C) IP Routing
D) Route Test Settings
```

PPP Dialout Settings displays settings pertaining to making outbound PPP network connections.

PPP Hosting Settings displays settings for hosting a PPP connection.

IP Routing displays settings for routing of IP packets between PPP connections and the LAN a T850 is connected to.

Route Test Settings displays settings for network monitoring/PPP backup connection settings. This menu allows you to configure up to three IP addresses to ping on a regular basis. If any of the IPs are down, the unit will fall back to a PPP dialout in order to maintain reliable network connectivity for sending SNMP traps.

PPP Dialout Settings

```
TeleBoss 850 - PPP Dialout Settings
A) PPP Dialout Enabled                  [OFF]
B) Telephone Number                     []
C) User Name                            []
D) Password                             [********]
E) Idle Connection Disconnect (sec)     [60]
F) Maximum Retries                      [3]
G) Carrier Detect Timeout (sec)         [60]
H) Login Sequence Timeout (sec)         [30]
I) Dialout Modem Init String            []
J) IP Address to Suggest                [0.0.0.0]
```

PPP Dialout Enabled is an ON/OFF toggle to enable PPP dialout. Default setting is OFF.
Telephone Number sets the phone number of the PPP host the T850 is to dial into. (Max length 48 chars)

User Name / Password sets the login credentials that are used to log into the PPP host. (Max length for each is 64 chars)

Idle Connection Disconnect (sec) sets the number of seconds to wait before disconnecting an idle connection. A setting of 0 means the unit does not disconnect due to an idle connection. Default setting is 60 seconds.

Maximum Retries defines the maximum number of times to retry a failed connection. Default setting is 3.

Carrier Detect / Login Sequence Timeout (sec) configure standard login timeouts, from 0 to 65535 seconds. Default setting is 60 seconds for Carrier Detect, and 30 seconds for Login Sequence.

Dialout Modem Init String sets the modem initialization string. (Max length 48 chars)

IP Address to Suggest sets an IP to try to acquire, if defined. Default setting is 0.0.0.0

> **Setting Key:**
> **`net.pppdial.downafter.ftppush`**
> Values are ON or OFF (default OFF). ON means that if FTP Push raised PPP, then it kills PPP when finished.

PPP Hosting Settings

```
TeleBoss 850 - PPP Hosting Settings
A) PPP Hosting Enabled                  [OFF]
B) Idle Connection Disconnect (sec)     [60]
C) Local (Device) IP Address            [192.168.105.1]
D) Remote (Caller) IP Address           [192.168.105.2]
```

PPP Hosting Enabled is an ON/OFF toggle to enable inbound PPP connection hosting. Default setting is OFF.

Idle Connection Disconnect (sec) sets the number of seconds (0 – 65535) to wait before disconnecting an idle connection. A setting of 0 means the unit does not disconnect due to an idle connection. Default setting is 60 seconds.

Local (Device) IP Address sets the IP address of the T850 for the PPP session. Default is 192.168.105.1

Remote (Caller) IP Address sets the IP address of the calling device for the PPP session. Default is 192.168.105.2.

IP Routing

```
TeleBoss 850 - IP Routing
A) Route PPP to Ethernet                [OFF]
B) Route Ethernet to PPP                [OFF]
C) Ethernet to PPP NAT Enable           [ON]
D) Ethernet Interface                   [ETH1]
```

Each of the above options toggles settings for routing TCP/IP packets of specific types and origins to and from a device connected via PPP.

Route PPP to Ethernet toggles ON/OFF to enable the T850 to forward IP frames originating on PPP that are not IP-addressed to the unit, as well as forward IP frames received on Ethernet that are associated with forwarded frames that originated on PPP. Default setting is OFF.

Route Ethernet to PPP toggles ON/OFF to enable the T850 to forward IP frames originating on Ethernet that are not IP-addressed to the unit, as well as forwards IP frames received on PPP that are associated with forwarded frames that originated on Ethernet. Default setting is OFF.
Ethernet to PPP NAT Enable toggles ON/OFF to enable the T850 to do network address translation on these forwarded frames. Default setting is ON.

Ethernet Interface toggles between ETH1, ETH2, or any of the six VLANs that can be configured on either ETH1 or ETH2, to inidcate which interface to use for the PPP connection. Default setting is ETH1.

Refer to the IP Routing section in the Features chapter for a detailed explanation of IP Routing.

Route Test Settings

```
TeleBoss 850 - Route Test Settings
A) Route Test Enable                    [OFF]
B) Minutes Between Tests                [10]
C) IP Address 1                         []
D) IP Address 2                         []
E) IP Address 3                         []
```

Route Test Enable is an ON/OFF toggle to enable route testing. Default setting is OFF.

Minutes Between Tests sets the number of minutes (0 – 65535) to wait between each round of testing. Default setting is 10 minutes.

IP Address *n* sets the hostnames or IP addresses to ping for the test.

Email Settings

```
TeleBoss 850 - Email Settings
A) SMTP Server Hostname/IP Address      []
B) Email Domain Name                    [asentria.com]
C) Authentication (LOGIN)               [OFF]
```

SMTP Server IP Address sets the hostname or IP address of the outbound mail server. (Max length 64 chars)

Email Domain Name sets the *@domain_name.com* to use when the T850 sends an Email. Default setting is "asentria.com". (Max length 48 chars)

Authentication (LOGIN) displays a menu to configure the credentials that may be required by your server for SMTP authentication. Some SMTP servers require an authentication to relay Emails. Default setting is OFF.

```
TeleBoss 850 - Email Authentication Settings
A) Authentication Enabled               [OFF]
B) Username                             []
C) Password                             [********]
```

Authentication Enabled is an ON/OFF toggle to enable Email authentication. Default setting is OFF.

Username / Password defines the login credentials. (Max length for each is 48 chars)

Real-Time Socket Settings

```
TeleBoss 850 - Real-Time Socket Setup
A) FILE1
B) FILE2
C) EVENTS

Enter your Selection: a

TeleBoss 850 - FILE1 Real-Time Data Socket Setup
A) Real-Time Socket Mode                [LISTEN]
B) Show Answer String on Connection     [ON]
C) Require Xon to Start Data Flow       [OFF]
D) Idle Connection Close Timer          [0]
E) Close Socket When File Empty         [OFF]
F) Real-Time Socket Push Hostname/IP    []
G) Real-Time Socket Push Port Number    [3000]
H) Real-Time Socket Push Retry Timer    [5]
```

Real-Time Socket Mode can be toggled to LISTEN, PUSH, and OFF. When set to LISTEN this functions like traditional real-time sockets on TCP port 2201. When set to PUSH the unit tries to make a TCP connection on the TCP port specified in G) Real-Time Socket Push Port Number. As long as a connection exists, the unit sends all data in the specified file on the connection as data become available. Default setting is LISTEN.

Show Answer String on Connection is an ON/OFF toggle to enable the prompt indicating successful connection to the Real-Time Socket (RTS) port. Default setting is ON.

Require Xon to Start Data Flow is an ON/OFF toggle to enable the Xon/Xoff data flow control requirement. Default setting is OFF.

Idle Connection Close Timer sets the number of seconds (0 – 255) to wait before disconnecting an idle connection. A setting of 0 means the connection will never automatically close. Default setting is 0.

Close Socket When File Empty is an ON/OFF toggle to set whether or not the T850 will automatically terminate the RTS connection when the file for this port has been emptied. Default setting is OFF.

Real-Time Socket Push Hostname/IP sets the hostname or IP address of the server where the unit will push the data if the RTS Mode is set to Push. (Max length is 64 chars)

Real-Time Socket Push Port Number sets the TCP-port number the RTS push should use. Default setting is port 3000.

Real-Time Socket Push Retry Timer sets the number of minutes (1 – 255) to wait before retrying an RTS push that has previously failed. Default setting is 5 minutes.

SNMP Trap Capture Settings

```
TeleBoss 850 - SNMP Trap Capture Settings
A) SNMP Trap Capture Enable             [OFF]
B) Store Collected Traps In             [FILE1]
```

SNMP Trap Capture Enable is an ON/OFF toggle to enable the capturing of SNMPv1 traps and SNMPv2c inform-requests (informs). Default setting is OFF.

Store Collected Traps In toggles between FILE1 and FILE2 to set the data file in which the collected traps/informs are stored. Default setting is FILE1.

Refer to the SNMP Trap Capture section in the Features chapter for a detailed explanation of SNMP Trap Capture.

IP Address Restrictions

```
TeleBoss 850 - IP Address Restrictions
 No IP Restrictions Established
A) Add Item to Table

Enter your Selection: a
Enter IP addresses that are allowed access:
0.0.0.0 allows all IP addresses
255.255.255.255 restricts all IP addresses
XXX.XXX.XXX.0 allows all IP addresses in a subnet
XXX.XXX.XXX.255 restricts all IP addresses in subnet

New IP Restriction:
```

This menu is used to manipulate the IP Restrictions table. Refer to the IP Address Restrictions section in the Features chapter for a detailed explanation of IP Address Restrictions. By default, no address restrictions are configured.

Static Route Settings

```
TeleBoss 850 - Static Route Settings
A) Route 1
. . .
H) Route 8

Enter your Selection: a

TeleBoss 850 - Static Route 1 Settings
A) Enable                               [OFF]
B) Destination Network                  [0.0.0.0/0]
C) Gateway                              [0.0.0.0]
D) Interface                            [NONE]

Enter your Selection:
```

Static routes are network routes that specify in a more or less permanent way (*static*) that traffic to a certain destination (destination host or destination network) gets *routed* out a certain interface or via a certain gateway. Static routes gives you the ability to fine-tune how outbound network traffic leaves the unit for up to eight different routes.

Enable is an ON/OFF toggle to enable a static route. Default setting is OFF

Destination Network is the network notation, i.e., w.x.y.z/s, where s is the significant bits. Default is 0.0.0.0/0.

Gateway is the IP address of the gateway. Default setting is 0.0.0.0

Interface displays a listing from which to select any one of the interfaces available on this T850 – None, Ethernet 1, Ethernet 2, Ethernet 1 VLAN 1, 2, 3, 4, 5, 6, Ethernet 2 VLAN 1, 2, 3, 4, 5, 6, Dialup Modem PPP, and Wireless Modem PPP. Default setting is NONE.

Refer to the Static Routes section in the Features chapter for a detailed explanation of Static Routes.

DSL Settings

```
TeleBoss 850 - DSL Settings
A) Start Mode                           [MANUAL]
B) Type                                 [PPPOA]
C) VPI                                  [0]
D) VCI                                  [0]
E) Encapsulation                        [VCM]
F) Mode                                 [BRIDGED]
G) Username                             []
H) Password                             [********]
I) IP Address                           [0.0.0.0]
J) Mask                                 [0.0.0.0]
K) Router                               [0.0.0.0]
```

Following describes the menu options for configuring the optional ADSL Modem. For more information regarding the operation of the ADSL modem, Setting Keys, DSL Routing example, and DSL Glossary, please refer to the ADSL Modem chapter later in this manual.

Start Mode toggles between MANUAL and AUTO to set how the DSL interface is to be raised. Set this to MANUAL to require user intervention to raise the DSL interface, or to let a VPN (if it is configured to use DSL) raise the DSL interface when the VPN needs to use DSL. Set this to AUTO to tell the unit to automatically raise the DSL interface upon boot. Default setting is MANUAL.

Type toggles between PPPoA, PPPoE, Static, or DHCP. This should be set as directed by your ADSL provider. This is the most important DSL setting since its value determines what other DSL settings are applicable to the DSL configuration. Default setting is PPPoA.

VPI sets the VPI (Virtual Path Identifier) used on the DSL interface. This should be set as directed by your ADSL provider and is required for DSL operation. Values are: 0 to 4095 Default setting is 0.

VCI sets the VCI (Virtual Channel Identifier) for the DSL interface. This should be set as directed by your ADSL provider and is required for DSL operation. Values are: 0 to 65535. Default setting is 0.

Encapsulation toggles between VCM and LLC to control whether the encapsulation is LLC (Logical Link Control) or VCM (Virtual Channel Multiplexed). This should be set as directed by your ADSL provider and is required for DSL operation. Default setting is VCM.

Mode toggles between BRIDGED and ROUTED to control whether the DSL is set up for Bridged mode or Routed mode when the DSL type is STATIC. Default setting is BRIDGED.

Username and **Password** specify the PPP Username and PPP Password for the DSL interface when the DSL type is set to PPPoA or PPPoE. These should be set as directed by your ADSL provider and are required for DSL operaton. Values are text strings, max length 64 characters.

IP Address sets the public IP address of the unit in the case where the DSL link is active. If the DSL type is STATIC, the user needs to set this. If the DSL Type if DHCP, it is set automatically. This should be set as directed by your ADSL provider. Value is a dotted quad IP address. Default setting is 0.0.0.0

Mask sets the subnet mask used on the DSL interface. If the DSL type is STATIC, the user needs to set this. If the DSL Type if DHCP, it is set automatically. This should be set as directed by your ADSL provider. Value is a dotted quad subnet mask. Default setting is 0.0.0.0

Router sets the router for the DSL interface. If the DSL type is STATIC, the user needs to set this. If the DSL Type if DHCP, it is set automatically. This should be set as directed by your ADSL provider. Value is a dotted quad IP address. Default setting is 0.0.0.0

VPN Settings

```
TeleBoss 850 - VPN Settings
A) General Settings
B) VPN 1
C) VPN 2
D) Commissioning Settings
```

Following describes the menu options for configuring VPN Settings. These settings are only for use with the Asentria SitePath secure, unified administration portal software. More information concerning the use of VPNs can be found in the VPN chapter in this User Manual, or in the SitePath User Manual. Contact Asentria Technical Support for more information.

General Settings displays a sub-menu where the VPN Mode, On-Demand Port as well as Active and SitePath VPN channels are configured.

VPN1 / VPN2 displays the configuration menu for each VPN.

Commissioning Settings displays a sub-menu where all the parameters for commissioning the T850 with the SitePath application are configured. Commissioning is the process of automatically configuring a unit and making SitePath aware of it at the same time. Commissioning is covered in detail in the SitePath User Manual.

General Settings

```
TeleBoss 850 - General VPN Settings
A) Mode                                [SSL CLIENT]
B) VPN On-Demand Port                  [60001]
C) Active VPN                          [NONE]
D) SitePath VPN                        [NONE]
```

Mode toggles between SSL Client, SSL Server, IPSec Host, and IPSec Private Subnet to specify the VPN mode configured on Asentria units that are currently connected for commissioning.

VPN On-Demand Port sets the port to use for VPN on-demand (VOD) communication. Values are: 0 to 65535. Default setting is 60001. See the VPN on-demand section in the VPN Chapter for more information.

Active VPN toggles VPN1, VPN2, or None to set which, if any, of the two available VPNs is active. Only one VPN can be active at a time. To disable VPN functionality, set this to “None”. Default setting is None.

SitePath VPN toggles VPN1, or None to control which VPN is used for SitePath. Currently only VPN1 can be used for SitePath. When SitePath is in use, set this to "VPN1". When SitePath is not in use, set this to "None”. SitePath typically configures this automatically. Default setting is None.

VPN1 / VPN2

```
TeleBoss 850 - VPN 1 Settings
A) Description                          []
B) Start Mode                           [MANUAL]
C) Public Interface                     [ANY]
D) Remote Address                       []
E) Remote Network                       [0.0.0.0/0]
F) IPsec Remote Authentication Key      []
G) IPsec Key Lifetime (seconds)         [3600]
H) Private Network                      [0.0.0.0/0]
I) SSL Protocol                         [UDP]
J) SSL Port                             [1194]
K) SSL Username                         []
L) SSL Password                         [********]
M) SSL Manual Configuration
```

Description sets identifying data concerning the VPN.

Start Mode toggles between MANUAL, AUTO-PASSIVE and AUTO-ACTIVE

- MANUAL means either the user starts the VPN, or in the case of VPN on-demand with SitePath, when conditions arise that require a VPN to be up (See VPN on-demand documentation for more details).
- AUTO-PASSIVE means that for a VPN in IPsec or SSL VPN server mode, the units listens for a VPN connection when the unit starts.
- AUTO-ACTIVE means that for a VPN in IPsec or SSL VPN client mode, the unit starts connecting to a VPN peer when the unit starts. When a VPN is started, it is in that starting mode until it is stopped. It can be stopped any any time, regardless of start mode, by a user (via the **`net.vpn.cmd`** key), or by conditions warranting the VPN to be down in VPN on-demand with SitePath.

Public Interface toggles between ANY, ETH1, ETH2, PPPP, WPPP, and DSL to set on what interface the VPN to SitePath rides.

- ETH1: Ethernet1
- ETH2: Ethernet2
- PPPP: POTS modem PPP (if PPP is down, unit will raise PPP to raise the VPN, so long as PPP dialout is configured).
- WPPP: Wireless modem PPP (if PPP is down, unit will wait until a connection be established, so long as Wireless modem is enabled).
- DSL: ADSL modem (if ADSL link is down, unit will raise ADSL to raise the VPN, so long as it is configured)

This setting must make sense with the default router and the network configuration. This means:

- If SitePath is off a local network, then the default router must be on the same interface as the VPN network interface.
- If SitePath is on a local network, then the VPN network interface must be for the network on which SitePath lies, and the default router is don't-care.

Remote Address sets the public IP address of the appropriate VPNG used in a VPN.

Remote Network sets the remote network for the VPN in network notation: the public IP of the appropriate VPNG suffixed with "/32" to specify that the VPN-tunneled network only goes to the VPNG.

IPsec Remote Authentication Key sets the authentication key required.

IPsec Key Lifetime (seconds) sets the amount of time in seconds (1200 – 86400) that will pass before automatic key regeneration occurs. Default setting is 3600 seconds.

Private Network sets the reserved subnet that the Element Management System (EMS) calculated for this unit.

SSL Protocol toggles between UDP and TCP to set the protocol SSL VPN uses to carry VPN traffic. Default setting is UDP.

SSL Port sets what port (TCP or UDP, as determined by the SSL Protocl) number the VPN uses. Default setting is 1194.

SSL Username / Password sets the username and password that a VPN in SSL CLIENT mode uses when it connects to an OpenVPN server. If the username is blank then the username "u<serial number>" will be used. E.g., "u5500009999" is the username the unit sends to the OpenVPN server if this setting is blank and the SSL Password setting is not blank. The Username and Password make it so there is an extra layer of authentication to fulfill in order for the VPN to connect. Note: the OpenVPN server must be configured appropriately for this.

SSL Manual Configuration displays a menu to set up to 16 manual configuration items for OpenVPN, when the VPN mode is either SSL Client or SSL Server. Any configuration items you need which are not automatically handled for you by the unit (such as SSL port, SSL password, certificates, etc.) should be configured here.

Commissioning Settings

```
TeleBoss 850 - Commissioning Settings
A) IPsec Remote Private IP Address      [0.0.0.0]
B) IPsec Commissioning Network          [0.0.0.0/0]
C) Group Settings
D) Contact Name                         []
E) Contact Number                       []
F) Commissioning State                  [Commission Unit Now]
G) Commissioning IP Address             [0.0.0.0]
```

Commissioning is covered in detail in the SitePath User Manual. Contact Asentria Technical Support for more information.

CPE Settings

```
TeleBoss 850 - CPE Pages
A) CPE Page 1 (CPEs 1-16)
B) CPE Page 2 (CPEs 17-32)
C) CPE Page 3 (CPEs 33-48)
D) CPE Page 4 (CPEs 49-64)

Enter your Selection:

TeleBoss 850 - CPE Settings
A) CPE 1                                [0.0.0.0]
.. . .
P) CPE 16                               [0.0.0.0]

Enter your Selection:

TeleBoss 850 - CPE 1 Settings
A) IP Address                           [0.0.0.0]
B) Name                                 []
C) Description                          []
D) Alarm Keep-alive Period (seconds)    [0]
E) Alarm Threshold                      [1]
F) Enable SitePath Access               [ON]
G) SSH to Telnet Bridging               [OFF]
```

Following describes the menu options for configuring CPE Settings. These settings are only for use with the Asentria SitePath secure, unified administration portal software and set up is beyond the scope of this manual. Contact Asentria Technical Support for further information.

IP Address sets the IP address of the CPE. Value is a dotted quad IP address. Default setting is 0.0.0.0

Name sets the name given to the CPE. The only restriction on the name is that it cannot have any double or single quotes (' or ") in it. (Max length is 24 chars)

Description sets a description of what the CPE device is. The only restriction on the description is that it cannot have any double or single quotes (' or ") in it. (Max length is 64 chars)

Alarm Keep-alive Period (seconds) set the number of seconds between periodic pings (ping cycle) sent by the T850 to the CPE to make sure it is "alive“. 1 ping frame is transmitted per CPE per ping cycle. Values are: 0 to 65535. Default setting is 0.

Alarm Threshold sets the number of times that the unit receives no response to the keep-alive ping from the device before triggering the CPE down event. Values are: 1 to 255. Default setting is 1.

Enable SitePath Access is an ON/OFF toggle to enable SitePath to communicate with the CPE through the unit.

SSH to Telnet Bridging is an ON/OFF toggle on CPE 1 thru 4 only, that enables an authorized user to make a Telnet connection to a Telnet-only CPE device while on an SSH connection to the T850. Refer to the SSH to Telnet Bridging section in the Features chapter for more information.

Serial Settings

```
TeleBoss 850 - Serial Settings
A) 1-I/O 1 Settings
B) 2-I/O 2 Settings
```

Note: Because I/O2 has all the settings the other serial ports have, plus a few more, it will be described in the section below with differences in other ports mentioned when necessary.

Serial Port Menu

```
TeleBoss 850 - Serial 2
A) Target Name                        [I/O 2]
B) Baud Rate                          [19200]
C) Data Format                        [8N1]
D) Handshaking                        [NONE]
E) Wrap Around                        [OFF]
F) Record Stamping
G) Character Masking                  [ON]
H) Data Alarm Enable                  [OFF]
I) Store Data To                      [2]
J) Store Alarms During Pass-Through   [OFF]
K) Duplex                             [FULL]
L) Inactivity Timeout                 [0]
M) Port Mode                          [COMMAND]
N) Inline Mode Handshaking            [XON/XOFF]
O) Strip Sent Pass-Through LFs        [OFF]
P) Strip Received Pass-Through LFs    [OFF]
Q) Disable Serial Setup via DIP Switch  [OFF]
R) Multiline Record Settings          [OFF]
S) Data Type                          [ASCII]
T) Change ETX to CR/LF                [OFF]
```

Target Name is the name given to the device connected to the other end of each port. The target name is used in event notifications. Default setting is I/O n. (Max length is 24 chars)

Baud Rate displays a selection menu for baud rates available for the port. These values range from 300 baud to 115200 baud. Default setting is 19200.

Data Format toggles settings for word length, parity, and stop bit settings. The available options are: 8N1, 7E1, 7O1, 7N1, and 8O2. Default setting is 8N1.

Handshaking is a toggle item with the following options: None, Xon/Xoff, Both, and DTR. Default setting is None.

Wrap Around is an ON/OFF toggle to set whether the incoming data will wrap (overwrite) the oldest data in the file should it become full. Default setting is OFF.

Record Stamping displays a menu that allows you to select whether the Date/Time and/or the Unit ID are pre-pended to each incoming data string. Default setting for Date/Time and Unit ID is OFF.

Character Masking is an ON/OFF toggle to enable the character mask. The character mask allows you to block most non-printing ASCII characters. Specifically, the following ASCII character values are blocked: 0, 1, 4-9, 11, 12, 14-31, and 128-255. Default setting is ON.

Data Alarm Enable is an ON/OFF toggle to enable data alarm monitoring for this port. Default setting is OFF.

Store Data To displays a menu that allows you to toggle ON/OFF the files to which incoming data on this port should be stored, if any.

Store Alarms During Pass-Through is an ON/OFF toggle to determine whether data strings that meet data alarm criteria are stored in the Events File when a pass-through session is active on this port. Default setting is OFF.

Duplex (Port 2 only) toggles between FULL and HALF. Full duplex causes the unit to echo all characters sent to the connected terminal when in Command mode. Half duplex turns off character echo. Default setting is FULL.

Note: If Duplex is set to Half, set `sys.terminal.mode=OFF`. Otherwise, characters will continue to be echoed to the terminal screen.

Inactivity Timeout (Port 2 only) is the time (1 - 255 minutes) before a serial connection with no activity will be terminated. A setting of 0 means an inactive connection will not be terminated. Default setting is 0.

Port Mode sets the port function.

- **I/O 1** toggles between DATA and ESBUS. DATA configures the port as an inbound RS232 data port. ESBUS configures the port to communicate with external RS485 Asentria EventSensors. (This requires the use of an RS232-RS485 adapter). Default setting is DATA.
- **I/O 2** toggles between COMMAND, DATA, and INLINE. COMMAND allows for serial command processor access. DATA configures the port as an inbound RS232 data port. INLINE causes the unit to perform a direct connection between I/O 1 and I/O 2. Default setting is COMMAND.
- **I/O *n*** for all other serial I/O ports is set to DATA and cannot be changed.

Inline Mode Handshaking (Port 2 only) toggles the handshaking method used during Inline mode operation. Available options are XON/XOFF, DTR, and Both. Default setting is XON/XOFF.

Strip Sent Pass-Through LFs is an ON/OFF toggle to enable the stripping of linefeeds on pass-through data *sent out* of the T850. Default setting is OFF.

Strip Received Pass-Through LFs is an ON/OFF toggle to enable the stripping of linefeeds on pass-through data *received* by the T850. Default setting is OFF.

Disable Serial Setup via DIP Switch (Port 2 only) is an ON/OFF toggle to disable the DIP switches. Default setting is OFF.

Multiline Record Settings displays the Multiline Record Settings menu.

Data Type toggles between ASCII and Binary to indicate the type of data being collected on this port. Default setting is ASCII.

Change ETX to CR/LF is an ON/OFF toggle to set whether ETX characters in the incoming data should be converted to CR/LF characters. Default setting is OFF.

Multiline Record Settings

```
TeleBoss 850 - Serial Port 1 Multiline Record Settings
A) Multiline Record Enable               [OFF]
B) Blank Line Count                      [0]
C) Complex Multiline Detection           [OFF]
```

The T850 has the ability to monitor incoming serial data for multi-line records (individual records that are broken into multiple lines with carriage returns). If the records are separated by a specific number of blank lines, this basic configuration menu will suffice. If a more complex delineation scheme is used, enable Complex Multiline Detection.

Multiline Record Enable is an ON/OFF toggle to enable multiline record detection. Default setting is OFF.

Blank Line Count sets the number of blank lines that must come between records. Default setting is 0.

Complex Multiline Detection displays settings for detecting more complex multiline records. Default setting is OFF.

```
TeleBoss 850 - Serial Port 1 Complex Multiline Record Settings
A) Complex Multiline Record Enable      [OFF]
B) Start Field 1 Character Position     [0]
C) Start Field 1 Text                   []
D) Start Field 2 Character Position     [0]
E) Start Field 2 Text                   []
F) Collect Lines Before Start Record    [0]
G) End Detection                        [FORMULA]
H) Line Count                           [0]
I) End Field 1 Character Position       [0]
J) End Field 1 Text                     []
K) End Field 2 Character Position       [0]
L) End Field 2 Text                     []
```

Complex Multiline Record Enable is an ON/OFF toggle to enable advanced multiline detection. Default setting is OFF.

Start Field *n* Character Position sets the character position used to define the beginning of the multiline field. This option is used with "Count" method record end detection.

Start Field *n* Text sets the text used to determine the beginning of the multiline field. This option is used with "Formula" method record end detection.

Collect Lines Before Start Record sets the number of blank lines that are between each record.

End Detection toggles between FORMULA, COUNT, and BLANKS to set the method of detecting the end of each record. Default setting is FORMULA.

Line Count is the number of lines to meter each record at. This option is used with "BLANKS" record end detection.

End Field *n* Text/Character Position is the counterpart to start the text or character position option. This option sets the end delimiter for multiline records.

Modem Settings

```
TeleBoss 850 - Modem Settings
A) Dialup Modem
B) Wireless Modem
```

The Modem Settings menu displays two sub-menus for configuring either the internal 56K modem, or a optional wireless modem expansion card.

Dialup Modem

```
TeleBoss 850 - Dialup Modem Settings
A) Data Format                          [8N1]
B) Duplex                               [FULL]
C) Init String                          [ATM1]
D) Inactivity Timeout                   [0]
E) Upon Modem Connect Go Directly To    [LOGIN]
F) TAP Init String                      [ATM0]
G) TAP Uses 8N1 Data/Parity/Stop        [0]
H) Caller ID Security                   [OFF]
```

Note: If the optional 56K dialup modem is not installed in the T850, this menu is displayed, but changing any of the settings will not do anything.

Data Format toggles settings for word length, parity, and stop bit settings. The available options are: 8N1, 7E1, 7O1, and 7N1. Default setting is 8N1.

Duplex controls the echo settings for the modem command processor. Full duplex causes the T850 to echo all characters sent to the remote device. Half duplex turns off character echo. Default setting is FULL.

Note: If Duplex is set to Half, set **`sys.terminal.mode`**`=OFF`. Otherwise, characters will continue to be echoed to the terminal screen.

Init String sets the user-defined modem initialization string. This string is sent to the modem before important factory modem initialization settings, so certain settings in this init string may be overridden. Default setting is ATM1. (Max length 126 chars) Note: Make sure to enter 'AT' at the beginning of this initialization string.

Inactivity Timeout sets the number of minutes (0 – 255) to wait before disconnecting an idle modem connection. A setting of 0 means the connection will never automatically expire. Default setting is 0.

Upon Modem Connect Go Directly To toggles through a list of actions to control what a user sees directly after connecting via modem. LOGIN requires the user to login with username and password, and will then take them to a command prompt. A serial port (I/O1, I/O2, etc.) redirects a modem user directly to that serial port upon connecting. In this passthrough mode, the command processor of the T850 is transparent. Default setting is LOGIN.

TAP Init String is the user-defined modem initialization string used only when the modem is making an alphanumeric modem callout. Default setting is ATM0. (Max length 126 chars) Note: Make sure to enter 'AT' at the beginning of this initialization string.

TAP Uses 8N1 Data/Parity/Stop toggles between 1, to force the TAP initialization string data/parity/stop settings to 8N1, and 0 to not force this setting. Default setting is 0.

> **Setting Key:** **`modem.hsk`**
> Values are `RTS` (default), `None` and `Xon`. RTS means that on serial pass-through, the modem uses RTS handshaking; None means no handshaking is used; and Xon means XON/XOFF characters are used.

Caller ID Security displays a menu that allows you to configure from one to twenty inbound phone numbers to restrict modem access.

Caller ID Security

```
TeleBoss 850 - Caller ID Security
A) Enable                               [OFF]
B) Caller ID 1                          []
   ...
U) Caller ID 20                         []
V) Add Number From Log List
```

Note: Caller ID must be available on the phone line connected to the T850 for this feature to work.

Enable is an ON/OFF toggle to enable caller ID restrictions. When enabled, the T850 will only answer the modem if caller ID indicates one of the allowed phone numbers is connecting. Default setting is OFF.

Caller ID *n* allows you to add or change a specific phone number. You are allowed to use simple wildcards in phone numbers: An asterisk (*) wildcard allows for any number of digits to appear to the right of that position. A question mark (?) matches any single digit. If no numbers are defined in this menu, all incoming calls are accepted. (Max length 47 chars)

Add Number From Log List displays a list of phone numbers that have recently dialed into the T850 for addition to this list.

Wireless Modem

```
TeleBoss 850 - Wireless Modem Settings
A) Mode                                [OFF]
B) APN                                 []
C) PIN                                 []
D) Idle Timeout (minutes)              [5]
E) Band (GPRS only)                    [DUAL-850/1900]
F) PPP/Wireless User Name              []
G) PPP/Wireless Password               [********]
H) Default Route Enable                [OFF]
```

Note: If the optional wireless modem expansion card is not installed in the T850, this menu is displayed, but changing any of the settings will not do anything, except for the PPP/Wireless User Name and Password settings (see below).

Note: For a complete description of the setup and operation of the wireless modem, please refer to the Wireless Modem chapter later in this manual.

Mode toggles between OFF (disable modem), PERMANENT (maintain "always-on" connection with modem), and CIRCUIT–SWITCHED. Default setting is OFF.

APN sets the Access Point Name as defined by your wireless provider. Default setting is " ". (Max length is 31 chars)

PIN sets the PIN associated with the SIM card (if any). Default setting is " ". (Max length is 15 chars)

Idle Timeout sets the number of minutes (3 – 255) to wait before disconnecting an inactive modem connection. The purpose of this setting is to allow the modem to get reset after a period of inactivity to ensure the modem connection is working properly. Default setting is 5 minutes.

Band (GPRS only) toggles between DUAL - 850/1900, DUAL – 900/1800, DUAL – 900/1900, MONO – 850, MONO – 900, MONO-1800, and MONO – 1900. This sets the GSM bands on which the modem will operate. Default setting is DUAL - 850/1900.

Note: This setting is only used with the GPRS modem. For this setting to take effect, the wireless modem must be reset; this can be accomplished by restarting the host unit, or by setting the wireless modem mode to OFF for at least 10 seconds, then back to a GPRS setting.

PPP Wireless User Name / Password sets the login credentials for the PPP connection. These settings are identical to the same settings in the PPP Dialout Settings menu– so a change in one menu will change the settings in the other. (Max length for each is 64 chars)

Default Route Enable is an ON/OFF toggle to enable the wireless interface to be the default route when connected. Default setting is OFF.

Security Settings

```
TeleBoss 850 - Security Settings
A) Security Mode                       [USER PROFILES]
B) Specific Security Settings
C) General Security Settings

Enter your Selection:
```

The Security Settings menu displays options for setting the security mode, as well as specific and general security settings.

Security Mode toggles between USER PROFILES and RADIUS to determine which Specific Security Settings menu to be displayed.

Specific Security Settings menu is determined by toggling Security Mode. USER PROFILES causes option B) Specific Security Settings to display the User Profile Security Settings menu where twelve individual User Profiles can be configured along with Authentication Settings. RADIUS causes option B) Specific Security Settings to display the RADIUS Security Settings menu where RADIUS authentication server settings can be configured. Default setting is USER PROFILES.

General Security Settings displays a menu with options that apply to ***every*** user of this T850.

Specific Security Settings – User Profile Security Settings

```
TeleBoss 850 - User Profile Security Settings
A) User 1: User1/********/COMMAND/FILE1
B) User 2: User2/********/COMMAND/FILE1
C) User 3: User3/********/PASSTHROUGH/FILE1
D) User 4: User4/********/PASSTHROUGH/FILE2
E) User 5:
F) User 6:
G) User 7:
H) User 8:
I) User 9:
J) User 10:
K) User 11: admin/********/COMMAND/FILE1
L) User 12:
M) Authentication Settings
```

User n displays the configuration menu for each user profile.

Note: User 11 is preconfigured in the unit, for use by SitePath.

Authentication Settings displays a menu of global authentication options.

Note: Passwords are case sensitive and are masked in all menus and while typing them from the command line, for security reasons. If a user without permissions accesses the User Profile Settings menus, they will see all fields in this menu either masked or with no data in them. If they select an option, a message will be displayed that says: “You do not have permission to change this setting.”

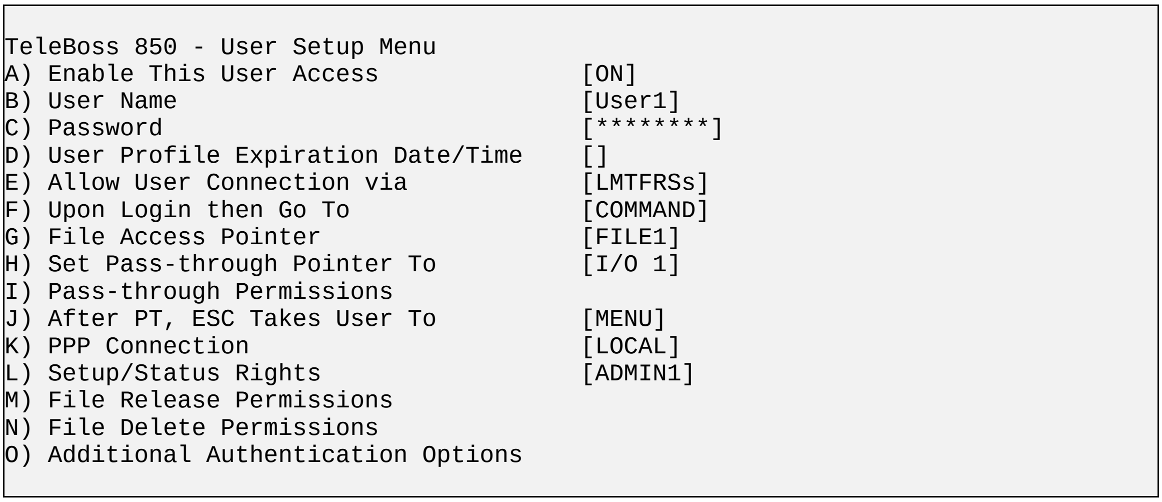
Note: When configuring a new username, and an invalid or duplicate username is entered, the T850 responds as follows:

```
Invalid Entry.
Press any key to continue...
```

Note: When configuring a new password, the T850 will ask you to re-enter the password. If the second entry of the password does not match the first, the T850 responds as follows:

```
Invalid Entry - Confirm Password does not match.
Press any key to continue...
```

User Setup Menu



```
TeleBoss 850 - User Setup Menu
A) Enable This User Access              [ON]
B) User Name                            [User1]
C) Password                             [********]
D) User Profile Expiration Date/Time    []
E) Allow User Connection via            [LMTFRSs]
F) Upon Login then Go To                [COMMAND]
G) File Access Pointer                  [FILE1]
H) Set Pass-through Pointer To          [I/O 1]
I) Pass-through Permissions
J) After PT, ESC Takes User To          [MENU]
K) PPP Connection                       [LOCAL]
L) Setup/Status Rights                  [ADMIN1]
M) File Release Permissions
N) File Delete Permissions
O) Additional Authentication Options
```

Enable This User Access is an ON/OFF toggle to enable access for this user profile.

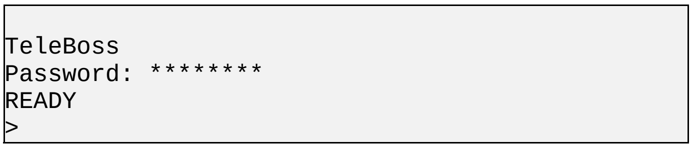
User Name / Password sets the username and/or password for this profile. (Max length for each is 31 chars)

User Profile Expiration Date/Time sets a date and/or time that this profile may be automatically disabled. This also provides an option to adjust the current date/time that is on the T850. Selecting that option will transfer you to the System Date/Time menu. If left blank, this user profile will not expire. Default setting is blank.

Allow User Connection via displays a menu allowing you to toggle ON or OFF access via Local (Console Port), Modem, Telnet, FTP, Real-Time Socket, and SSH (Secure Shell). These are abbreviated: LMTFRSs and default setting for all is ON.

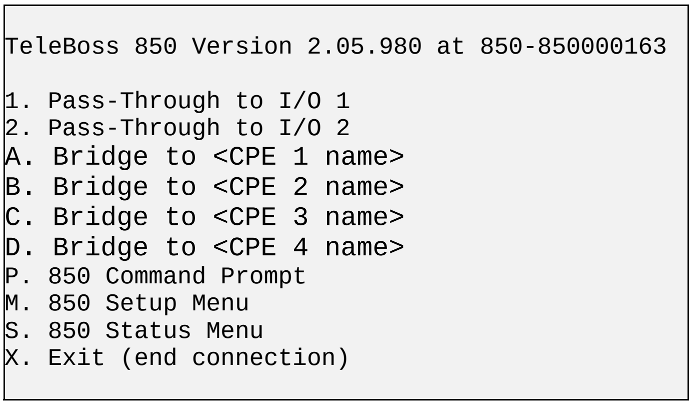
Upon Login then Go To toggles the action this user will be directed to upon logging in, with the following options: Command, Menu, and Passthrough as shown here:

Command



```
TeleBoss
Password: ********
READY
>
```

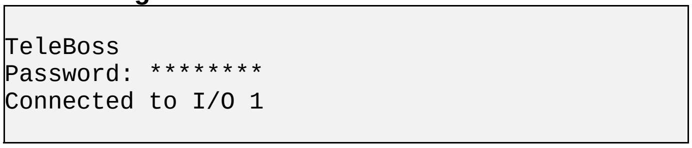
Menu



```
TeleBoss 850 Version 2.05.980 at 850-850000163

1. Pass-Through to I/O 1
2. Pass-Through to I/O 2
A. Bridge to <CPE 1 name>
B. Bridge to <CPE 2 name>
C. Bridge to <CPE 3 name>
D. Bridge to <CPE 4 name>
P. 850 Command Prompt
M. 850 Setup Menu
S. 850 Status Menu
X. Exit (end connection)
```

Passthrough



```
TeleBoss
Password: ********
Connected to I/O 1
```

File Access Pointer toggles through each of the data files on the unit (FILE1, FILE2, etc) to set what data file the user has access to when logging in to the command processor.

Set Pass-through Pointer To is in effect if the “Upon Login then Go To” action is set to Passthrough. Whatever this option is set for determines where this user will be routed to on a pass-through connection. This option toggles the serial port (I/O 1, I/O2, etc). Or, it can be toggled to any one of the four CPE devices (CPE1 thru CPE4). If the user is connects and logs in via an SSH connection, and the CPE Settings/SSH to Telnet Bridging option for that CPE device is set to ON, then the user will be automatically bridged to that CPE. If the user is not authorized to connect to that CPE then a message will be output saying as much, and the user will be disconnected. Default setting is FILE1.

Pass-through Permissions is in effect if the “Upon Login then Go To” action is set to Menu. This option displays a menu showing all serial ports and CPE devices 1 thru 4, and toggles ALLOW or DENY for each as needed. If a port or CPE device is set as ALLOW, then that serial port is displayed in the Menu after the user logs in. If a port or CPE device is set as DENY, then it is not displayed in the Menu. Default setting for all ports is ALLOW.

Note: If a CPE device name is defined in the CPE Settings menu, then that name will be displayed in the Login Menu when set to ALLOW. If a name is not defined, then the IP address of the device will be displayed.

After PT, ESC Takes User To sets the action this user can perform when they exit out of a pass-through connection.

PPP Connection toggles between LOCAL, ROUTING and NONE. LOCAL allows PPP access, but denies all routing to whatever LAN the T850 is connected to. ROUTING enables Route Ethernet to PPP and Route PPP to Ethernet for the user, but only if those settings are enabled globally. NONE disables PPP access for the user.

Setup/Status Rights toggles through the actions available to the user if they are given access to the command prompt. Options are MASTER, NONE, VIEW, ADMIN1, ADMIN2, and ADMIN3. See the User Rights Table for more information on each access level. Default setting is MASTER.

File Release / Delete Permissions displays a menu showing all data files, Events Log and Audit Log, and toggles ALLOW or DENY for each as needed. Default setting for all is ALLOW.

Additional Authentication Options displays extra-high security options.

```
TeleBoss 850 - Additional Authentication Options
A) Secure Authentication via Telnet             [OFF]
B) For Telnet, Send Password To                 []
C) Secure Authentication via Modem              [OFF]
D) For Modem, Send Password To                  []
E) Secure Authentication via Local Command Port [OFF]
F) Password Expires After                       [30]
G) Secure Callback 1                            []
H) Secure Callback 2                            []
I) Secure Callback 3                            []
```

Secure Authentication via Telnet/Modem toggles between OFF (regular), CHALLENGE, and SEND PASSWORD authentication modes. Default setting for each is OFF.

OFF (regular) authentication requires only the normal username/password authentication.

CHALLENGE requires the user send their username/password and then they are prompted with a short challenge code. That code must be plugged into a program called Response Code Generator (RCG). This software can be found on the Documentation and Utilities CD. Contact Asentria for more information on how to use or obtain this application. RCG requires a shared secret as well as the challenge code generated by the T850. The user must then respond with the proper hash generated by RCG in order to gain access.

SEND PASSWORD will generate a single-use password and send it to the Email address(es) specified by the next option. That password will only allow a login for the user whom it was generated for.

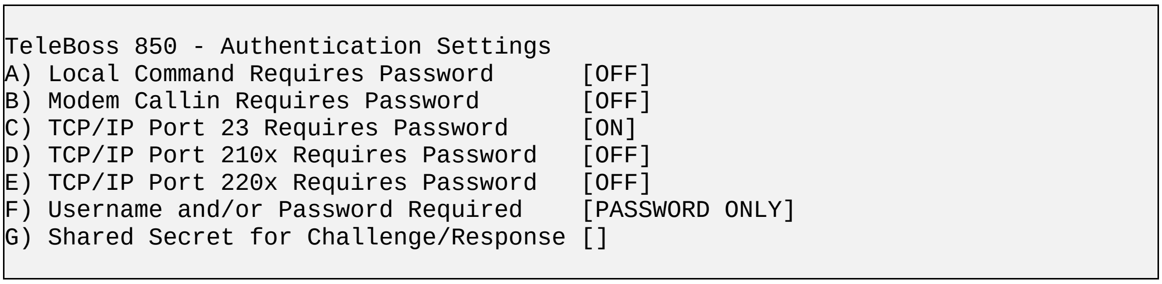
For Telnet/Modem, Send Password To sets the Email address(es) where the single-use password is to be sent.

Secure Authentication via Local Command Port toggles between OFF (regular), and CHALLENGE. Because the user is connected via the local Console port, Send Password is not an option. Default setting is OFF.

Password Expires After sets the number of minutes (0 – 180) before the single-use password expires. A setting of 0 means the password will never automatically expire. Default setting is 0.

Secure Callback *n* sets the modem callback numbers. If configured, the T850 will disconnect any modem connections from this user and then attempt to dial out to each of these numbers. If one of the numbers answers, the other end must respond with the login credentials of the user used to initiate the callback. (Max length 48 chars)

Authentication Settings



```
TeleBoss 850 - Authentication Settings
A) Local Command Requires Password      [OFF]
B) Modem Callin Requires Password       [OFF]
C) TCP/IP Port 23 Requires Password     [ON]
D) TCP/IP Port 210x Requires Password   [OFF]
E) TCP/IP Port 220x Requires Password   [OFF]
F) Username and/or Password Required    [PASSWORD ONLY]
G) Shared Secret for Challenge/Response []
```

Authentication Settings set parameters for passwords and security that are required for ***every*** user who attempts to log into the T850.

Local Command Requires Password is an ON/OFF toggle to set whether a password for I/O2 users is required. Default setting is OFF.

Modem Callin Requires Password is an ON/OFF toggle to set whether a password for modem users is required. Default setting is OFF.

TCP/IP Port 23 Requires Password is an ON/OFF toggle to set whether a password for Telnet (port 23) users is required. Default setting is ON.

TCP/IP Port 210x Requires Password is an ON/OFF toggle to set whether a password for passthrough (port 210x) users is required. Default setting is OFF.

TCP/IP Port 220x Requires Password is an ON/OFF toggle to set whether a password for Real-Time Socket (port 220x) users is required. Default setting is OFF.

» **Note:** When any of the above options is set to OFF, users connecting via that method are automatically granted all access.

Username and/or Password Required toggles between: PASSWORD ONLY, USERNAME/PASSWORD (PW), or PASSWORD(PW)/USERNAME. Default setting is PASSWORD ONLY.

Shared Secret for Challenge/Response sets the shared secret used to generate Challenge/Response codes. (Max length 48 chars). Challenge/Response requires the use of the free Asentria Response Code Generator program. Contact Asentria Technical Support for this, or download (named “Password Generator”) from the Product Resources page on the Asentria website: http://www.asentria.com/docsandsoftware/productmanuals.aspx

Specific Security Settings – RADIUS Security Settings

```
TeleBoss 850 - RADIUS Security Settings
A) Primary Server                         []
B) Primary Secret                         []
C) Secondary Server                       []
D) Secondary Secret                       []
E) Fallback Mode                          [NONE]
F) Authentication Port                    [1812]
G) Accounting Port                        [1813]
H) CHAP                                   [OFF]
I) Timeout                                [3]
J) Retries                                [3]
```

Primary / Secondary Server sets the IP Address or host name of the primary and secondary RADIUS server.

Primary / Secondary Secret sets the secret for the primary and secondary RADIUS server. The secret is used to authenticate RADIUS network traffic. (Max length for each is 16 chars)

Fallback Mode toggles between NONE and USER PROFILES. If the unit gets no response from any RADIUS server when attempting to authenticate a user, no further action is taken if this option is set to NONE. The unit falls back to the User Profiles configuration for authentication if this is set to USER PROFILES. Default setting is NONE.

Authentication Port sets the UDP port (1 – 65535) that the RADIUS server uses for authentication/authorization. Default port is 1812.

Accounting Port sets the UDP port (1 – 65535) that the RADIUS server uses for accounting traffic. Set to 0 to disable RADIUS accounting. Default port is 1813.

CHAP is an ON/OFF toggle to set whether the unit uses CHAP (Challenge-Handshake Authentication Protocol) authentication when using RADIUS. ON sets authentication to CHAP. OFF sets authentication to PAP (Password Authentication Protocol). Default setting is OFF.

Timeout sets the number of seconds (1 – 30) the unit waits for a response from the RADIUS server. Default setting is 3.

Retries sets the number of times (1 – 30) the unit should try a RADIUS request again after getting no valid response. (Valid meaning a response that is verified as really coming from the RADIUS server.) Default setting is 3.

Note: For a complete description and explanation of RADIUS security, please refer to the RADIUS Security section in the Features chapter.

General Security Settings

```
TeleBoss 850 - Global Password/Security Settings Menu
A) Show Username/Password Prompt          [OFF]
B) Globally Allow Access via              [MTFRSs]
C) Button Tap Allows Console Access       [ON]
```

Global Password/Security Settings set security options that are required for ***every*** user who attempts to log into the T850.

Show Username / Password Prompt is an ON/OFF toggle to set whether a prompt for logging in is displayed. Default setting is OFF.

Globally Allow Access via displays a menu allowing you to toggle ON or OFF access via Modem, Telnet (ports 23, 200x, 210x), FTP, Real-Time Socket, and Secure Shell (SSH). These are abbreviated: MTFRSs. Default setting for all is ON.

Button Tap Allows Console Access is an ON/OFF toggle to give access to a user who has forgotten their log on credentials. This is an insurance policy against locking yourself out of the unit. When set to ON, someone local to the unit can tap the Reset button 5 times quickly (1-2 times per second), at which point the front-panel LEDs will flash briefly for several seconds, giving the user immediate Console access using the default MASTER username and password. Refer to the Securing a TeleBoss 850/Button Unlock section for more details about this. Default setting is ON.

Alarm / Event Definitions

Note: Refer to the Data Events section in the Features chapter for an example-driven approach to defining alarm definitions.

```
TeleBoss 850 - Alarm/Event Definitions Menu
A) Class Table
B) Data Alarm/Filter Settings
C) EventSensor Device Settings
D) No-Data 1 Alarm Settings              [OFF]
E) No-Data 2 Alarm Settings              [OFF]
F) Percent Full Alarm Settings           [OFF]
G) Scheduled Event 1 Settings            [OFF]
H) Scheduled Event 2 Settings            [OFF]
I) IPRC Alarm Settings                   [OFF]
J) Serial Handshaking Alarm Settings
K) CPE Alarm Settings
L) Data Filter Action                    [REJECT]
M) Event Message Settings
```

Class Table displays the menu for configuring event classification settings.

Data Alarm/Filter Settings displays the menus for configuring serial data event monitors.

EventSensor Device Settings displays the menus for configuring internal and external sensors and modules that may be installed and/or connected to I/O 1 using the Serial to ESBus Adapter. (Contact Asentria Technical Support for more details).

No-Data *n* Alarm Settings displays the menus for configuring alarms based on period of time when no-data is received on a specific serial port.

Percent Full Alarm Settings displays the menu for configuring alarms based on how full the call record database of the T850 is.

Scheduled Event *n* Settings displays the menus for configuring alarm notifications for specific times and days of the week.

IPRC Alarm Settings displays the menu for configuring alarms for a lost IP Record Collection connection.

Serial Handshaking Alarm Settings displays the menu for enabling serial handshaking alarms for specific ports.

CPE Alarm Settings displays the menu for configuring “CPE Down” events. These are used in conjunction with devices managed by the Asentria SitePath application.

Data Filter Action toggles between REJECT and ACCEPT to indicate whether data filters are configured to reject or accept specific incoming data string(s). Default setting is REJECT.

Event Message Settings displays the menu that permits customization of the event message that appears in traps, Emails, pages, etc.

Class Table

```
TeleBoss 850 - Class Table
A) Class 1                                           [Info]
B) Class 2                                           [Minor]
C) Class 3                                           [Major]
D) Class 4                                           [Critical]
E) Class 5                                           []
   ...
L) Class 12                                          []
```

Class *n* defines the event classification assignable to events detected by the T850. (Max length 47 chars)

Info, Minor, Major, and Critical are the default class names assigned to the first four classes. These can be changed and others added as desired to meet your specific needs.

The class number and name are reported in Asentria Alarms, and SNMP traps. It is a mechanism for you to provide varying severities for different alarms so that you can act on them upon receipt.

Data Alarm/Filter Settings

```
TeleBoss 850 - Data Alarm/Filter Settings
A) Data Alarm Field Settings
B) Data Alarm Macro Settings
C) Data Alarm Settings
D) Display Alarm Status
E) Exit Upon True Data Alarm            [OFF]
```

Data Alarm Field Settings displays the menu for configuring up to 16 data alarm fields.

Data Alarm Macro Settings displays the menu for configuring up to 100 macros to be used for data alarming.

Data Alarm Settings displays the menu for configuring up to 1000 data alarms or filters.

Display Alarm Status displays real time information on data event monitors you've configured.

Exit Upon True Data Alarm is an ON/OFF toggle to set whether the T850 will stop processing more data event evaluations on a single record after it has found one match. This should be disabled if it is possible to have more than one event in a record. This is a global setting – it applies to ALL configured data alarms. Default setting is OFF.

Data Alarm Field Settings

```
TeleBoss 850 - Data Alarm Field Definition Table
                     Start    Length   Line     Type       Name
A) Definition A      0        0        0        [Alpha]
B) Definition B      0        0        0        [Alpha]
   ...
P) Definition P      0        0        0        [Alpha]

Enter your Selection: a

TeleBoss 850 - Data Alarm Field Definition
Data Field: A
A) Start Position                       [0]
B) Field Length                         [0]
C) Field Name                           []
D) Field Line Number                    [0]
E) Field Type                           [Alpha]
```

Start Position sets the number of the characters to begin a particular alarm field starting from position 1. Field definition is disabled if set to 0.

Field Length sets the length of this particular alarm field.

Field Name sets the name given for the alarm field. This name must be unique, is limited to 12 characters, and it must not contain any spaces. It can contain alphanumeric characters and the underscore, but it must start with a letter. These field names are case sensitive. If left blank, you can refer to the field by it's field letter (A,B, etc…).

Note: To avoid naming conflicts, the T850 does not allow duplicate field names. The T850 will respond with "Invalid Entry, Press any key to continue" if a duplicate field name is entered.

Field Line Number sets the optional line number the field should be limited to in multiline records.

Field Type toggles between Alpha and Numeric. Alpha is used for most alphanumeric data alarming, and Numeric is used if you need to alarm on a range of numbers. Default setting is Alpha.

Data Alarm Macro Settings

```
TeleBoss 850 - Data Alarm Macro Settings
A) Macro 1                          []
B) Macro 2                          []
   ...
P) Macro 16                         []
Q) Next Macro Page

Enter your Selection: a

TeleBoss 850 - Settings for Data Alarm Macro 1
A) Name                             []
B) Equation                         []
```

Data alarm macros provide a way to define up to 100 equations that can be used in one or more data alarm equations. Each macro consists of an equation and an associated name that can be used to reference the macro in a data alarm equation. Refer to the Data Alarm Macros section in the Features chapter for more information.

Data Alarm/Filter Settings

```
TeleBoss 850 - Data Alarm/Filter Settings
A) Alarm/Filter Page 1 (Alarms 1-16)
B) Alarm/Filter Page 2 (Alarms 17-32)
   ...
P) Alarm/Filter Page 16 (Alarms 241-256)
Q) Next Page Selection Screen
```

Data alarms are configured by selecting an option from the main Data Alarm/Filter Settings menu, then selecting one of the options which will give you a group of 16 data alarm/filters (1-16, 17-32, etc) or selecting the Next or Previous Page Selection Screen. This will display a menu where you can select from those 16 data alarm options as follows:

```
TeleBoss 850 - Data Alarm/Filter Settings
A) Alarm/Filter 1           []                       [OFF]  [ALARM]
   ...
P) Alarm/Filter 16          []                       [OFF]  [ALARM]
Q) Next Alarm/Filter Page
R) Setup Alarm/Filter Fields
S) Display Alarm Status
T) Exit Upon True Data Alarm                         [OFF]

Enter your Selection:
```

Alarm/Filter n displays the menu where an individual data alarm or filter can be configured.

Next or Previous Alarm/Filter Page displays either the next or previous set of 16 Data Alarm/Filters.

Setup Alarm/Filter Fields displays the identical Data Alarm Field Setting menu as described above. This is simply an easy way to access that menu without having to exit back through the previous menus.

Display Alarm Status displays real time information on data event monitors you've configured.

Exit Upon True Data Alarm is an ON/OFF toggle to set whether the T850 will stop processing more data event evaluations on a single record after it has found one match. This should be disabled if it is possible to have more than one event in a record. This is a global setting – it applies to ALL configured data alarms. Default setting is OFF.

Data Alarm/Filter *n* Settings

```
TeleBoss 850 - Settings For Data Alarm/Filter 1
A) Alarm/Filter Enable                  [OFF]
B) Alarm/Filter Mode                    [ALARM]
C) Alarm/Filter Name                    []
D) Alarm/Filter Equation                []
E) Threshold                            [1]
F) Auto-Clear when Threshold Reached    [ON]
G) Alarm Counter Clear Interval         [12 HOURS]
H) Alarm Counter Reset Time             [00:00]
I) Actions                              []
J) Class                                [Info]
K) Data Alarm Trap Number               [503]
L) Clear This Alarm Counter Now
```

Alarm/Filter Enable is an ON/OFF toggle for each individual data event monitor. Default setting is OFF.

Alarm/Filter Mode toggles between Alarm and Filter to indicate whether the T850 will recognize this data event as an Alarm and take some action, or as a Filter and either accept or reject the data string. Default setting is ALARM.

Alarm/Filter Name sets the name for the event monitor. This name is reported with the specified actions. (Max length 16 chars)

Alarm/Filter Equation defines the event equation using the event fields defined in the previous menu. (Max length 160 chars) Refer to the Configuring Data Alarm Equations section in the Features chapter for more information.

Threshold sets the number of times the event equation must be matched before an event is triggered. If the event counter is allowed to grow beyond the threshold, the unit will not trigger an event again until after the counter is reset. Default setting is 1.

Auto-Clear when Threshold Reached is an ON/OFF toggle to control whether the unit will clear the event counter each time the threshold is met. Default setting is ON.

Alarm Counter Clear Interval sets an interval at which the unit should clear the match counter for an individual data event. Available options are: 2 hours, 4 hours, 6 hours, 8 hours, 12 hours, Daily, and Never. The first clear occurs at midnight. Default setting is 12 Hours.

Alarm Counter Reset Time sets the time at which the daily clear should take place if it is enabled in the Alarm Counter Clear Interval. This value is in 24-hour format. Default setting is 00:00.

Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured.

Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this data alarm.

Data Alarm Trap Number sets the number to be sent with any SNMP traps for this event. Default is 503, but trap number can also be set in the range of 1000 – 1199 as needed.

Clear This Alarm Counter Now allows you to clear the counter for the selected data alarm manually. This happens as soon as this option is selected, so make sure you really want to clear the counter before selecting it.

Actions List

```
Enter one or more actions using this format:
(For more details see the users manual)
----------------------------------------------
Cancel : cancel(idname)
Dialup Pager : dpage(index)
Dispatcher : dispatch(phone# or index)
Email : email(email or index)
Group : group(groupname)
ID : id(id name)
Inform : inform(ipaddress or index)
Malert : malert(phone# or index)
Modem : modem(phone# or index)
Postpone : postpone(idname, seconds)
Pause : pause(seconds)
Relay : relay(action, eventsensor, point)
Script : script(action, name or number)
SMS : sms(phone# or index)
Talert : talert(ipaddress or index)
Trap : trap(ipaddress or index)
Stop if any/all actions OK : okstop(any|all)
Continue: continue(id)
(separate multiple actions using semicolon)

Current Actions:
Enter Actions:
```

The Actions List provides you with a flexible mechanism to tell the unit how to react to events. An action is a text string that specifies what the unit should do upon an event. It's comprised of a list of keywords and parameters separated by semicolon. Each keyword specifies a certain action and has its own parameter set, which is enclosed in parentheses. Refer to Action List in the Features chapter for more information.

EventSensor Device Settings

The T850 supports a wide variety of internal and external sensor devices and relays, including contact closures, temperature and humidity sensors, analog voltage sensors, and relays. For the purposes of clarity, all of these will be generally referred to as "EventSensors" (ES) unless a specific type of sensor or relay is being described.

The Sensor Events Menu is used to configure and control both internal and external sensors and relays. If you don't have any internal sensors or relays, or remote ES devices connected, this menu will be unpopulated. Because of the numerous ES configurations possible, menus shown in this section probably will not look exactly like the ones for your T850. (The menu below shows a T850 with on-board 8 contact closures.)

```
TeleBoss 850 - Sensor Events Menu
   Name                 ID          Alive       Number      Configuration
A) INTERNAL             --------    -           200         2-CC
B) unnamed              0301F5C4    Y           1           8-CC
C)  <none>
   ...
Q)  <none>
R) Sensor Unresponsive Settings
```

The T850 supports a maximum of 16 external EventSensor slots. Some larger EventSensors occupy more than one slot. For example, the ES-CCU32 requires two slots and the ES-CCU64 requires four.

EventSensor Slots (A thru P) displays the settings menu for each ES.

Sensor Unresponsive Settings displays the Sensor Unresponsive Menu where you can configure the actions the T850 takes if an ES becomes unresponsive.

EventSensor Slots

```
TeleBoss 850 - Internal Events Menu
A) Device Name                                    [INTERNAL]
B) Contact Closure 1                              [unnamed]
C) Contact Closure 2                              [unnamed]
D) EventSensor Reporting Enabled                  [OFF]
```

The display for each ES will vary depending on configuration. For example, an ES could be either internal or external. EventSensors can be configured with varying combinations of the I/O types. Refer to the Event Sensor Configuration Setup section in the Features chapter that can be referred to for more information.

EventSensor Reporting Enabled is an ON/OFF toggle to enable the Event Sensor Reporting feature. See the Event Sensor Reporting section in the Features chapter for more information

Contact Asentria for more information on obtaining Expansion Cards or Type2 EventSensors for use with the T850. See the EventSensor documentation for more information about configuring a specific ES device.

Sensor Unresponsive Settings

```
TeleBoss 850 - Sensor Unresponsive Menu
A) Sensor Unresponsive Timeout                     [30]
B) Sensor Unresponsive Actions                     []
C) Sensor Unresponsive Trap Number                 [50]
D) Sensor Unresponsive Class                       [Info]
```

Sensor Unresponsive Timeout sets the time (10 - 65535 seconds) to wait before declaring a non-communicative EventSensor unresponsive. Default setting is 30.

Sensor Unresponsive Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Sensor Unresponsive Trap Number sets the number to be sent with any SNMP traps for this event. Default is 50, but trap number can also be set in the range of 1000 – 1199 as needed.
Sensor Unresponsive Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

No-Data *n* Alarm Settings

No Data Alarms can be configured on the T850 to monitor data coming in via the serial ports, and take an alarm action if a certain period of time passes with no data.

```
TeleBoss 850 - No-Data Alarm 1 Settings
A) Alarm Enable                        [OFF]
B) Alarm Actions                       []
C) Alarm Message                       [No-Data Timeout 1]
D) Alarm Class                         [Info]
E) Trap Number                         [505]
F) Schedule 1 Begin Time               [00:00]
G) Schedule 1 End Time                 [00:00]
H) Schedule 1 Duration (minutes)       [0]
I) Schedule 2 Begin Time               [00:00]
J) Schedule 2 End Time                 [00:00]
K) Schedule 2 Duration (minutes)       [0]
L) Apply Alarm on Days                 [MTuWThF]
M) Enable Ports
N) Add Exclusion
O) Delete Exclusion
   []
   []
```

No-Data *n* Alarm Settings allows you to configure two separate No-Data Alarms, each of which can be configured for two different ranges of times with different time durations. The periods of time should be configured to match the calling patterns of your business or organization. For example, if your normal business hours are M-F 8:00 to 5:00, you will want to set lower time durations during those hours than you would “after hours” when call volumes are lighter and the periods of time where there is "no data“ might be longer.

Alarm Enable is an ON/OFF toggle to enable the no-data monitor. Default setting is OFF.

Alarm Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Alarm Message sets the text string to be delivered with this event’s alarms. Default setting is "No-Data Timeout *n*“. (Max length 126 chars)

Alarm Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Trap Number sets the number to be sent with any SNMP traps for this event. Default is 505, but trap number can also be set in the range of 1000 – 1199 as needed.

Schedule *n* Begin Time / End Time sets the beginning and ending times (24 hr clock) for each of two ranges of time.

Schedule *n* Duration is the number of minutes (0-65535) the unit will wait without receiving data before alarming.

Apply Alarm on Days displays a menu where the seven days of the week are listed, and each can be toggled ON or OFF to designate whether this particular No-Data alarm is active on that day. Default setting is ON for Monday thru Friday, and OFF for Saturday and Sunday.

Enable Ports displays a menu where the installed serial ports are listed and each can be toggled ON or OFF to designate whether this particular No-Data alarm is active on that port. Default setting is OFF for all ports.

Add Exclusion / Delete Exclusion allow you to add or delete specific dates when this No-Data Alarm should “take the day off”. For example, Christmas is a day you might want to add here. Select Add Exclusion and type in **12/25**. To delete a date, you select Delete Exclusion and type in the date you want to remove. After an exclusion date is added it appears in the brackets at the bottom of the menu. 15 dates can be entered to be excluded.

Percent Full Alarm Settings

```
TeleBoss 850 - Percent Full Alarm Settings
A) Alarm Enable                        [OFF]
B) Percent Full Threshold              [80]
C) Alarm Actions                       []
D) Alarm Message                       [DB Exceeds Threshold]
E) Alarm Class                         [Info]
F) Trap Number                         [501]
```

Alarm Enable is an ON/OFF toggle to enable the percent full alarm. Default setting is OFF.

Percent Full Threshold set the percent full level at which the alarm will be triggered. Default setting is 80 percent.

Alarm Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Alarm Message sets the text string to be delivered with the percentage full alarm. Default setting is DB Exceeds Threshold. (Max length 111 chars)

Alarm Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this percent full alarm.

Trap Number sets the number to be sent with any SNMP traps for this event. Default is 501, but trap number can also be set in the range of 1000 – 1199 as needed.

Scheduled Event Settings

Scheduled Events allow you to schedule specific a specific date/time for an alarm action to occur. For example, you might want the T850 to send you an Email every morning at 8:00 just so you know it is live on the network.

```
TeleBoss 850 - Scheduled Event 1 Setup
A) Enable Event                        [ON]
B) Event Actions                       []
C) Event Message                       [Scheduled Event 1]
D) Event Class                         [Info]
E) Trap Number                         [506]
F) Event Time Sunday                   [OFF]
G) Event Time Monday                   [OFF]
H) Event Time Tuesday                  [OFF]
I) Event Time Wednesday                [OFF]
J) Event Time Thursday                 [OFF]
K) Event Time Friday                   [OFF]
L) Event Time Saturday                 [OFF]
M) Add Exclusion
N) Delete Exclusion
   []
   []
```

Scheduled Event *n* Setup allows you to configure two separate Scheduled Events, each of which can be configured for any one time on any day of the week. Each day's time can be scheduled independently from the others.

Enable Event is an ON/OFF toggle to enable the Scheduled Event. Default setting is OFF.

Event Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Event Message sets the text string to be delivered with this event's action. Default setting is "Scheduled Event *n*". (Max length 126 chars)

Event Class sets the class for the event. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Trap Number sets the number to be sent with any SNMP traps for this event. Default is 506, but trap number can also be set in the range of 1000 – 1199 as needed.

Event Time *day* sets the time (24 hour clock) each day at which the scheduled event action will occur. If no time is configured for any day, this menu displays OFF.

Add Exclusion / Delete Exclusion allow you to add or delete specific dates when this Scheduled Event should "take the day off". For example Christmas is a day you might want to add here. Select Add Exclusion and type in **12/25**. To delete a date, you select Delete Exclusion and type in the date you want to remove. After an exclusion date is added it appears in the brackets at the bottom of the menu. 15 dates can be entered to be excluded.

IPRC Alarm Settings

An IPRC alarm allows the T850 to monitor the IPRC connection and alert you if the connection is lost. This would be an indicator that the IP-enabled switch has failed, something has failed on the network connection between the T850 and the switch, or a number of other reasons depending on the device.

```
TeleBoss 850 - IPRC Alarm Settings Menu
A) Connection Lost Alarm Enable         [OFF]
B) Connection Lost Timeout              [60]
C) Connection Lost Alarm Actions        []
D) Connection Lost Alarm Message        [Connection Lost]
E) Alarm Class                          [Info]
F) Trap Number                          [508]
```

Connection Lost Alarm Enable is an ON/OFF toggle to enable the IPRC alarm. Default setting is OFF.

Connection Lost Timeout is the number of seconds to wait before declaring the connection lost. The exact conditions for timeout vary according to the IPRC method. Default setting is 60 seconds.

- **Alcatel OmniPCX**, this setting determines the number of seconds the client will wait before timing out. A timeout will occur if the client has not received any data (either ticket data or any protocol control data) from the switch in the amount of time set forth here. This timeout value can be set from 45 to 90 seconds.

- **Avaya Definity RSP**, this event is triggered when either the socket or the session is down. With RSP, there is an application-layer connection called a Session that runs on top of the lower-layer socket. It is possible that the socket is established but the session is not. RSP will not run if the session is not established. This timeout can be set from 3 to 600 seconds.

- **Generic Server** does not use any application layer protocols so the socket lost timeout is triggered only by the loss of the TCP connection. This timeout can be set from 3 to 600 seconds.

Connection Lost Alarm Actions displays a list of actions from which the action(s) to be taken for this alarm are configured. This field will be empty [ ] if no actions have been configured. Refer to Action List in the Features chapter for more information.

Connection Lost Alarm Message sets the text string to be delivered with this event's action. Default setting is Connection Lost.

- Upon timeout, an Alcatel OmniPCX client disconnects and will attempt to reconnect in 10 seconds. The timeout value is restricted between 45 and 90 seconds. Once the alarm is activated, the alarm will not re-arm until a socket connection is re-established. Avaya RSP and Generic Server are passive TCP servers and cannot attempt to reconnect to the client. They must wait for the client to reestablish the connection.

Alarm Class sets the class for the event. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Trap Number sets the number to be sent with any SNMP traps for this event. Default is 508, but trap number can also be set in the range of 1000 – 1199 as needed.

Serial Handshaking Alarm Settings

Serial Handshaking Alarms allows the T850 to monitor each of its serial ports and alert you if the DTR signal from the connected devices drops low. This would be an indicator that the connected device has failed, the cable between the T850 and the device has been disconnected, or a number of other reasons depending on the device. It can also alert you when the DTR signal goes high again.

```
TeleBoss 850 - Serial Handshaking Alarm Settings
A) I/O 1 Serial Handshaking Alarms      [OFF]
B) I/O 2 Serial Handshaking Alarms      [OFF]
```

I/O *n* Serial Handshaking Alarms displays a menu for configuring alarming on serial DTR handshaking conditions.

```
TeleBoss 850 - I/O 1 Serial Handshaking Alarms
A) Serial Handshaking Low Alarm Enable  [OFF]
B) Serial Handshaking Low Alarm Actions []
C) Serial Handshaking Low Alarm Message [Handshake Low]
D) Serial Handshaking Low Alarm Class   [Info]
E) Serial Handshaking Low Trap Number   [510]
F) Serial Handshaking High Alarm Enable [OFF]
G) Serial Handshaking High Alarm Actions[]
H) Serial Handshaking High Alarm Message[Handshake High]
I) Serial Handshaking High Alarm Class  [Info]
J) Serial Handshaking High Trap Number  [510]
```

Serial Handshaking Low/High Alarm Enable is an ON/OFF toggle to enable alarming on high or low handshaking levels. Default setting is OFF.

Serial Handshaking Low/High Alarm Actions displays the Actions List, a menu where the action string for the alarm is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Serial Handshaking Low/High Alarm Message is the message sent with any text-based action for this event. Default setting is “Handshake Low/High”. (Max length for each is 126 chars)

Serial Handshaking Low/High Alarm Class sets the class for the event. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Serial Handshaking Low/High Trap Number sets the number to be sent with any SNMP traps for this event. Default is 510, but trap number can also be set in the range of 1000 – 1199 as needed.

CPE Alarm Settings

```
TeleBoss 850 - CPE Alarm Settings
A) Alarm Enable                            [OFF]
B) Alarm Actions                           []
C) Alarm Trap Number                       [511]
D) Alarm Class                             [Info]
E) Return to Normal Actions                []
F) Return to Normal Trap Number            [511]
G) Return to Normal Class                  [Info]
```

These settings are only for use with Customer Premises Equipment (CPE) managed via the Asentria SitePath secure, unified administration portal software. Contact Asentria Technical Support for further information.

Alarm Enable is an ON/OFF toggle to enable the CPE Down Event. Default setting is OFF.

Alarm Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Alarm Trap Number sets the number to be sent with any SNMP traps for this event. Default is 511, but trap number can also be set in the range of 1000 – 1199 as needed.

Alarm Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Return to Normal Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Return to Normal Trap Number sets the trap number which can be useful when using SNMP trap managers that employ a trap numbering system to help identify incoming traps. The default trap number for CPR Down Events is 511, but any number in the alternate range of 1000 – 1199 can be used.

Return to Normal Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Event Message Settings

```
TeleBoss 850 - Event Message Settings
A) Include Date and Time                                [ON]
B) Include Site Name                                    [ON]
C) Include Sensor ID                                    [ON]
D) Include User Defined Name                            [ON]
E) Include User Defined State                           [ON]
F) Include Event Class                                  [OFF]
```

Include Date and Time / Site Name / Sensor ID / User Defined Name / User Defined State / Event Class are each ON/OFF toggles to permit customization of the event message that appears in SNMP traps, Emails, SMS messages, pages, etc. sent by the T850. Default setting for each is ON, except for Include Event Class which defaults to OFF.

Action Definitions

This menu is where you configure all of the actions possible when events are detected.

```
TeleBoss 850 - Actions Definition Menu
A) Hostname/IP Address 1                []
B) Hostname/IP Address 2                []
C) Hostname/IP Address 3                []
D) More Hostnames/IP Addresses          []
E) Email Address 1                      []
F) Email Address 2                      []
G) Email Address 3                      []
H) More Email Addresses                 []
I) Phone Number 1                       []
J) Phone Number 2                       []
K) Phone Number 3                       []
L) Phone Number 4                       []
M) Pager Number 1                       []
N) Pager Number 2                       []
O) Pager Number 3                       []
P) Pager Number 4                       []
Q) Action Settings
```

Hostname / IP Address *n* sets the hostname or IP address of the device(s) receiving SNMP Traps. The number (1,2,3) corresponds to the “index” number for Traps as discussed in the Action List section of the Features chapter.

More Hostnames / IP Addresses displays the Hostname/IP Address Definition Menu where three more hostnames or IP Addresses (index 4,5,6) can be configured.

Email Address *n* sets the Email address of the person(s) receiving Email alerts. The number (1,2,3) corresponds to the “index” number for Email alerts as discussed in the Action List.

More Email Addresses displays the Email Address Definition Menu where three more Email Addresses (index 4,5,6) can be configured.

Phone Number *n* sets the phone number (index 1,2,3,4) to call for each dispatch, malert or modem callout as discussed in the Action List.

Pager Number *n* displays the Pager *n* Settings menu where each of four individual pager settings (index 1,2,3,4) can be configured.

Action Settings displays the Action Settings menu where specific settings to manage actions can be configured.

Pager Number *n*

```
TeleBoss 850 - Pager 1 Settings
A) Pager Type                           [NUMERIC]
B) Pager Callout Number                 []
C) Pager ID                             []
D) Numeric Message                      []
E) Post Callout Delay (seconds)         [15]
F) Post ID Delay (seconds)              [5]
```

Pager Type toggles between NUMERIC and ALPHA to set the type of pager being called.

Pager Callout Number sets the phone number for the pager.

Pager ID is used only with paging systems where many pagers share the same phone number. This is common with alphanumeric pagers. (Max length is 19 chars)

Numeric Message sets the series of digits (typically callback number) sent to a numeric pager. (Max length is 19 chars)

Post Callout Delay sets the number of seconds (0 to 255) the unit will wait before sending the pager ID. Default setting is 15 seconds.

Post ID Delay sets the number of seconds (0 to 255) the unit will wait before sending any message data. Default setting is 5 seconds.

Action Settings

```
TeleBoss 850 - Action Settings
A) Callout Attempts                               [5]
B) Callout Delay (seconds)                        [60]
C) Action Schedule                                [OFF]
D) Reminder Interval (minutes)                    [120]
E) Asentria Alarm Version                         [1.1]
F) Require Asentria Alarm ACKs                    [OFF]
```

Callout Attempts sets the total number of times to attempt dispatch, Malert or modem callouts if previous attempts fail. Default setting is 5.

Callout Delay sets the time in seconds (0 - 400) to wait between callout attempts. Default setting is 60 seconds.

Action Schedule displays the Action Schedule Settings menu where actions can be limited to defined days and times.

Reminder Interval sets the time in minutes (0 – 65535) at which an action is repeated if the sensor (contact closure, temperature, humidity, or voltage) that triggered the alarm is still in the “active” state. When the sensor has been returned to the inactive state, the reminder interval is no longer in effect. Default setting is 120 minutes.

Asentria Alarm Version toggles between 1.1 and 1.0 to indicate which type of Asentria Alarm notification will be displayed. Refer to the Asentria Alarms section in the Features chapter for a detailed explanation of Asentria Alarms.

Require AsentriaAlarm ACKs is an ON/OFF toggle to enable or disable forcing the unit to require an acknowledgment when first connecting, and after each Asentria Alarm. If disabled, the T850 will allow non-CRC mode where Asentria Alarms are delivered without waiting for any indication that the messages were properly delivered. If enabled, CRC mode is required by the T850. Refer to the Asentria Alarms section for more information about Asentria Alarms and CRC and non-CRC modes. Default setting is OFF.

Action Schedule

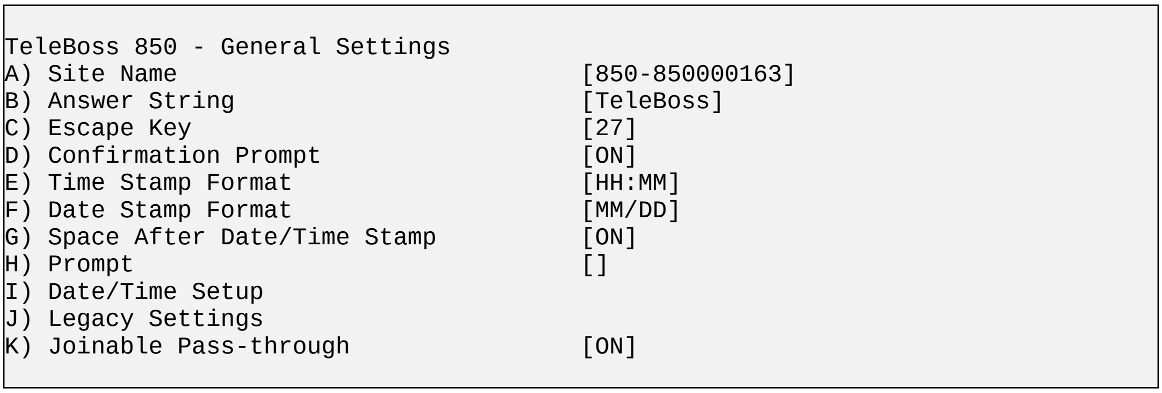
```
TeleBoss 850 - Action Schedule Settings
A) Action Schedule Enable                [OFF]
B) Begin Time                            [08:00]
C) End Time                              [17:00]
D) Weekdays Only                         [ON]
```

Actions Schedule Enable is an ON/OFF toggle to enable the action schedule. Default setting is OFF.

Begin Time/End Time sets the beginning and ending times (24 hr clock) during which alarm actions can be taken. Default settings are 08:00 (Begin Time) and 17:00 (End Time).

Weekdays Only toggles whether actions are only performed Monday thru Friday. Default setting is ON.

General Settings



```
TeleBoss 850 - General Settings
A) Site Name                                [850-850000163]
B) Answer String                            [TeleBoss]
C) Escape Key                               [27]
D) Confirmation Prompt                      [ON]
E) Time Stamp Format                        [HH:MM]
F) Date Stamp Format                        [MM/DD]
G) Space After Date/Time Stamp              [ON]
H) Prompt                                   []
I) Date/Time Setup
J) Legacy Settings
K) Joinable Pass-through                    [ON]
```

Site Name sets the name assigned to this T850. This name is included with alarm messages (Traps, Emails, etc.) and is displayed at the top of the Status screen. The name should be unique for clarity. (Max length 40 chars) Default setting is “850 - <*serial number*>”

Answer String sets the string that is presented when a user connects to the T850 via Telnet or modem. (Max length 31 chars)

Escape Key is the decimal ASCII character code of the key you must press three times to escape from passthrough or other transparent modes. Default is 27, the <ESC> key.

Confirmation Prompt is an ON/OFF toggle to set whether a confirmation prompt (`Are you sure (y/n)?`) is displayed when the commands **COLDSTART, DEFAULT, DEFAULT ALL,** and **ZERO** are issued, and when clearing the settings for an EventSensor in the EventSensor Setup menu. If there is no response within 30 seconds, the T850 will cancel the command. Default is ON.

Time Stamp Format toggles through three options for how time stamps are formatted: HH:MM, HH:MM:SS, or Blank. Default setting is HH:MM.

Date Stamp Format toggles through four options for how date stamps are formatted: MM/DD, MM/DD/YY, MM/DD/YYYY, or Blank. Default setting is MM/DD.

Space After Date/Time Stamp is an ON/OFF toggle to set whether a space is appended to the end of the Date/Time stamp. Default setting is ON.

Prompt sets the character(s) or settings values displayed as the command line prompt. Refer to the Customizable Command Prompt section in the Features chapter for more information. (Max length 63 chars)

Date/Time Setup displays the System Date/Time menu where you can manage the clock, daylight savings control, and configure a networked time server.

Legacy Settings displays a menu for configuring legacy products that may be connected to the T850.

Joinable Pass-through is and ON/OFF toggle to allow or disallow multiple user pass-through sessions. ON allows more than one user to connect on a pass-through session. OFF does not allow more than one concurrent pass-through session, and those attempting to join after the first user is connected will receive a “port in use” error message. Default setting is ON.

Date/Time Settings

```
TeleBoss 850 - System Date/Time
A) Current Date                     [04/29/2010]
B) Current Time                     [09:53:39]
C) Adjust for Daylight Savings      [ON]
D) GMT Difference (hours)           [8]
E) GMT Difference Direction         [BEHIND]
F) Enable Time Protocol             [OFF]
G) Time Servers
```

Current Date sets the date. The unit automatically calculates the day of the week to display on the Status screen.

Current Time sets the time (24 hr clock).

Note: The date and time settings are maintained by means of an internal battery backup when power is removed from the T850.

Adjust for Daylight Savings is an ON/OFF toggle that allows automatic daylight savings time updating.

- A brief explanation of daylight savings time (effective 2007): On the second Sunday in March, clocks are set ahead one hour at 2:00 a.m. local standard time, which becomes 3:00 a.m. local daylight time. On the first Sunday in November, clocks are set back one hour at 2:00 a.m. local daylight time, which becomes 1:00 a.m. local standard time.

GMT Difference (hours) sets the number of hours the current time zone is offset from GMT. Valid input ranges from 0 to 12. Default setting is 8 hours.

GMT Difference Direction sets whether you are east (ahead) or west (behind) of GMT. For example, Pacific time (GMT-8) is behind and Tokyo time (GMT +9) is ahead. Default setting is BEHIND.

Enable Time Protocol toggles between OFF, SIMPLE, and NTP.

- SIMPLE - With network time set to SIMPLE the unit attempts to contact the configured time servers (see Time Servers setting below) periodically, attempting to query each using Simple Network Time Protocol (SNTP), Time, and Daytime protocols, in that order. Once a response is received for any protocol, the unit sets the system clock to the new time, updates the real time hardware clock (RTC), then the network time process dies. The interval for checking network time is hard-coded to 12 hours plus or minus a random several hours.

- NTP **–** With network time set to Network Time Protocol (NTP), the NTP daemon is kept running at all times. Unlike the SIMPLE setting, with NTP the clock is not immediately set as soon as a time server is contacted. Rather, the NTP daemon utilizes various algorithms to set the time in an accurate and robust manner. Since the NTP daemon updates the system time asynchronously, the current time is stored in the RTC every 30 minutes while it is running. Note that if you change the clock manually, it may be a period of an hour or more before NTP resets it.

Time Servers displays a menu where the hostname or IP address of six time-servers can be configured. (Max length 64 chars) The T850 uses the following servers by default:

- time.nist.gov - 192.43.244.18 - Boulder, CO
- time-b.nist.gov - 129.6.15.29 - Gaithersburg, MD

Legacy Settings

```
TeleBoss 850 - Legacy Settings
A) Release Compressed                  [OFF]
B) Autodelete After Polling            [OFF]
C) Wait for NEXT                       [OFF]
D) Omit END DATA                       [OFF]
E) Line Tag                            [OFF]
F) Release Mode                        [LINE]
G) CBB DLE Stuffing                    [OFF]
H) CBB Retransmits                     [5]
I) CBB Timeout                         [15]
   Note: These settings are for support of older systems and
         should NOT be used for new implementations. We do
         not guarantee that these settings will be present in
         future versions or products.
```

Released Compressed is an ON/OFF toggle to enable release of data in a compressed or uncompressed format. Default setting is OFF.

Autodelete After Polling is an ON/OFF toggle to enable the deletion of data from the call record database once it has been polled. Default setting is OFF.

Wait for NEXT is an ON/OFF toggle that causes the unit to wait for the NEXT command before sending data once the RL command has been issued. Default setting is OFF.

Omit END DATA is an ON/OFF toggle that causes the unit to send or omit the string "END DATA" when a command processor poll is complete. Default setting is OFF.

Line Tag is an ON/OFF toggle that adds or omits the serial number line tags on each line of stored data. Default setting is OFF.

Release Mode toggles the following modes of releasing stored data: LINE, XMODEM, and CBB. Unless your application specifically uses XMODEM or CBB, leave this set to the default setting of LINE.

CBB DLE Stuffing / Retransmits / Timeout are specific configuration options for polling via Compressed Binary Block (CBB) mode. CBB is a release method included for compatibility and is not otherwise documented in this manual.

Event Log Settings

The Event Log is a record of all data events that occur within the T850.

```
TeleBoss 850 - Event Log Settings
A) List Events File
B) Clear Events File
C) Enable Events Log File              [ON]
D) Maximum File Size                   [32]
E) Store Data Alarm Records            [OFF]
F) Store Sensor Events                 [OFF]
G) Date/Time Stamp Data Alarm Records  [OFF]
H) Prepend Data Alarm Name             [OFF]
```

List Events File displays the contents of the Events File, if any records exist.

Clear Events File purges the records within the Events File. Records in the Events File are deleted immediately when this option is selected, so make sure you want to do this before selecting.

Enable Events Log File is an ON/OFF toggle to enable Event logging. Default setting is ON.

Maximum File Size sets the maximum number of KB the Event File can reach before overwriting the oldest records. Available options are 0, 32, 64, 128, 256, 512 and 1024. Default setting is 32.

Store Data Alarm Records is an ON/OFF toggle to enable storing data alarm records. Default setting is OFF.

Store Sensor Events is an ON/OFF toggle to enable storing records generated by environmental sensors. Default setting is OFF.

Date/Time Stamp Data Alarm Records is an ON/OFF toggle to prepend a Date/Time stamp to the beginning of data alarm records. Default setting is OFF.

Prepend Data Alarm Name is an ON/OFF toggle to prepend the name of the Data Alarm to the beginning of the data alarm record. This aids in identifying which Data Alarm an alarm record is associated with. Default setting is OFF.

Audit Log Settings

The Audit Log is a record of a variety of actions that occur within the T850. The Audit Log is stored as a CRDB file; that is, it is accessed and controlled under the same policies which govern how you would generally access buffered data. For example, you can have the Audit Log FTP-pushed like any other CRDB file. Unlike other CRDB files, you can view the Audit Log from within the Audit Log Settings menu. The Audit Log overwrites itself when it becomes full. The **`audit.log.maxsize`** setting controls the maximum size (in K) to which the file should be limited. If the setting is 0 then the Audit Log's only constraint on size will be the available physical memory. This available memory could be used for more important data buffering which is why the default maximum audit log file size is 32, not 0.

```
TeleBoss 850 - Audit Log Settings
A) List Audit Log File
B) Clear Audit Log File
C) Enable Audit Log File                [ON]
D) Maximum File Size                    [32]
E) Store Reset Events                   [ON]
F) Store Command Entry                  [ON]
G) Store Output Activity                [ON]
H) Store Alarm Actions Taken            [ON]
I) Store Password Failures              [ON]
J) Store Logins/Disconnects             [ON]
K) Store Serial Handshaking Alarms      [ON]
L) Store Pass-through Activity          [ON]
M) Store Inactivity Timeouts            [ON]
N) Store Polling Activity               [ON]
O) Store DIP Switch Changes             [ON]
```

List Audit Log File displays the contents of the Audit Log file, if any records exist.

Clear Audit Log File purges the records within the Audit Log file. Records in the Audit Log File are deleted immediately when this option is selected, so make sure you want to do this before selecting.

Enable Audit Log File is an ON/OFF toggle to enable Audit logging. Default setting is ON.

Maximum File Size is the maximum number of KB the Audit Log can reach before overwriting the oldest records. Available options are 0, 32, 64, 128, 256, 512, and 1024. Default setting is 32.

The remaining options are ON/OFF toggles to enable logging of the action described. Default settings for all is ON.

How to secure the Audit Log

Secure the Audit Log with any of the following steps:

1. Restrict what information it contains.
2. Restrict which users can access it.

3. Restrict which users can configure the Audit Log to be FTP-pushed.

The most you can do to restrict what information the Audit Log contains is to disable the Audit Log by setting **audit.log.enable** to OFF, also in Setup Menu -> Audit Log and the Logs -> Audit Log portion of the Web UI. Nothing goes in the Audit Log when it is disabled. If you want some information to go in the Audit Log then configure the settings for the kind of information you want logged.

Restrict which users have access to the Audit Log by changing their file permissions. There are two permissions: read (a.k.a. release) and write (a.k.a. delete), which are accessible with **sec.user[x].audit.readaccess** and **sec.user[x].audit.writeaccess**, also available in the Setup Menu -> Security -> Specific Security -> User Profile x -> File Release Permissions/File Delete Permissions, and the Security -> User Profiles -> User Profile x -> File Release/Delete Permissions portion of the Web UI. The possible values for these settings are ALLOW and DENY.

The FTP Push settings are writable by a user with ADMIN1 rights or greater. Restrict which user(s) have ADMIN1 rights in order to prevent users who would ordinarily not have permissions to view the Audit Log to configure the unit to FTP-push the Audit Log to some server where that user could read them.

Scripting Settings

Scripting is a T850 feature that is complex enough that it has been given it's own chapter in this User Manual. The initial Scripting Settings menu is displayed below, but a full description of the options along with other information necessary to use the scripting functions can be found in the Scripting chapter.

```
TeleBoss 850 - Scripting Settings
A) Enable Scripting                     [OFF]
B) Clear Pending Records                [0]
C) DTR Override Ports
D) List Allocated Devices
E) List Scripts
F) Manage Script Files
```

Features and How To Use Them

Upgrading the T850

Save the update file (850-x.yy.zzz-std-a71.udf) to a directory on your PC or an FTP server. FTP upgrades can be done in either of two ways: by using the T850's FTP client to get the update file, or sending the update file from another host to the T850's FTP server. Following are the instructions for both methods.

Note: Before upgrading it is always a good idea to make a copy of the Setting Keys file in your T850, in case settings are lost during the upgrade. This usually does not happen, but it's better to be safe than sorry.

T850 as FTP client method:

From the command line type: **xf f get <update filename> <host> <username>**

(note: you can type 'xf' at the command prompt to get usage for this command.)

Here is an actual session:

```
➢  xf f get 850-x.yy.zzz-std-a71.udf 10.10.5.32 anonymous
Receiving 850-x.yy.zzz-std-a71.udf via FTP
Anonymous’s password:
……..
COMPLETE

<and the update starts here>
```

T850 as FTP server method:

1) Make an FTP connection to the T850 using a username and password that has MASTER rights.

2) Type **hash** at the ftp prompt. (This is optional - it just creates hash marks (###) while the file is transferring so you can see something happening.)

3) At the next ftp prompt type: **put drive:\directory\<update filename>**
For example: put C:\upgrades\850-x.yy.zzz-std-a71.udf

4) Hash marks will now appear to show you that the file is transferring. When the transfer is complete you will be returned to an ftp prompt.

5) Type: **BYE** at the ftp prompt. The unit still has to process this file, which takes about 5 minutes, at which time the unit will reboot. When the unit detects the update file and begins processing it. Wait until the unit reboots before proceeding.

6) After the T850 reboots, connect to it and either check the top line of the Status screen, or type **VER** at the command line. You should see that the unit is now upgraded to the new version.

7) Check your settings to be sure none have been lost. If they have, reload the Setting Keys file.

Note: While the T850 is processing the update file, it is very important that the unit not be power-cycled, nor should the Reset button be pushed.

Note: The update file can be transferred via several other methods, including Xmodem, Zmodem, Ymodem, ASCII, TFTP and SFTP. Contact Asentria Technical Support for instructions.

Setting Keys

Setting Keys (SK) provide a flat file, human readable, means of setting and retrieving settings within the unit. Setting Keys are commonly used to clone settings across multiple units or in automated processes.

Setting Keys is abbreviated when used on the command line as **SK**. Following are commands when working with the Setting Keys File from the command line of the unit.

SK [KEY[=*value*]] allows for reading or setting a single Setting Key. If the value portion of the command is omitted, the T830 will report back the value stored in that key. If the value is given, it will be stored in the key.

SK GET [X|A [CUSTOM] [*filter*]] initiates a download of unit settings. This listing can be retrieved either by Xmodem or plain ASCII using the X and A attributes, respectively. If the transfer mode attribute is omitted, the unit will prompt for the download method. The CUSTOM tag may be used to retrieve only the settings that are not set to factory defaults. A filter may be applied to limit the keys output to just the branch specified. For example, to retrieve an ASCII listing of all EventSensor settings, use the command: **SK GET A EVENT.SENSOR**

SK SET [X|A] puts the unit in bulk Settings Keys upload mode. Any of the settings retrieved by **SK GET** can be manipulated and uploaded with new values. The unit will process settings in any order or number; not all settings need to be uploaded each session. As with **SK GET**, both ASCII and Xmodem transfer methods may be used to upload settings to the unit. These transfer methods are indicated by using the X and A attributes, respectively. The T850 monitors for invalid Setting Keys and will notify you after the upload if any invalid data was received.

When using **SK SET** in ASCII mode, the data uploaded must end with a line consisting of the word "END" followed by a return.

SK HERE allows you to set or get individual keys interactively. Typing just the key name will cause the value to be displayed. Typing the key name plus a new value will set that key. The unit will keep prompting for a new key or key/value pair until you press <Esc> or <Enter>.

SK LOG displays a list of any errors generated during an SK Set.

Setting Keys can also be retrieved and loaded via FTP.

FTP> GET SKALL FILENAME.TXT retrieves all of the Setting Keys for the unit, similar to the **SK GET A** command described above.

FTP> GET SKCUSTOM FILENAME.TXT retrieves any settings that are not set to factory default, similar to the **SK GET A CUSTOM** command described above.

FTP> PUT FILENAME.TXT SKALL and **PUT FILENAME.TXT SKCUSTOM** load the settings in FILENAME.TXT onto the T850.

Upon successful completion of loading the settings FTP will respond with "`226 - Transfer complete`". If there is a problem in the Setting Keys file then FTP will respond with "`226 - Transfer complete; errors in setting key file! Type Get SKLOG to view`"

FTP> GET SKLOG retrieves the Setting Keys log as described above.

SK Commands	Description
sk get script sk r	Dumps all scripts
sk get sk g	Dumps all Setting Keys followed by scripts. Setting Keys are wrapped with <keys>…</keys> XML-like header and footer text.
sk get custom sk c	Dumps only the custom Setting Keys – no scripts. Setting Keys are not wrapped with <keys>… </keys> XML-like header and footer text.
sk get status sk ?	Dumps only the Status Keys – no scripts. Status Keys are not wrapped with <keys>… </keys> XML-like header and footer text.

Status Keys

Status Keys are read-only keys that can be read using the command **SK [KEY]**. The current value of the key will be displayed.

Other commands to obtain a dump of all current Status Key values are:

sk get x status	to start an Xmodem download of the Status Key file
sk get a status	to start an ASCII download of the Status Key file
sk g status	to start an ASCII download of the Status Key file
sk ?	to start an ASCII download of the Status Key file

Or, you can log into the FTP server and issue the "**get skstatus**" command.

The table below lists the general Status Keys available on the T850. Other Status Keys that apply to specific hardware (wireless modem, GPS) are defined in those sections of this manual.

Status Key syntax	**Values**	**Description**
`event.mgmt.count`	Non-negative integer	Returns the number of events currently under management in the unit. It is the combined size of the RAM and flash queues of events. Upon generation, events go into the flash queue unless there is space in the RAM queue. Only when the events are in the RAM queue can their respective actions be processed. Events are moved from the flash queue to the RAM queue as soon as RAM queue space becomes available.
`event.sensor[x].cc[y].value`, where x is the EventSensor number and y is the contact closure number	0 to 3600	Returns 0 for contact closure open, and 1 for contact closure closed.
`event.sensor[x].relay[y].value` , where x is the EventSensor number and y is the relay number	0 to 3600	Returns 0 (inactive, de-energized) or 1 (active, energized).
`net.dsl.error`	String	Returns what errors (if any) were detected with the DSL interface configuration after the unit was commanded to try and use the DSL interface and the configuration was detected to be invalid.
`net.dsl.info.isp.discreason`	String	Returns why, if available, DSL connectivity was lost.
`net.dsl.info.isp.ip`	Dotted quad	Returns what IP address the DSL interface is using with the ISP.
`net.dsl.info.isp.linktime`	String	Returns how long the unit has been connected to the ISP since the connection was started.
`net.dsl.info.isp.status`	Connected, Not connected	Returns whether the unit is connected to the ISP; it returns "Connected" or "Not Connected". Another key that gives the same information in a different format is: `net.dsl.status`.
`net.dsl.info.link`	Connected, Not connected	Returns whether the unit is connected to the ISP; it returns "Connected" or "Not Connected". Another key that gives the same information in a different format is: `net.dsl.info.isp.status`
`net.dsl.info.speed`	String	Returns the speed of the link (provided there is DSL connectivity, as shown with `net.dsl.info.link`).

net.dsl.info.updated	String	Returns the last date/time at which the values in the **net.dsl.info.*** key hierarchy were last updated. These values are updated when directed by the user (by setting **net.dsl.command** to 20) or every few seconds by the unit until the ADSL modem is connected to the ISP (at which time it doesn't update until directed by the user or ISP connectivity is lost).
net.dsl.info.ver.atm	String	Returns the ADSL modem ATM driver version.
net.dsl.info.ver.dslhal	String	Returns the ADSL modem DSL HAL version.
net.dsl.info.ver.fw	String	Returns the ADSL modem firmware version.
net.dsl.info.ver.pump	String	Returns the ADSL modem data pump version.
net.dsl.info.ver.sarhal	String	Returns the ADSL modem SAR HAL version.
net.dsl.info.ver.sw	String	Returns the ADSL modem software version.
net.dsl.status	Integer >=0	Returns the state of the DSL interface. Refer to DSL Status section for more details.
net.eth[x].link, where x is 1 or 2 for first or second Ethernet interface	Up, Down	Returns whether the interface is up or down. It's used to determine if there is anything on the other end of the Ethernet cable ("Up" if yes).
net.vpn.error	String	Returns any error in VPN setup that was detected when a VPN connection (either passive or active) was attempted.
net.vpn[x].status, where x is the VPN number	0..2	Returns the status of VPN x. Refer to the VPNs section for more details.
sec.vpn.auth.pubkey	String	Returns the public part of the unit's IPsec key pair for RSA digitial signature authentication
sys.commission.ip	Dotted quad	Returns the reserved IP address the unit uses only for SitePath commissioning.
sys.commission.state	0..17	Returns values indicating the state of the SitePath commissioning process.
sys.crdb.file[x].records, where x is 1 to <number of serial ports>, or 17 for the Event Log or 18 for the Audit Log.	Integer >=0	Returns the number of records in a CRDB file.

Securing a TeleBoss 850

This section discusses all facets of security that must be considered when installing a TeleBoss 850. For adequate security, you must consider the following:

- Security mode
- SNMP
- Telnet/FTP
- SSH (Secure Shell)
- RTS (Real Time Sockets)
- Web UI (User Interface)
- Button Unlock
- IP Address Restrictions
- VPN (Virtual Private Network)
- NetPoll Feature

Security mode

The security mode (`sec.mode`) tells the unit how to control users' access to it. You can configure either User Profiles mode or RADIUS mode. (See Security Settings Menu). For either mode, you can restrict by what methods a user can connect, as well as whether the user receives "Username:" and/or "Password:" when prompted for those items. Be careful to always preserve a way to access the unit as a MASTER user (that is, a user with rights=MASTER). This is the user with full access to configure all settings and invoke all commands. If you are using User Profiles, ensure, before you log out, that you have a MASTER user configured and that you don't forget its password. If you are using RADIUS then you can configure a MASTER user any time as long as you can configure users on the RADIUS server. Before logging out of the unit when configuring RADIUS, ensure the unit can ping the RADIUS server, and that you verify that a user can access the unit via RADIUS. If the user cannot log in to the unit via RADIUS then you will need your existing login in order to gather data to help troubleshoot why the RADIUS user cannot log in.

If you are logged into the unit, you can put traffic on any network to which the unit is connected. For example, pinging a host on the network, FTP-ing to it, SSH-ing to it, Telnet-ing to it. Therefore good security comes from making it so no unauthorized persons have access to the unit. This is something you must ensure with the User Profiles or RADIUS security mode configurations.

SNMP

By default anyone can access the unit via SNMP, and the TeleBoss's MIB is fully featured with configuration objects. Therefore if you don't take care to secure SNMP, you leave the unit open to unauthorized users. There are 3 ways to secure SNMP.

1. turn it off (`net.snmp.enable=OFF`)
2. leave it enabled for all SNMP versions (`net.snmp.enable=ALL VERSIONS`) but ensure that the community name is a strong password and that all user profiles have strong passwords. Be aware however then for snmpv1 and v2c, the community names are transmitted in the clear, as with Telnet, so anyone eavesdropping on the network may get unauthorized access to the unit.
3. set it to V3 only (`net.snmp.enable=V3 ONLY`) and either use RADIUS or use a User Profiles configuration that has strong passwords.

Telnet/FTP

Keep in mind that like SNMP, login credentials (and all application content) are transmitted in the clear for Telnet and FTP, so anyone eavesdropping on the network could gain unauthorized access to the unit. Therefore, to tighten security on Telnet, either do not use it, forbid it (with `sec.connectvia`), or use it with RADIUS/CHAP or User Profiles with one-time password or challenge response.

SSH (Secure Shell)

To enable SSH access to the T850, you must generate a host key with the SSHC command (see the section on SSHC for details). This is the preferred network access method over telnet of course because the traffic is encrypted.

RTS (Real Time Sockets)

Out of the box the T850 allows connections to TCP port 220x unauthenticated. So unauthorized access to FILEx data is possible unless you tighten RTS via the authorization controls in RADIUS or User Profiles security modes. Remember that just like SNMP, Telnet, and FTP, any login credentials you require for RTS connections are passed in the clear, so anyone eavesdropping on the network could gain unauthorized access. To limit exposure of the user password, use RADIUS/CHAP or User Profiles with one-time password or challenge response. Alternatively, you can forbid RTS connections altogether with the `sec.connectvia` setting.

Web UI (User Interface)

The T850 supports both HTTP and HTTPS. Like SNMP, Telnet, and FTP, HTTP is vulnerable to eavesdropping. Therefore to tighten security for web UI access, do not use it or only access the unit via HTTPS (which is encrypted with SSL).

Button Unlock

With the Button Unlock feature, you can regain access to a unit that you have been locked out of. This is meant as an insurance policy against the only other resort to locking yourself out, which is returning the unit to Asentria.

When this feature is set to ON (default setting), the user can tap the Reset button 5 times quickly (1-2 times per second), at which point the front-panel LEDs will flash briefly for several seconds, giving the user immediate Console access using the default MASTER username and password.

These are the settings that are defaulted by this process:

`sec.mode` (reset to USER PROFILES)
`sec.consolereq` (reset to OFF)
`sec.connectvia` (reset to every method of connecting)
"admin/password/MASTER" credentials for the user profile appropriate to the product

If you do not want the Button Unlock feature enabled, for example in environments where physical access is not assumed to be trusted with access, then be sure to turn it off (`sk sec.button.unlock=OFF`), or set the Button Tap Allows Console Access in the Security Settings/General Security Settings menu to OFF.

If you lock yourself out and gain access again with the Button Unlock feature, remember to reconfigure the settings that were defaulted by the Button Unlock feature to maintain your prior security configuration!

IP Address Restrictions

With the IP Address Restrictions feature you can select what kind of network traffic the unit should ignore or heed based on the source IP address of such IP frames.

VPN

For the highly secure, flexible, and centralized network access control (aside from unplugging the network cable), use IPsec VPNs to SitePath (Asentria's secure, unified administration portal software). VPNs are disabled and unconfigured by default. Refer to SitePath documentation for details on how to manage units with SitePath via VPN.

NetPoll Feature

NetPoll is a feature developed for one customer of Asentria's which all other users will never use. However it can pose a security risk if it is enabled. When enabled, it causes the T850 to listen on TCP port 3001 for an incoming connection from the polling machine, which it then accepts. This feature is set using one of the following two Setting Keys:

```
sec.connectvia=ON
sec.connectvia.netpoll=ON
```

By default, neither of these Setting Keys are set to these values, so unless they are specifically set as such the T850 will not accept any connection attempt from TCP port 3001.

Telnet/TCP Connections

The T850 provides support for Telnet/TCP connections via two internal Ethernet interfaces. Refer to the Ethernet Settings menu for information on how to configure these.

All Telnet connections are TCP connections but not all TCP connections are Telnet connections. A Telnet connection is made to the T850 by using the Telnet protocol and by specifying a TCP port address. 'Telnet' refers to a TCP connection made on port address 23, which specifies that characters are supposed to be handled a certain way. The T850 supports Telnet connections and also supports some custom assigned port numbers to facilitate certain connection features.

The following information assumes that you know how to run your computer to establish and use Telnet/TCP connections and only require the specific information relating to the T850 features. Port numbers below include "x" where "x" is the corresponding T850 file or port number. (ie; 2101 refers to the Telnet passthrough connection made on serial port 1.)

- **Port Address 200x**: A connection to port 200x is just like a regular Telnet connection to port 23, except it sets the default file for retrieving data or the default port when the **BYPASS** command is given.

- **Port Address 210x** : A connection to port 210x routes you directly to the device connected to the corresponding serial (I/O) port. A banner message will be displayed indicating you are connected to that I/O port. To disconnect from this access mode press the <ESC> key twice. Refer to the Passthrough section in this chapter for more information.

- **Port Address 220x**: A connection to port 220x is referred to as a Real-Time Socket. These are sockets that are dedicated to exporting data from file "x" in the T850. If there is any data already stored in a particular file, it will first be transferred out of the T850 to the user or machine initiating the connection. After all the data currently in the file is transferred out, any data that is coming into the T850 will be immediately transmitted out and across this connection. Refer to the Real-Time Sockets menu for information on how to configure these.

How to secure Telnet

Secure Telnet with any of the following steps:

1. Disable Telnet.
2. Enable the "authentication required" options for Telnet-based services.
3. Restrict which users are allowed to log in via Telnet.
4. Enable enhanced authentication for Telnet-based services.
5. Enable RADIUS security mode.

Disable Telnet by setting `sec.connectvia.telnet` to OFF, also in Setup Menu -> Security -> General Security -> Globally Allow Access via, and the Security -> General Settings portion of the Web UI.

Enable the "authentication required" options for Telnet-based services (applicable to User Profiles security mode only) by setting the `sec.tcp23req`, and `sec.tcp210xreq` settings to ON. This is also in the Setup Menu -> Security -> Specific Security -> Authentication Settings -> TCP/IP Port 23 Requires Password and TCP/IP Port 210x Requires Password, and the Security -> User Profiles portion of the Web UI. You may also choose to make the unit require the username upon login by setting `sec.authmode` to "USERNAME/PW", also in the nearby portion of the Setup Menu labeled "Username and/or Password Required".

When authentication is required (in RADIUS security mode, or in User Profiles security mode when the "authentication required" options are enabled), then you can restrict which users are allowed to log in via Telnet. Do this my setting `sec.user[x].connectvia.telnet` to ON or OFF (where x is the User Profile number, 1 – 12), also in the Setup Menu -> Security -> Specific Security -> User Profile x -> Allow User Connection via, also in the Security -> User Profiles -> User Profile x portion of the Web UI. When using RADIUS security mode, use the Asentria-Connect-Via-Telnet vendor-specific attribute.

Enable enhanced authentication for Telnet-based services (applicable to User Profiles security mode only) by setting `sec.user[x].challenge.telnetmode` to CHALLENGE or SEND PASSWORD, also in the Setup

Menu -> Security -> Specific Security -> User Profile x -> Additional Authentication Options -> Secure Authentication via Telnet, or the Security -> User Profiles -> User Profile x portion of the Web UI. Enhanced authentication uses challenge/response or one-time-password mechanisms; this avoids having to transmit user credentials in the clear. For more detail on enhanced authentication options, refer to the "Setup Menu -> Main Setup Menu -> Security Settings" section of the User's Manual.

When RADIUS security mode is enabled, Telnet must be enabled (i.e., `sec.connectvia.telnet` must be ON) in order to use Telnet with RADIUS. The other two ways of securing Telnet ("authentication required" setting and enhanced security settings) are not applicable to RADIUS security mode.

VLANS

A VLAN (802.1Q Virtual Local Area Network) is used to separate broadcast domains via software instead of via hardware (physical layout of network devices and cabling). Software on network nodes (like the T850) abstracts this into virtual network interfaces, so each interface can have its own virtual interface configuration (static address, subnet mask, router). The unit operates with virtual interfaces the same as it would with real interfaces.

Configuration
Each Ethernet interface can have up to 6 VLANs bound to it. Access configration items via any of the following:

Menu: Setup -> Network Settings -> Ethernet Settings -> Ethernet x -> VLAN Settings
Web: Networking -> Ethernet Settings -> Ethernet x Settings -> scroll down to VLANx Settings
Keys:
```
net.eth[].vlan[].id
net.eth[].vlan[].priority
net.eth[].vlan[].ip
net.eth[].vlan[].mask
net.eth[].vlan[].router
net.eth[].mode
```

VLAN ID

0 to 4094; this is what identifies the VLAN.

VLAN priority

0 to 7; this is the priority assigned to egress frames.

IP, mask, router

Configured like any other interface. This router setting is included in the set of candidate default routers which the unit can use. The unit does not yet support configuration of individual host and network routes. Select the default router with the `net.default.router` setting, if the unit has not already selected an appropriate one for you. Note that the unit does not heed changes to network configuration while you are connected to a command processor via Telnet or SSH. Changes, including population of the candidate default router set, are pended until all network-based command processor sessions have ended.

Network mode

Set this to VLAN to engage the interface in VLAN mode. While the interaface operates in VLAN mode, its normally configured settings (IP, mask, router) are still configured but the interface does not use them. The interface heeds those settings only when it's in STATIC mode.

Example
Put the unit on three VLANs bound to the cable attached to the first Ethernet adapter, 10.20.20.0/24, 10.30.30.0/24, and 10.40.40.0/24, with VLAN ids 20, 30, and 40, respectively. The unit will route off its local nets via the 10.30.30.1 router.

Configure:
```
net.eth[1].vlan[1].id=20
net.eth[1].vlan[1].ip=10.20.20.2
net.eth[1].vlan[1].mask=255.255.255.0
net.eth[1].vlan[2].id=30
net.eth[1].vlan[2].ip=10.30.30.2
net.eth[1].vlan[2].mask=255.255.255.0
net.eth[1].vlan[2].router=10.30.30.1
net.eth[1].vlan[3].id=40
net.eth[1].vlan[3].ip=10.40.40.2
net.eth[1].vlan[3].mask=255.255.255.0
net.eth[1].mode=vlan
```

If no other interfaces are active then the unit will select 10.30.30.1 as the default router (gateway); if other routers are configured for other interfaces then you can override this by configuring `net.default.router`.

VPNs

This section of the Features chapter is a discussion of Virtual Private Networks relating to how the T850 communicates with SitePath, Asentria's secure, unified administration portal software. For a full description of how SitePath is configured and administered, please refer to the SitePath User Manual and other user documentation that comes with SitePath.

A Virtual Private Network (VPN) is a network that is tunneled (the virtual part), typically across a public network, and secured (the private part), typically with IPsec or SSL.

VPN on-demand (VOD)

VPN on-demand (VOD) is a feature where the VPN between a deployed unit and SitePath is not always up. Instead it is brought up in response to:

- a command to bring it up sent by SitePath
- a purpose to bring it up generated by the unit, after that purpose has been authorized by the SitePath Message Processor (SMP).

It is brought down in response to USC Proxy (USCP) authorizing a request made by the unit to bring down its VPN. SitePath examines conditions and determines yes/no decisions for authorizing a VPN to come up and go down.

The VPN architecture in SitePath version ***1.00.xxx*** is one where all deployed units always have a VPN up to SitePath. Remote access, alarm management, and configuration management were handled transparently with the assumption that there is always a secure tunnel between SitePath and every deployed unit.

The VPN architecture in SitePath versions **>= *1.01.000*** is one where deployed units can be commissioned to either always have a VPN up to SitePath, or only have a VPN up when needed. To make more conservative use of resources, it is recommended such that units be commissioned such that VPNs are brought up only when needed. That is, with VOD is enabled (this is done by enabling it in the unit web UI upon commissioning). Because units are typically deployed behind firewalls at customer sites, the unit must initiate any kind of network traffic -- SitePath cannot ordinarily initiate a VPN to a unit deployed behind a firewall. For this reason a lightweight UDP network channel is implemented called the Unit SitePath Channel (USC). When the VPN is not up, the USC is used to control when the VPN must be raised. When the VPN is up, the USC (which then operates over the VPN) is used to control what the VPN can be used for and when the VPN can go down.

If SitePath needs to do remote access or configuration management of a deployed unit, it commands the unit to raise the VPN via the USC. When the unit needs to send any traffic to SitePath (alarm traffic, email, etc.), it uses the USC to raise the VPN. When the VPN is no longer needed (no remote access or configuration management, and no traffic to send to SitePath from the unit), the VPN is taken down. The USC is always running between the unit and SitePath and the unit can only initiate the USC (because the unit is typically behind a firewall). Without the USC, the VPN cannot be raised, and without the VPN, you cannot do remote access, alarming, email, FTP push, and SNMP notifications via SitePath.

The USC itself is selectively secure. That is, traffic is only secure (i.e., encrypted and authenticated with 256-bit Blowfish and HMAC-SHA1) when it needs to be secured and is not secure when it does not need to be secured. Currently the only USC traffic that is transmitted non-secure is traffic that does not need to be secure: the serial number of the unit. This data is transmitted in keepalive frames which are used to keep the channel between SitePath and the unit open through routers and firewalls.

Configuration

To use VPN on-demand, configure **`net.vpn.ondemand.enable`**`=on` on the unit. This setting is on by default in unit version >= 2.04.040 and off by default in previous versions. No SitePath configuration is necessary.

Usage

In addition to the two areas where the user notices the impact of VPN on-demand – Raising a VPN and Lowering a VPN –VOD can also be used for Automatic Data Delivery and Restricted Trust.

Raising a VPN
In SitePath version < *1.01.000*, a SitePath user clicked the Connect button in the SitePath web UI in order to initate remote access. The Connect button immediately turned into a Disconnect button (meaning the connection was set up immediately). This speed is because the VPN to the unit is always up. Now with VPN on-demand (SitePath version >= 1.01.000), the VPN may be down when a SitePath user clicks the Connect button. To raise the VPN there is a delay of typically 15 seconds while the VPN is negotiated. During this time the Connect button (labeled as "Connect (will entail a delay)") turns dim. Once the VPN is up the dim Connect button turns into a non-dim Disconnect button.

On units with version >= *2.04.030*, the vpn can be raised multiple ways:

- **`sk net.vpn.1.cmd`**=2
- cause an event that has an action that causes the unit to connect to SitePath
- enter **DOTRAP**, if any of the configured SNMP managers are the address of SitePath
- enter **DOMAIL**, if the configured SMTP server is the address of SitePath
- enter **PUSHTEST** or **PUSHNOW**, if the configured FTP push server address is the address of SitePath
- wait for the unit to raise a VPN on its own (or SitePath's own) accord, which can happen in multiple ways:
  - SitePath user wants access to the unit or any of its configured CPEs that are visible to SitePath
  - unit needs to sync its clock (clock sync is automatically configured during commissioning)
  - unit needs to deliver event actions to SitePath or to a machine via SitePath
  - unit needs to FTP push CDR to SitePath

When raising a VPN via **DOTRAP**, **DOMAIL**, or **PUSHTEST**, the user receives feedback about SitePath connectivity progress, much like the user receives feedback when they use those commands and cause PPP to be raised. There are two main factors to consider when the unit sends data to SitePath:

1. the VPN status; if it is down, it needs to be raised.
2. the authorization status; all types of traffic sent over the VPN first needs to be authorized to be able to use the VPN, and this is negotiated over the VPN with SitePath before that type of traffic (e.g., email, alarms, etc.) is commenced. Once a type of traffic is authorized for a VPN, it remains authorized until the VPN goes down.

Once a VPN is raised, it will remain up until it is decided and agreed by both the unit and SitePath that the VPN should go down. This typically happens due to inactivity timeout, which can controlled by the SitePath key **`vpn.idle.timeout`**. (3 minutes by default) Note that so long as a SitePath user is connected to a unit or any of its CPEs, the VPN will not go down, even if there is no activity on the VPN to warrant the inactivity timeout triggering.

Lowering a VPN
A VPN between SitePath and a deployed unit is lowered when no SitePath user has a remote access connection to the unit or to a CPE attached to the unit, and the inactivity timer for the VPN has expired. The inactivity timer is 3 minutes by default, but can be changed with SitePath key **`vpn.idle.timeout`**. When the VPN is lowered, a subsequent operation to raise the VPN has a typical delay of 15 seconds, but can be longer depending on unpredictable factors such as processor loading and network integrity.

Automatic Data Delivery
Automatic Data Delivery is a general term to describe any data the unit needs to send to SitePath: alarms, emails, SNMP notifications, polling data, etc., which happens over a VPN. An end user may notice the effect of VOD when they try to, for example, send an alarm to SitePath and the VPN between the unit and SitePath is down. The attempt to send a trap causes the unit to raise the VPN, which has an inherent delay. After the VPN is up then the trap is sent. Therefore sending a trap appears to take as long as it took to raise the VPN under this circumstance.

Restricted trust

Restricted trust (introduced in SitePath 1.01.000 and Omnix Release 2.04.030) is a way of using a unit with SitePath such that the end user does not trust SitePath completely; in other words, the end user maintains full admin privileges over the unit (and SitePath does not have full admin privilege of the unit) and restricts their trust of SitePath. The unit and SitePath are still connected but SitePath (and any SitePath users or the SitePath administrator) is not always authorized (i.e., is not completely trusted) to access the unit and CPEs behind that unit. Restricted trust helps end users have more control over what CPEs are accessible when by SitePath, as well as the degree to which SitePath can do certain functions on the unit (such as loading updates and settings).

There are two ways of thinking about restricted trust: coarse adjustment and fine adjustment.

Coarse adjustment

Restricted trust is configured with a setting called `sys.sitepath.trustmode` on the unit at the time of commissioning (also in the Commissioning page of the unit web UI). There are two values: FULL and RESTRICTED.

- FULL means the unit (and the end user) trust SitePath fully: SitePath or anyone behind SitePath can do anything on the unit (this is called master access to the unit) and the end user network.
- RESTRICTED is for end users less trusting of SitePath or at least more strict about authorizing what SitePath can do on their networks. It means the unit (and end user) do not trust SitePath fully. In this mode of operation, SitePath does not have master access to the unit. Without master access, you can't configure CPE's, and you can't Telnet/SSH to nodes on the end user's LAN from the unit.

Restricted trust must be configured at the time of commissioning. If one configures full trust, commissions the unit, and then changes the trust mode setting to restricted trust, that alone is not enough to make the unit restricted from SitePath's perspective -- you must recommission (i.e., decommission and then commission again) the unit while the unit is configured with restricted trust.

Restricted trust also has two other associated settings, `sec.action.loadsk` and `sec.action.loadupdate`. These control whether a unit commissioned under restricted trust allows SitePath to load update files onto the unit or load settings onto the unit. By contrast, when a unit is commissioned under full trust, SitePath always has the authority to load settings and updates. In the unit web UI, these two settings are represented by the "Trust SitePath to load settings/updates" controls in the Commissioning page. These two drop-down controls are yes or no, but the actual values of the settings are are access levels (0-7). In a more general sense, these settings specify the minimum access level (master, admin3, etc.) of a user that is necessary for that user to load settings or updates. Specifically for SitePath, this means that:

- when the web UI control is set to YES and trust mode is RESTRICTED, then the `sec.action.*` setting is set to access level 5 (which equals admin3). Since SitePath is given admin3 rights to the unit in restricted trust mode, this setting being 5 means that SitePath can do what the setting says (either load settings or updates).
- when the web UI control is set to NO and trust mode is RESTRICTED, then the `sec.action.*` setting is set to access level 6, meaning that SitePath cannot do the associated action (load settings or updates). In FULL trust mode, SitePath is given master rights to the unit, so it does not matter what the `sec.action.*` settings are (which is why their associated controls in the web UI are dimmed out when the trust mode is set to FULL).

Restricted trust affects a SitePath user in that when they go to initiate access to any CPEs they have permission to access (permission as granted by the SitePath Administrator, confgured via the SECURITY section of the SitePath Web User Interface), they may get a message saying that a CPE is unauthorized. They then have the option of requesting authorization from the end user through in that same web UI page. When the end user authorizes access, the SitePath user can then proceed with their remote access tasks. At any time the end user can deny access to SitePath (and by extension, all SitePath users).

Restricted trust affects end users in that they can feel comfortable knowing that although they have outsourced management of certain aspects of their network, the end user solely posseses the authority on deciding what gets accessed when on their network. End users also have a fine-grained way to control access to CPEs which is discussed in the next section.

In sum, restructed trust means that SitePath, and by extension the SitePath administrator, and by further extension the SitePath users, cannot access any end-user-LAN IP address unless it is configured as a CPE, and only the end user can configure the CPEs (because the CPE settings require master rights to change). Under restrictred trust, SitePath (and its adminsitrator and its users) do not have master rights to a unit. Therefore, this feature solves of the problem of "how to prevent SitePath from unauthorized access to nodes on the end user LAN". End users authorize access when end users configure CPEs, which happens at commissioning time -- presumably the end user does the commissioning, not a technician from the entity running SitePath. Under restricted trust, end users have master rights (somebody/something must and in restricted trust mode, it is not SitePath), so they (end users) are the ones that authorize access.

Fine adjustment
There is also the problem of "how to more finely adjust when a CPE can be accessed", which is where the CPE authorization feature comes in. CPE authorization means that for each CPE, there is a setting that specifies whether the CPE is currently authorized for SitePath access (and by extension anyone behind SitePath: its administrator and its users). In this way, the CPE can be in the SitePath web UI, but not accessible until the end user excplicity authorizes access, once access is requested by a SitePath user, via the actions configured for the CPE Authrorization Requested event on the unit (introduced in unit version 2.04.030). This is explained in further detail in the next paragraph.

When a user clicks the connect button for a CPE, and the CPE is not currently authorized, SitePath causes the unit to generate an event that means "SitePath wants to access CPE x -- please authorize?". The end user can configure actions for this event, like emails or traps. So for example the end user could get an email saying "please authorize CPE x". Once the end user authorizes access, the CPE is accessible from SitePath (and by extension, its administrator and its users), and the end user can deny access at any time after that. The way that the end user authorizes and denies access to the unit from SitePath is by browsing to the General->Commission Settings->Network CPE Devices section of the unit web UI. For each CPE, the end user can choose to

- deny
- authorize indefinitely
- authorize for a set of preset durations (1 hour, 6 hours, 24 hours). When authorizing for these durations, it means that a timer is set for each CPE for the chosen duration. The unit automatically denies access to that CPE when that CPE's timer expires, or if the unit is reset.

The ability for SitePath users to route to CPEs depends on both SitePath and the unit. SitePath has its own permissions architecture for managing who is authorized to access certain CPEs on its end. The unit also has its own similar permissions architecture for authorizing which CPE is accessible from SitePath, and this is something the end user has complete and exclusive control over in restricted trust.

In sum, the problem of authorizing CPE access is a legitimate concern for IT administrators. Coarse adjustment of authorzation happens with the feature of restricted trust. This is a blanket way of saying only certain CPEs are accessible, and SitePath has limited capability/authority to affect the unit, particularly no authority when it comes to configuring CPEs. Fine-grain adjustment of authorization happens with the CPE routing authorization feature. So under restricted trust, the end user blanketly says SitePath:

1. has limited privilege to do certain things,
2. cannot change the CPE configuration, and
3. for the set of configured CPEs, may need additional on-the-fly authorization from the end user. The authorization and denial of this access all happens through SitePath and the unit. For SitePath users, it happens through SitePath (in the form of a button labeled "Request Authorization" or Re-request authorization" in the CPE detail page of the SitePath Web User Interface). For end users, it happens by browsing to the unit web UI and selecting an authorization option next to a certain CPE.

Also, a single SitePath installation can operate with a mix of units: some commissioned with FULL trust, others commissioned with RESTRICTED trust.

VPN Client

SSL VPN Client support is where the unit runs OpenVPN version 2.1_rc15 to connect to a an OpenVPN server to form a VPN where SSL/TLS is used for authentication and key exchange.

The benefits of using SSL VPN Client are:

- SSL VPNs are simple, unlike other VPN technologies such as IPsec.
- SSL VPNs can work through NAT-ing routers/firwalls, unlike other VPN technologies such as IPsec.
- The OpenVPN distribution is freely available and works on a variety of platforms including Unix/Linux, Windows, and Mac.

When configuring SSL VPN Client it is best to use a question and answer format because it is relatively complex.

How do I specify SSL VPN Client mode?

Set **`net.vpn.mode`** to `SSL CLIENT`.

How many VPNs can I configure?

The unit can be configured with up to 2 VPNs. The configuration settings for these VPNs are under the **`net.vpn.*`** key branch.

How many VPNs can I run at one time?

Although the unit supports multiple VPN configurations, only 1 VPN can be operational at any one time. The setting that controls which VPN can be operational is the **`net.vpn.active`** key. It has values of VPN1, VPN2, or NONE.

Is my VPN connecting to SitePath?

The unit uses this feature to connect to SitePath. If you are using it with SitePath, typically most of the more arcane configuration items are automatically configured by SitePath. However, if you are configuring your own VPN server then you need to tell the unit that by setting **`sys.sitepath.vpn`**`=NONE`.

Where is my VPN connecting to?

As a client, the unit must know where the server is. You tell it the server's address with the **`net.vpn[x].remote.host`** key. Set it to an IP address or DNS name of the server, or the IP address or DNS name of the NAT-ing firwall viewable from the unit that will route the VPN connections to the server. Note that if you use a DNS name, you must have DNS configured on the unit. Sometimes, DNS can be configured automatically when you choose DHCP Ethernet addressing and **`the net.dns.mode`** to be `ETH1-DCHP` or `ETH2-DHCP`.

What network medium (network interface) should my VPN use?

Depending on the application, the unit can have multiple network interfaces at its disposal: Ethernet, wireless modem, ADSL, and POTS PPP. The **`net.vpn[x].if.public`** key controls which interface the VPN uses. By default the unit uses the network interface that owns the IP route to the VPN server. (This is when **`net.vpn[x].if.public`** is set to `ANY`.) But you may want to have the unit use an explicit interface for VPN. The primary purpose for this that if the VPN is not always used, and the interface you want the VPN to use is not always used, then the unit knows that to bring up the VPN, it must first bring up the interface. The secondary purpose is to provide protection for situations where the VPN is using one interface, but then another interface that's not always used comes up, possibly overriding the default route, and you don't want the VPN to follow the default route and hop on to the other interface unintentionally (thus breaking VPN connectivity).

Should my VPN start automatically when the unit starts?

If yes, then set **`net.vpn[x].startmode`** to `AUTO-ACTIVE`. If no then set it to MANUAL. When in MANUAL startmode, start the VPN by setting **`net.vpn[x].cmd`**`=2`. Once started, the VPN will maintain connectivity until told to stop (either by setting **`net.vpn[x].cmd`**`=0`, or by the unit resetting when the VPN is in MANUAL startmode). If there is no connectivity to the server, as long as the VPN is configured correctly, the unit will keep trying to connect to the server until it connects or it is told to stop.

How do I know the VPN is working?

To check the status of the VPN, read the `net.vpn[x].status` key. It returns one of 3 values:

- 0 (which means the VPN is off)
- 1 (which means the VPN is trying to start)
- 2 (which means the VPN is operational)

Note that the return value of 2 means the tunnel is up, but does not necessarily preclude configuration errors from preventing VPN traffic to pass. So to ultimately know the VPN is operational, in addition to verifying `net.vpn.status` returns 2, you should also ping the server from the unit using the VPN address of the server. (Or you can ping the unit from the server, using the VPN address of the unit.)

You can also use the `net.vpn[x].cmd` key to read the status of the VPN.

Do I need to give the VPN a name?

You may want to describe the VPN or give it a name; use the `net.vpn[x].description` key for that. This has no functional purpose, it is just for making a note.

How does the unit know the VPN server is authentic (and vice versa)?

The unit uses certificate-based SSL/TLS security to authenticate the server (and the server uses the same thing to authenticate the unit). Configuring certificates can be done with Setting Keys, but is likely more simple for a user to use the SSLC command on the unit. The SSLC command allows unit administrators to manipulate the SSL VPN certificates and other authentication data associated with the VPN.

The SSLC command takes a variety of command line arguments that tell it what to do. These arguments are mainly broken down into "actions" and "items"

- actions
  - add: add an item (load it into the unit)
  - list: list an item (display what is already in the unit)
  - delete: delete an item
- items
  - certificate
  - key
  - CA certificate
  - DH parameters

The idea behind this paradigm is that you do something (an action) on something (an item).
The command line arguments that specify actions and items are:

-e	Specify item:	certificate
-k	Specify item:	key
-r	Specify item:	CA certificate
-t	Specify item:	TLS-auth key
-h	Specify item:	DH parameters
-l	Specify action:	list item
-a	Specify action:	add item
-d	Specify action:	delete item

You must also specify which VPN you want this applied to with the "-v" command line argument:

-v x Specify VPN x, where x is 1 or 2

For example, to load the CA certificate for VPN 1, enter **`SSLC -a -r -v 1`**

The unit cannot generate its own SSL authentication key/certificate. You must do this (presumably with an OpenVPN server installation) and load the certificates/keys on the unit with the SSLC command. It is recommended you use the SSLC command either in a trusted network environment via Telnet or via SSH. This is for two reasons:

1. The data you upload is text format, and is accepted without any application layer protocol like Xmodem. Therefore to make eliminate communcation errors, use the protocol on a TCP-based command processor (like Telnet or SSH).

2. Some of the things you must transfer using the SSLC command are secret data (the key and the TLS-auth key). "Secret" means that only the unit knows about it (and possibly the server as well, if that is kept in secure location), and if this key is compromised then the security of the entire VPN is compromised.

The CA certificate is the certificate of the certificate authority that both the unit and the server trust. The CA signs both the certificate for the server and the certificate for the unit. The CA certificate must exist on both machines.

So it works through NAT-ting routers, that means it uses TCP or UDP, right?

It can use either UDP or TCP, although it works optimally with UDP. Change this to suit your firewall access policies with the `net.vpn[x].ssl.proto` key (its values are "TCP" and "UDP"), and the `net.vpn[x].ssl.port` keys (its value is an integer for the TCP/UDP port you choose).

I'm paranoid about security, how do I make it as secure as possible?

There are four things you can do to improve security with OpenVPN.

1. Add more HMAC authentication using a pre-shared key called a TLS-auth key. This is manipulated with the SSLC command with the "TLS-auth key" item. The key must be generated by the OpenVPN server.
2. Add the requirement that the unit must specify the credentials of a user account on the OpenVPN server in order for the unit to connect. The credentials are specified on the unit with the `net.vpn[x].ssl.username` and `net.vpn[x].ssl.password` keys.
3. Configure a cipher you are comfortable with. See the next question for how to configure the cipher.
4. Use a server certificate with the "server" nsCertType value, and configure the client to require a "server" nsCertType certificate (more on this in the next section).

I already have a server...how do I make the unit cooperate?

The server is configured with a text configuration file; this is the first place to look to figure out what you need to configure on the unit. The unit essentially maintains the same configuration file, but you cannot edit it directly. Instead, you specify settings via the unit's setting keys, and then the unit generates the configuration file from the setting keys.

Some keys are specific: they specify the VPN protcol and VPN port, or the certificate to use. The previous answers in this section have discussed how to configure such things on the unit. Other setting keys on the unit are generic: they merely specify text where you can enter an OpenVPN configuration option. The idea is to look at the server configuration to see what configuration items it requires on the client, and then supply any further configuration items that you require on the unit, minus any configuration items that the unit handles automatically for you. First, let's go over what a generic key is.

A generic key is of this form: `net.vpn[x].ssl.conf[y]`, where y is a number between 1 and 16. For example, by default, the cipher is "BF-CBC" (128-bit Blowfish CBC). You can change this to be stronger with, say, AES-256-CBC (256-bit AES CBC), with the following setting:

- `net.vpn[1].ssl.conf[7]="cipher AES-256-CBC"`

"cipher AES-256-CBC" is the OpenVPN configuration item, 1 is VPN slot 1 (which could also be slot 2), and 7 is an arbitrary number between 1 and 16 that is unique among any other "ssl.conf" setting keys. In other words, 7 is just an index used to denote you multiple configuration items. You can configure multiple settings, and the 'y' in `net.vpn[x].ssl.conf[y]` can be in any order and not necessarily adjacent. For example:

- `net.vpn[x].ssl.conf[7]="cipher AES-256-CBC"`
- `net.vpn[x].ssl.conf[3]="comp-lzo"`
- `net.vpn[x].ssl.conf[9]="persist-key"`

Some values of OpenVPN configuration items cannot be specified in a generic key. For example, the "ca" OpenVPN configuration item is required. But you cannot specify the "ca" OpenVPN configuration item because the unit already configures that item from the data you provide via the SSLC command.

Now that we've identified what a generic key is, examine the example below to see how to make the unit cooperate.

Example

Here is an example OpenVPN server configuration. It discusses what it means for the server and what it means for the unit. To get a better understanding of OpenVPN configuration, consult the documentation at www.openvpn.org.

```
tls-server
local 10.0.5.171
port 1194
proto udp
dev tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/myserver.crt
key /etc/openvpn/myserver.key
dh /etc/openvpn/dh1024.pem
server 10.8.0.0 255.255.255.0
client-config-dir /etc/openvpn/ccd
tls-auth /etc/openvpn/tlsauth.key
cipher AES-256-CBC
comp-lzo
max-clients 8190
ping 15
ping-restart 60
verb 3
client-connect /etc/openvpn/openvpn.connect.sh
client-disconnect /etc/openvpn/openvpn.disconnect.sh
learn-address /etc/openvpn/openvpn.updown.sh
up /etc/openvpn/openvpn.up.sh
tmp-dir /etc/openvpn/tmp
daemon
management 127.0.0.1 1195
writepid /var/run/openvpn.pid
```

The "tls-server" item specifies that the server will operate in the mode secured by SSL/TLS. This the only mode the unit supports, so if the server does not use tls-server mode then the unit is incompatible with it.

The "local 10.0.5.171" item specifies the address the server listens on. The only impact this has on the unit is that the unit must connect to the server such that its connection ultimately arrives on 10.0.5.171 on the server. Use the **net.vpn[x].remote.host** key to specify this address. Also, if firewalls separate the unit and the server, you should be aware of the firewall configuration, so that the firewall routes traffic to the address on which the server is listening.

The "port" and "proto" items specify what TCP/UDP port is used. The values for these items should match the values for the **net.vpn[x].ssl.port** and **net.vpn.ssl[x].proto** keys on the unit.

The "dev" item specifies whether the server uses bridging or routing. The unit supports routing only (dev tun). If the server says "dev tap" then the unit is incompatible with the server.

The "ca" item specifies the CA certificate. Use the SSLC command to load the CA certificate on the unit.

The "cert" and "key" items specify the server certificate and key. This is only for the server so there is nothing we have to change on the unit to support this. However, note that the unit must be configured with a certificate (and key) (dedicated to the unit, not the same certificate and key used by the server) using the SSLC command. Note also that if the server certificate is generated with the "nsCertType" value of "server", then you can add the "ns-cert-type server" config item to the unit (using the generic **net.vpn[x].ssl.conf[y]** key).

The "dh" item specifies the Diffie Hellman parameters. This is used only on the server so we don't have to configure anything on the unit. (The SSLC command allows for adding DH parameters, but that is used when the unit is in SSL VPN server mode, not SSL VPN client mode as is discussed here.

The "server 10.8.0.0 255.255.255.0" item specifies the addressing method; again this is used only for the server, but impacts the unit in that the unit typically is assigned its address on the VPN from the server.

The "client-config-dir /etc/openvpn/ccd" item specifies the directory for client-specific configuration. Each client (including units) are identified in the client config directory by the common name of its certificate (loaded onto the unit by the SSLC command).

The "tls-auth /etc/openvpn/tlsauth.key" item specifies the key used for the additional HMAC layer. If the server uses this, then the unit must use this too. Specify this key with the SSLC command.

The "cipher AES-256-CBC" item specifies the cipher to use on the VPN; it must match the unit VPN configuration. Specify this item with a generic key, for example: **`sec.vpn[x].ssl.conf[7]`**`="cipher AES-256-CBC"`.

The "comp-lzo" item specifies LZO compression to be used on the VPN; it must match the unit VPN configuration. Specify this item with a generic key, for example: **`sec.vpn[x].ssl.conf[7]`**`="comp-lzo"`.

The "max-clients" item specifies the maximum number of clients that can connect. This is used only the server so we don't have to configure anything on the unit.

The "ping 15" and "ping-restart 60" items specify that the server will send a frame to the client no less often than 15 seconds and restart the VPN after 60 seconds. This does not require the unit to have a similar configuration, although it is recommended that the unit is configured with the "ping" and "ping-restart" items so that the unit does not think the VPN is up when the physical connection is broken.

The "verb 3" item specifies the verbosity level of the OpenVPN syslog output. This configuration on the server is independent of the client. If you want to configure it on the unit then use a generic key to specify it.

The "client-connect", "client-disconnect, "learn-address", and "up" items specify scripts to invoke on the server upon certain client events. This cannot be configured on the unit.

The "tmp-dir" item specifies a temporary directory; again, this is not configurable on the unit.

The "daemon" item specifies that OpenVPN is to run as a daemon on the server. Daemon mode is mandated on the unit, so this is automatically configured and not user-configurable.

The "management 127.0.0.1 7385" item specifies that OpenVPN is to run a management interface accessible on the server's loopback interface via TCP port 7385. This is not configurable on the unit.

The "writepid" item specifies that OpenVPN is to record its process ID to a file; again, this is not configurable on the unit.

In sum, the server configuration file in this example is by no means exhaustive, but it does cover what a typical OpenVPN configuration may look like and how to make the unit work with it in SSL CLIENT VPN mode.

VPN Server

SSL VPN Server support is where the unit runs OpenVPN version 2.1_rc15 to listen for a connection from an OpenVPN where SSL/TLS is used for authentication and key exchange.

The benefits of using SSL VPN Server are:

- SSL VPNs are simple, unlike other VPN technologies such as IPsec.
- SSL VPNs can work through NAT-ing routers/firewalls, unlike other VPN technologies such as IPsec.
- The OpenVPN distribution is freely available and works on a variety of platforms including Windows and Mac

When configuring SSL VPN Server it is best to use a question and answer format because it is relatively complex.

Quick Start

Use this procedure to quickly connect an OpenVPN client to the unit operating as an OpenVPN server.

1. Build keys and certificiates on the client machine. You will need the CA certficate, the client certificate, the client key, the unit certificate, and the unit key.

2. Configure the following OpenVPN configuration file on the client machine:

```
client
dev tun
proto udp
port 1194
remote <address of unit>
persist-key
persist-tun
ca <filanem of CA certificate>
cert <filename of client certificate>
key <filename of client key>
ns-cert-type server
ping 15
ping-restart 60
```

3. Configure the following keys on the unit:

```
sk net.vpn[1].ssl.conf[1]= server 10.99.148.0 255.255.0.0
sk net.vpn.mode= SSL SERVER
sk net.vpn.active= VPN1
```

4. Configure the CA certificate, unit certificate, and unit key on the unit.

Enter the following:

sslc -arv 1

Then load the CA certificate as prompted. All that is necessary is to copy and paste it into your command processor session, starting with "-----BEGIN" and ending with "-----END" and press enter.

sslc -aev 1

Then load the unit certificate as prompted.

sslc -akv 1

Then load the unit key as prompted.

5. Start the VPN on the unit in server mode by entering:

```
sk net.vpn[1].cmd=1
```

6. Start the VPN on the client machine by entering:

openvpn <configuration file>

This procedure forms a VPN where the unit is addressed at 10.99.148.1. Any communication with the unit via this address will happen on the VPN. On the unit, enter the command "sk net.vpn[1].cmd=0" to shut down the VPN.

VPN Server FAQs

How do I specify SSL VPN Server mode?

Set **net.vpn.mode** to `SSL SERVER`.

How many VPNs can I configure?

The unit can be configured with up to 2 VPNs. The configuration settings for these VPNs are under the **net.vpn.*** key branch.

How many VPNs can I run at one time?

Although the unit supports multiple VPN configurations, only 1 VPN can be operational at any one time. The setting that controls which VPN can be operational is the **net.vpn.active** key. It has values of VPN1, VPN2, or NONE.

Am I using this VPN with SitePath?

The unit cannot use this feature to form a VPN with SitePath. If you need to use SitePath, let SitePath configure the unit, which results in using the SSL VPN Client function.

Should my VPN start automatically when the unit starts?

If yes, then set **net.vpn[x].startmode** to `AUTO-PASSIVE`. If no then set it to MANUAL. When in MANUAL startmode, start the VPN by setting **net.vpn[x].cmd**`=1`. Note that this is different than manually starting an SSL VPN client. Once started, the VPN will listen until told to stop (either by setting **net.vpn[x].cmd**`=0`, or by the unit resetting when the VPN is in MANUAL startmode).

Can multiple VPN clients connect to the unit?

Yes. You can enforce the maximum number of clients the unit will support with the “max-clients” OpenVPN configuration item (configurable with the **net.vpn[x].ssl.conf** key, discussed below).

How do I know the VPN is working?

To check the status of the VPN, read the **net.vpn[x].status** key. It returns one of 3 values:

- 0 (which means the VPN is off)
- 1 (which means the unit is listenting for a VPN connection)
- 2 (which means the VPN is operational (and still listening for a VPN connection)

Note that the return value of 2 means the tunnel is up, but does not necessarily preclude configuration errors from preventing VPN traffic to pass. So to ultimately know the VPN is operational, in addition to verifying **net.vpn.status** returns 2, you should also ping the client from the unit using the VPN address of the client. (Or you can ping the unit from the client, using the VPN address of the unit.)

You can also use the **net.vpn[x].cmd** key to read the status of the VPN.

Do I need to give the VPN a name?

You may want to describe the VPN or give it a name; use the **net.vpn[x].description** key for that. This has no functional purpose, it is just for making a note.

How does the unit know the VPN client is authentic (and vice versa)?

The unit uses certificate-based SSL/TLS security to authenticate the client (and the client uses the same thing to authenticate the unit). Configuring certificates can be done with Setting Keys, but is likely more simple for a user to use the SSLC command on the unit. The SSLC command allows unit administrators to manipulate the SSL VPN certificates and other authentication data associated with the VPN.

The SSLC command takes a variety of command line arguments that tell it what to do. These arguments are mainly broken down into "actions" and "items"

- actions
  - add: add an item (load it into the unit)
  - list: list an item (display what is already in the unit)
  - delete: delete an item

- items
  - certificate
  - key
  - CA certificate
  - DH parameters

The idea behind this paradigm is that you do something (an action) on something (an item).
The command line arguments that specify actions and items are:

-e	Specify item:	certificate
-k	Specify item:	key
-r	Specify item:	CA certificate
-t	Specify item:	TLS-auth key
-h	Specify item:	DH parameters
-l	Specify action:	list item
-a	Specify action:	add item
-d	Specify action:	delete item

You must also specify which VPN you want this applied to with the "-v" command line argument:
-v x Specify VPN x, where x is 1 or 2

For example, to load the CA certificate for VPN 1, enter **`SSLC -a -r -v 1`**

The unit cannot generate its own SSL authentication key/certificate. You must do this with another OpenVPN server installation and load the certificates/keys, DH parameters, and possibly TLS-auth key (if you choose the extra layer of security that TLS-auth provides), on the unit with the SSLC command. It is recommended you use the SSLC command either in a trusted network environment via Telnet or via SSH. This is for two reasons:

1. The data you upload is text format, and is accepted without any application layer protocol like Xmodem. Therefore to make eliminate communcation errors, use the protocol on a TCP-based command processor (like Telnet or SSH).

2. Some of the things you must transfer using the SSLC command are secret data (the key and the TLS-auth key). "Secret" means that only the unit knows about it (and possibly the server as well, if that is kept in a secure location), and if this key is compromised then the security of the entire VPN is compromised.

The CA certificate is the certificate of the certificate authority that both the unit and the server trust. The CA signs both the certificate for the server and the certificate for the unit. The CA certificate must exist on both machines.

The "DH parameters" item represents the Diffie Hellman parameters. By default the unit comes with 1024-bit parameters.

So it works through NAT-ting routers, that means it uses TCP or UDP, right?

It can use either UDP or TCP, although it works optimally with UDP. Change this to suit your firewall access policies with the `net.vpn[x].ssl.proto` key (its values are "TCP" and "UDP"), and the `net.vpn[x].ssl.port` keys (its value is an integer for the TCP/UDP port you choose).

I'm paranoid about security, how do I make it as secure as possible?

There are three things you can do to improve security with OpenVPN.

1. Add more HMAC authentication using a pre-shared key called a TLS-auth key. This is manipulated with the SSLC command with the "TLS-auth key" item. The key must be generated by another OpenVPN server installation.
2. Configure a cipher you are comfortable with. See the next question for how to configure the cipher.
3. Use a server certificate with the "server" nsCertType value, and configure the client to require a "server" nsCertType certificate (more on this in the next section).

I already have an OpenVPN client configuration in mind...how do I make the unit cooperate?

The client is configured with a text configuration file; this is the first place to look to figure out what you need to configure on the unit. The unit essentially maintains the same configuration file, but you cannot edit it directly.

Instead, you specify settings via the unit's Setting Keys, and then the unit generates the configuration file from the Setting Keys.

Some keys are specific: they specify the VPN protcol and VPN port, or the certificate to use. The previous answers in this section have discussed how to configure such things on the unit. Other Setting Keys on the unit are generic: they merely specify text where you can enter an OpenVPN configuration option. Once you have your client configuration in mind, you can see what configuration items it requires on the server, and then supply any further configuration items that you require on the unit, minus any configuration items that the unit handles automatically for you. First, let's go over what a generic key is.

A generic key is of this form: **`net.vpn[x].ssl.conf[y]`**, where y is a number between 1 and 16. For example, by default, the cipher is "BF-CBC" (128-bit Blowfish CBC). You can change this to be stronger with, say, AES-256-CBC (256-bit AES CBC), with the following setting:

- **`net.vpn[1].ssl.conf[7]`**`="cipher AES-256-CBC"`

"cipher AES-256-CBC" is the OpenVPN configuration item, 1 is VPN slot 1 (which could also be slot 2), and 7 is an arbitrary number between 1 and 16 that is unique among any other "ssl.conf" Setting Keys. In other words, 7 is just an index used to denote your multiple configuration items. You can configure multiple settings, and the 'y' in **`net.vpn[x].ssl.conf[y]`** can be in any order and not necessarily adjacent. For example:

- **`net.vpn[x].ssl.conf[7]`**`="cipher AES-256-CBC"`
- **`net.vpn[x].ssl.conf[3]`**`="comp-lzo"`
- **`net.vpn[x].ssl.conf[9]`**`="persist-key"`

Some values of OpenVPN configuration items cannot be specified in a generic key. For example, the "ca" OpenVPN configuration item is required. But you cannot specify the "ca" OpenVPN configuration item because the unit already configures that item from the data you provide via the SSLC command.

The generic key has been identified, now examine the example below to see how to make the unit cooperate.

Example

Here is an example OpenVPN client configuration. It discusses what it means for the client and what it means for the unit. For a better understanding of OpenVPN configuration, consult the documenation at www.openvpn.org.

```
client
remote 10.82.3.1
port 1194
proto udp
dev tun
ca /etc/openvpn/ca.crt
cert /etc/openvpn/myserver.crt
key /etc/openvpn/myserver.key
tls-auth /etc/openvpn/tlsauth.key
cipher AES-256-CBC
comp-lzo
ping 15
ping-restart 60
verb 3
daemon
```

The "client" item specifies that the server will operate in the mode secured by SSL/TLS. This the only mode the unit supports, so if the server does not use tls-server mode then the unit is incompatible with it. This item also specifies that the client will allow the server to configure addressing information for it. This implies that on the unit, there must be a "server" configuration option that specifies the virtual network. E.g., "server 10.8.0.0 255.255.255.0" means the server will hand out and address to the client in the 10.8.0.0/24 network. The unit keeps the ".1" address in the virtual network for itself; e.g., the unit would have address 10.8.0.1 in this example.

The "remote" item specifies the address the address to connect to. The only impact this has on the unit is that the unit must listen on the address that the connection ultimately arrives at. Use a generic key to specify this address (e.g., net.vpn[x].ssl.conf="local 10.82.3.1"). Also, if firewalls separate the unit and the server, you should be aware of the firewall configuration, so that the firewall routes traffic to the address on which the unit is listening.

The "port" and "proto" items specify what TCP/UDP port is used. The values for these items should match the values for the `net.vpn[x].ssl.port` and `net.vpn.ssl[x].proto` keys on the unit.

The "dev" item specifies whether the server uses bridging or routing. The unit supports routing only (dev tun). If the client says "dev tap" then the unit is incompatible with the client.

The "ca" item specifies the CA certificate. Use the SSLC command to load the CA certificate on the unit.

The "cert" and "key" items specify the server certificate and key. The unit must be configured with a certificate (and key) using the SSLC command. Note also that if the server certificate is generated with the "nsCertType" value of "server", then you can add the "ns-cert-type server" config item to the client configuration as an extra layer of authentication.

The "tls-auth /etc/openvpn/tlsauth.key" item specifies the key used for the additional HMAC layer. If the client uses this, then the unit must use this too. Specify this key with the SSLC command.

The "cipher AES-256-CBC" item specifies the cipher to use on the VPN; it must match the unit VPN configuration. Specify this item with a generic key, for example: `sec.vpn[x].ssl.conf[7]="`cipher AES-256-CBC".

The "comp-lzo" item specifies LZO compression to be used on the VPN; it must match the unit VPN configuration. Specify this item with a generic key, for example: `sec.vpn[x].ssl.conf[7]="comp-lzo".`

The "ping 15" and "ping-restart 60" items specify that the client will send a frame to the unit no less often than 15 seconds and restart the VPN after 60 seconds. This does not require the unit to have a similar configuration, although it is recommended that the unit is configured with the "ping" and "ping-restart" items so that the unit does not think the VPN is up when the physical connection is broken.

The "verb 3" item specifies the verbosity level of the OpenVPN syslog output. This configuration on the client is independent of the unit. If you want to configure it on the unit then use a generic key to specify it.

The "daemon" item specifies that OpenVPN is to run as a daemon on the server. Daemon mode is mandated on the unit, so this is automatically configured and not user-configurable.

In sum, the client configuration file in this example is by no means exhaustive, but it does cover what a typical OpenVPN client configuration may look like and how to make the unit work with it in SSL SERVER VPN mode.

Secure Shell (SSH) and Secure FTP (SFTP)

The unit offers an SSH client and server. The unit supports SSH version 2 only; SSH version 1 is not supported. SSH is implemented with OpenSSH_4.3p2 and OpenSSL 0.9.8d. The client is used for the SFTP Push feature. For more detail on FTP (and SFTP) push, refer to the FTP Settings section and the SFTP CDR out of the unit section below. The client (**SSH** command) is also usable by any user with at least VIEW rights. This section discusses how to configure authentication and the login banner for the SSH server on the unit. Note that RADIUS security mode cannot be used to authenticate SSH connections.

Configuring the SSH server for password authentication

First connect to the unit command processor via a conventional method in a trusted environment (serial port, Telnet, modem) to make these configuration changes:

1. Generate the host key.
2. Make a user profile with a username and password.
3. Configure network settings.
4. Enable SSH access.

Here are the steps in detail:

1. By default the unit requires password authentication for SSH and does not require public key authentication. To generate the host key, enter **`sshc -ht rsa`** (case sensitive) to create 1024-bit RSA host key.

2. Modify one or more of the user profiles (i.e., configure a strong password for the user profile(s)). This is done via Setup Menu->Security, **`sec.user.*`** settings, or the Security->User Profiles portion of the Web UI.

3. Configure network settings such that the unit is reachable on your network(s). For more detail on this, refer to the Network Settings section of this manual.

4. By default SSH access is enabled. To configure whether it is enabled, use the **`sec.connectvia.ssh`** setting, also in Setup Menu->Security->General Security, and the Security->General Settings portion of the Web UI.

At this point the unit is ready to receive password-authenticated SSH connections. You can do the same tasks you can do on a conventional connection, like unit administration and pass-through, only now it is secured by SSH.

Configuring the SSH server for public key authentication

With public key authentication you do not enter a password to authenticate yourself to the unit. Instead you load the public part of a key bound to your identity onto the unit. In order to use public key authentication:

1. Ensure the unit has a host key.
2. Ensure a user profile is configured with a username and password. Even though the password will not be used during SSH authentication, a password must be configured for the user profile.
3. Ensure SSH is configured for public key authentication. Do this by setting **`sec.ssh.auth.pubkey`** = ON. This overrides password authentication.
4. Ensure an authorized key is loaded on the unit for each client that needs to connect. To load an authorized key, enter **`sshc -ao`** (case sensitive). Then simply paste (or send directly via your terminal) the public key of each user. Terminate loading authorized keys by entering "END" on a line by itself. “END on a line by itself” means you hit enter, then type END, then hit enter again. It is recommended you do this on an error-correcting connection such as Telnet or SSH.

At this point the unit is ready to receive public-key-authenticated SSH connections. The user you connect to the unit as must be configured in a user profile. Also, the public key you use in your SSH client when connecting should be the one of the authorized keys you load on the unit.

The SSH server on the unit has the following preferred ciphers list:
AES-256,3DES,Blowfish,AES-192,AES-128,rijndael-cbc@lysator.liu.se,aes128-ctr,aes192-ctr,aes256-ctr

This means, for example, if your SSH client is configured to support and prefer AES-256 the most then that is the cipher that will be negotiated with the unit.

Configuring the SSH login banner

You can configure the unit to display a login banner when users connect via SSH. Configure this by entering **`sshc -an`** (case-sensitive). Input the banner by sending the banner directory (e.g., paste it into your terminal) and terminating it with "END" on a line by itself. It is recommended you do this on an error-correcting connection such as Telnet or SSH.

How to secure SSH

SSH is inherently a security asset. There is nothing about it that needs to be secured aside from what has already been discussed, namely enabling SSH access and configuring the authentication method. There is however one setting that is useful.

`sec.user[x].connectvia.ssh` (value = ON or OFF)
This setting allows and administrator to limit which users are allowed to log in via SSH (with either password or public key SSH authentication). This can also be set in the Setup Menu -> Security -> Specific Security -> User Profile x -> Allow User Connection via option, and the Security -> User Profiles -> User Profile x portion of the Web UI. When using RADIUS security mode, use the Asentria-Connect-Via-SSH vendor-specific attribute.

SFTP CDR out of the unit

The T850 uses SFTP to transfer CDR securely. SFTP runs on top of SSH version 2 and so has the same security as SSH. The unit supports password and public key authentication methods for SFTP.

If the SFTP host requires a password then the password entered in the FTP Settings menu is used.

If the SFTP host requires public key authentication then do the following configuration steps:

1. Create a client key on the unit. Enter **`sshc -t rsa`** (case sensitive) to create an RSA public/private key pair. The unit will generate the key and then output the key's fingerprint and public part as human-readable mostly base-64 text. The key text will begin with "ssh-" and end with "Asentria_clientkey_<serial number of unit>". You can see the unit's public client key at any time by entering **SSHC**.

2. Configure the SFTP server to make it aware that the unit is authorized to connect. - The SFTP server must know the unit's public client key in order to do public key authentication. This means taking the public client key output by the unit and configuring it in the SFTP server. For UNIX SSH servers (which typically support SFTP), this is done by appending the unit's public client key to the "authorized_keys" file in the ".ssh" directory of the user account the unit uses to SFTP-push CDR. Check with your System Administrator to determine exactly how to do this with your SFTP server.

3. Configure SFTP push - Go to the Setup->Network Settings->FTP Settings menu. Select option A until it reads "SECURE" and then configure the server address, username, password, etc.

4. Establish the authenticity of the SFTP host to the unit. - At this point the unit does not know whether to trust the configured SFTP host. (It may be a malicious host that is pretending to be your host.) Essentially you must tell the unit that you vouch for the host that is running the SFTP server; assuming you are 100% sure that the host to which the unit connects is really your host. Do this by entering **PUSHTEST**. This command is used to see that the connection between the unit and the SFTP (or FTP) host is working. For SFTP, it is also used to let you vouch for the host. The first time you make the unit connect to the SFTP host with the **PUSHTEST** command, you will see a message like the following:

```
The authenticity of host <your SFTP host> can't be established.
RSA key fingerprint is d4:1a:16:46:8a:36:59:24:22:e5:ec:6f:01:fc:74:78.
Are you sure you want to continue connecting (yes/no)?
```

You may enter **YES** (you vouch for the host) or **NO** (you do not vouch for the host) at this point. To help you vouch, the unit reports the host key fingerprint. If this fingerprint is equal to the fingerprint of the host key that you know really belongs to your host, then you can safely vouch for it.

If you enter **NO** then the unit will not be able to push CDR to the SFTP host because it is un-trusted. If you enter **YES** then the unit can trust the server and the server's host key is stored on the unit. As long as the SFTP host key does not change, future connection attempts from the unit to the SFTP host will be trusted.

If the host key does change and you do not vouch for the SFTP host again to the unit (since the host has a new host key) then the unit will revert to not trusting the host (and not push CDR). If this happens and you enter **PUSHTEST**, the unit will say you have to reestablish the authenticity of the SFTP host (see next section).

Reestablishing authenticity of the SFTP host

If the host key changes, you will see something like the following when you enter **PUSHTEST**:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
d7:3a:05:e0:70:4d:2c:15:ae:d2:f1:c2:75:d2:af:53.
Please contact your system administrator.
```

The unit will not push to a host that it sees has a different host key than the one you had vouched for. This is because the unit does not know if the host key changed due to the key of the real host actually changing or due to an imposter server coming on line to pretend to be your host (and thus having a different host key).

If you know your host key has not changed then you know the unit is being eavesdropped on. Otherwise, the host key simply changed and you must reestablish the authenticity of the host to the unit. Do this with the following steps:

1. Delete the old host key from the unit by entering **sshc -dkm <old hostname>**
2. Enter **PUSHTEST** to vouch for the host again.

SSH to Telnet Bridging

SSH to Telnet Bridging is used to allow authorized Telnet access to specific machines from the unit, upon succesfully connecting to the unit via SSH. The benefit of this feature is that if the unit is in a network environment where users can be enabled to have access to certain machines via Telnet (via an SSH connection to the unit) without being allowed access to any other Telnet hosts.

Configuration

There are four steps to configuring SSH to Telnet Bridging:

1) Configure the Telnet hosts to which users need acess. The first 4 CPE configuration slots have a setting which enables this bridging access. Select the CPE configuration slot, configure an IP address and a name, and enable SSH to Telnet Bridging. This can be done via the CPE Settings menu for CPE 1, 2, 3, or 4, or via the Setting Keys shown here.

```
net.cpe[x].ip = <some P address>
net.cpe[x].name = <some name, e.g., my telnet host>
net.cpe[x].stbridging.enable = on
```

2) Configure a user to be authorized to access this Telnet host.

» **Note:** RADIUS cannot be used to authorize users for this feature.

This can be done via the User Setup menu and setting the Pass-through Permissions option to ALLOW for the CPE device that this user will be allowed to access or via the Setting Key shown here.

```
sec.user[x].cpe[y].ptaccess = ALLOW
```

ALLOW is the default value for this setting, so if you have all other users for which you would like to deny bridging access to all CPEs, you have to explicity configure denying them. This can be done by first denying all users, and then allowing your particular user. This can be done by setting the following configuration:

```
sec.user[all].cpe[all].ptaccess = DENY
sec.user[x].cpe[1].ptaccess = ALLOW
```

3) Configure which CPE the user can access. This is done via the User Setup menu and setting the Set Pass-through Pointer To option to the CPE device (CPE 1, CPE 2, etc.) desired, or via the Setting Key shown here.

```
sec.user[x].pttarget = <CPE1> or <CPE2> or <CPE3> or <CPE4>
```

4) Configure how the user can access the bridged CPE. There are three options which can be configured via the User Setup menu and setting the Upon Login Then Go To option to one of the following:

- MENU – upon login, the user is presented with the login menu. If they are authorized to bridge to a CPE, a menu item will be present which they can use to engage this bridge.
- PASSTHROUGH – upon login, bridging access automatically engages to the CPE configured in 3) above.
- COMMAND – upon login the user is connected to the Command Processor, and then enters the **EXIT** command to display the login menu.

Or use this Setting Key: `sec.user[x].loginto = <MENU> or <PASSTHROUGH> or <COMMAND>`

Default Router

The Default Router setting allows you to select the default router (gateway) for the T850. This tells the T850 which router to use if a packet is not on any of the LANs defined on the network port. The default router is selected from the routers defined for the Ethernet ports.

More information for advanced users:

The Default Router setting allows you to select the default router (gateway) for the unit. The unit uses a routing table to determine how to send any outbound IP frame. Each entry in the routing table tells the unit how to send a frame whose destination address matches a rule in the routing table. Routing table entries are examined from most-restrictive to least-restrictive, so the default routing table entry is the last entry in the table since it is the least restrictive. It is the catch-all route: it tells the unit how to send a frame when it doesn't know how else to send it. The only routes on the unit are network interface routes, any static routes you configure, and the default route. Network interface routes tell the unit how to send a frame bound for a machine on one of the unit's local networks (subnets). These routes are automatically configured when you configure the address of a network interface. If an outbound frame is destined for a machine off all local networks then it is sent according to what the default route specifies. The default route specifies the default router to use for these frames.

Each network interface has a router setting which you can configure; this is the machine on that interface to which frames will be sent if they do not route to the local network of that interface. However the unit uses only one of those configured routers at a time - - the default router setting specifies which router the unit will use at a time. As you configure router settings the unit will choose a default router for you. This is available for you to see (and override) via this net.default.router setting. The values you may choose for this setting (i.e., router addresses) are:

- the set of routers which you have specified for Ethernet
- the ADSL interface peer, if you have ADSL hardware installed, represented as "DSL"
- that which is determined by dynamic network interfaces, represented as "DYNAMIC".

DYNAMIC is always a possible value for the default router. It simply means that the default router is set ***only*** according to the default routing rule of any dynamic network interfaces that may be up, such as PPP via the POTS modem or PPP via the Wireless modem. The rule for POTS modem PPP is that whenever that interface is up, it is always the default route and overrides any other default route. The rule for Wireless modem PPP is that it is the default route if the **`net.wireless.defaultrouteenable`** setting is enabled. (If it's disabled then the default route will not be set when the default router is "DYNAMIC".) If the default router is set to anything besides "DYNAMIC", then the default router will be either that (e.g., an Ethernet router) ***or*** that which is determined by the rules of the dynamic network interfaces. In other words, DYNAMIC default router means the default router will be whatever POTS/Wireless modem PPP decides when it is running, and it there will be no default router when POTS/Wireless modem PPP is not running (or when Wireless PPP is running but **`net.wireless.defaultrouteenable`** is off). Any other value for the default router means that the default router will be that value (e.g., an Ethernet router), unless POTS/Wireless modem PPP may be running and thus may override the default route. When POTS/Wireless modem PPP stops and the default router is not set to DYNAMIC, then the default router will revert to the value of the default router setting.

The default router setting is special in that its set of allowed values (the routers for the various network interfaces) are determined at runtime.

Values
Values are dotted-quads and must be in the set of routers configured with **`net.eth.router`** and **`net.eth.vlan.router`**, or they are the special values "DSL" (when ADSL hardware is installed) and "DYNAMIC".

Key syntax
`net.default.router`

Static Routes

Static routes are network routes that specify in a more or less permanent way (*static*) that traffic to a certain destination (destination host or destination network) gets *routed* out a certain interface or via a certain gateway. These give you the ability to fine-tune how outbound network traffic leaves the unit for up to eight different routes.

Configuration

The T850 has a set of 8 static route slots. Each slot has an option to enable it, set the destination net, set the gateway, and set the interface.

- **Enable** is ON/OFF, default OFF.
- **Destination Network** is network notation, i.e., w.x.y.z/s, where s is the significant bits. Default is 0.0.0.0/0.
- **Gateway** is the IP address of the gateway. Default setting is 0.0.0.0
- **Interface** is one of the allowed values: NONE, Ethernet 1, Ethernet 2, Ethernet 1 or 2 VLAN 1, 2, 3, 4, 5, 6, Dialup Modem PPP, and Wireless Modem PPP. Default setting is NONE.

To configure a static ***host*** route you

1. Enable it
2. Specify a destination net with sigbits == 32
3. Specify gateway or interface

To configure a static ***network*** route you

1. Enable it
2. Specify a destination net with sigbits < 32
3. Specify gateway or interface

You can specify a gateway or interface. If you specify a gateway only then the frame will be IP-addressed to the destination subnet and transmitted to the gateway, and the gateway needs to be either a local Ethernet subnet or the peer of a PPP connection (be it wireless or PSTN). If you specify an interface, regardless of specifying a gateway, then the frame will be transmitted out that interface. If it is an Ethernet interface then the destination address (which matches the destination net of the route) will be arped. If it is a PPP interface then the frame which matches its route will be transmitted to the PPP peer.

Note: Specifiying that certain traffic goes out a PPP interface does not cause PPP to be raised when that traffic needs to leave the unit. If a PPP interface is down then any static routes that specify a PPP interface are effectively disabled.

Note: Currently there is no support for Dialup Modem PPP and Wireless Modem PPP to be functional at the same time. Eventually this will not be the case, but in the meantime if you specify a static route with Wireless Modem PPP interface when the Dialup Modem PPP is up instead of the Wireless, then that traffic will go out the Dialup Modem PPP interface.

Setting Keys

- `Net.staticroute.enable`
- `Net.staticroute.destnet`
- `Net.staticroute.gateway`
- `Net.staticroute.if`

Example

Configure to route traffic to the the host 10.90.90.2 to go out via a special gateway 10.90.80.67.

```
net.staticroute[1].enable=on
net.staticroute[1].destnet=10.90.90.2/32
net.staticroute[1].gateway=10.90.80.67
```

Configure to route traffic to 192.168.1.0/24 (which means a subnet of 255.255.255.0) to go out the wireless interface, whenever wireless is up.

```
net.staticroute[1].enable=on
net.staticroute[1].destnet=192.168.1.0/24
net.staticroute[1].if=WPPP
```

IP Address Restrictions

IP Address Restrictions is the primary defense against unauthorized access via a network or PPP connection. An administrator can restrict access by configuring one or more IP addresses that will be the only ones allowed to access the unit. Restrictions can also be configured to allow or deny access to larger groups of IP addresses using .0 and .255 wildcards. IP Address Restrictions do not replace or override any restrictions set by User Profiles, but they do provide an extra level of protection by causing the unit to ignore all network traffic except from the addresses allowed.

IP Address Restrictions are configured from the Setup/Network Settings/IP Address Restrictions menu in all network-enabled Asentria products. When selected, you will see a submenu similar to the following. Selecting option A) Add Item to Table, presents a list of the different kinds of restrictions you can configure.

```
TeleBoss 850 - IP Address Restrictions
 No IP Restrictions Established
A) Add Item to Table

Enter your Selection: a
Enter IP addresses that are allowed access:
0.0.0.0 allows all IP addresses
255.255.255.255 restricts all IP addresses
XXX.XXX.XXX.0 allows all IP addresses in a subnet
XXX.XXX.XXX.255 restricts all IP addresses in subnet

New IP Restriction:
```

From the “New IP Restriction” prompt you can enter up to eight IP addresses that will be allowed access to the unit. The list is exclusive by default, so if you define a single IP address, that one is allowed access while all others are denied.

Wildcards are also available to allow or deny access to larger groups of IP addresses. 0 and 255 serve as wildcards for access and no-access, respectively. For example, an IP restriction of 0.0.0.0 would allow all access to the unit where 255.255.255.255 would allow none. More practically, 192.168.55.0 would only allow traffic from IP addresses beginning with 192.168.55.

Keep in mind that certain outbound network functions in the unit, such as FTP push, Email alerts, and pings, require a response from the receiving device. These devices should not be restricted so the function can be completed successfully.

The Asentria unit evaluates the list of IP restrictions from top to bottom. When it finds an entry that specifically allows or disallows access, it uses that entry and stops looking. For example, examine the following list:

```
TeleBoss 850 - IP Address Restrictions
 1. 192.168.100.20
 2. 192.168.100.1
 3. 0.0.0.0
 4. 192.168.99.255
A) Add Item to Table
B) Delete an Item from Table
C) Delete All Items from Table
```

A computer with a 192.168.99 IP would be granted access to the unit despite #4 because #3 is processed first. #3 allows everyone access. If you wanted to allow everyone access except computers on subnet 192.168.99 you should switch number 3 and 4.

Note: IP restrictions do not replace or override password protection; they simply provide an extra means of security by causing the unit to ignore all traffic from disallowed IP addresses.

If no IP restrictions are defined in this menu, all incoming connections are allowed.

IP Routing

Description
When you connect to the T850 via PPP you can make the unit act as a router between you and devices on one of the unit's local networks. This allows you to communicate IP traffic between you and devices you wish to remotely access. IP routing can also route traffic that originates on the remote site's network to you. By *traffic* we mean ICMP, TCP, UDP.

Benefit
IP Routing allows you remote network access (as opposed to remote RS-232 access) to devices at the unit's site.

Configuration
IP Routing is configured with the following settings.
All Products:

- **net.pprouting.enable**
  This setting controls whether the unit routes IP traffic from PPP to any Ethernet interface.
- **net.ethrouting.enable**
  This setting controls whether the unit routes IP traffic from the specified routing interface to PPP.
- **net.ethrouting.nat.enable**
  This setting controls whether the unit does NAT on routed frames egressing the unit on the PPP interface.
- **sec.user.ppptype**
  This is a per-user setting which controls whether the user under which the PPP session was authenticated can actually route frames to one of the unit's local networks. It is for added security.

Multihomed units only (T850):

- **net.eth.nat**
  This setting controls whether the unit does NAT on routed frames egressing the unit on this interface.
- **net.routing.if**
  This setting controls to which network interface the unit routes PPP traffic.

Example
You want to remotely access the SSH CLI of some piece of equipment at a remote site. SSH rides on TCP so it can be routed and NATted. Install a T850 at the remote site with the following configuration and connect the first Ethernet adapter to the network that has your equipment.

```
// set up ppp user
sec.user[1].name=pppuser
sec.user[1].password=ppppassword
sec.user[1].ppptype=routing

// set up ppp hosting
net.ppphost.enable=on

// set up routing
net.pprouting.enable=on

// set up nat
net.eth[1].nat=on

// set up routing interface
net.routing.if=ETH1
```

Now connect to the unit via PPP and then connect to your eqiupment via your SSH client.

SNMP Trap Capture

The T850 can receive and buffer SNMPv1 traps and SNMPv2c inform-requests (informs), collectively referred to here as “notifications”. Each notification can be subjected to data event evaluation, stored in the Event Log, and delivered via normal Event Log delivery.

When SNMP Trap Capture is enabled, the T850 listens on port 162 for notifications; those over 1024 bytes are ignored. The unit responds successfully to informs as soon as they arrive regardless of the content of the inform.

The first task the T850 does upon receiving a notification that is an inform, is to send a response. It then converts the notification to a multiline record (MLR). A multiline record is an ASCII data packet comprised of 1 or more lines. In this application each line is terminated by CRLF. A trap that is converted to an MLR is called a trap MLR; an inform that is converted to an MLR is called an inform MLR. They are generally called notification MLRs when the difference is irrelevant. There are specific format rules imposed to enable easy use of data events.

1. The first line of the trap MLR specifies the most important common attributes of a trap in this format:

TRAP AA:BBBBB CCCCCCCC DDDDDDDD FROM EEE.EEE.EEE.EEE ENTERPRISE FFF...

where the fields occupied by A - F are:

A. generic trap number (position 6, length 2, padded with 0s) The generic trap number indicates the generic trap type, of which there are 7:

0: coldStart
1: warmStart
2: linkDown
3: linkUp
4: authenticationFailure
5: egpNeighborLoss
6: enterpriseSpecific

B. specific trap number (position 8, length 5, padded with 0s)
C. date the trap was received (in MM/DD/YY format, position 15, length 8)
D. time the trap was received (in HH:MM:SS (24-hr) format, position 24, length 8)
E. source IP address (position 38, length 15, each octet is padded with 0s)
F. enterprise OID (position 65, variable length)

2. The first line of the inform MLR specifies the following:

INFORMREQUEST CCCCCCCC DDDDDDDD FROM EEE.EEE.EEE.EEE

where the fields occupied by C, D, & E are:

C. date the inform was received (in MM/DD/YY format, position 15, length 8)
D. time the inform was received (in HH:MM:SS (24-hr) format, position 24, length 8)
E. source IP address (position 38, length 15, each octet is padded with 0s)

3. Each additional line in the MLR (for both inform MLRs and trap MLRs) is devoted to 1 varBind in the notification.

The format of this varBind line is

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA = BBB...

where the fields occupied by A & B are:

A. varBind OID (position 1, length 40, left-justified, truncated or padded with spaces as necessary)
B. varBind value (position 44, variable length, limited to 115 bytes)

Note: Quote marks are never inserted by the unit in varBind values, even if the value type is OCTET STRING.

4. Every trap MLR and inform MLR has its last line be "END".

5. The entire MLR must conform to the following rules:

- The maximum size of a line is 160 bytes.
- The maximum number of lines allowed in an MLR is 12.
- The maximum total size for an MLR is 1200 bytes.

The unit ignores any varBinds which would cause it to break any of the above rules.

The unit stores notifications in the Event Log depending on the Event Log storage settings (Setup -> Event Log Settings -> Event Log menu). If Store Data Alarm Records is enabled (default is disabled), then all notification MLRs are stored in the Event Log. Since notification MLRs are stored in the Event Log, the user can poll them by any means of polling the Event Log (**TYPE EVENTS** command, FTP, or Setup menu).

The setting key for this feature is **`net.trapcap.enable`**

SNMP Informs

SNMP Inform requires a SMIv2 MIB. When loaded into an SNMP manager, the Asentria SMIv2 MIBs require an associated MIB called Asentria-Root. Both are available from the Asentria website (www.asentria.com) or Asentria Technical Support.

SNMP Inform support (that is, sending SNMP Informs) was added in T850 version 2.00.150.

Unlike SNMP Traps, which do not require acknowledgement from the receiving node, SNMP Informs do require an acknowledgement, thus ensuring guaranteed delivery.

Configuration

SNMP Informs are configured using the following Setting Keys:

`net.snmp.ntfn.attempts`

This is the number of attempts of sending a notification (trap/inform) per cycle (that is, the initial attempt + retries). If this is 0 then there is 1 infinite cycle.

`net.snmp.ntfn.timeout`

This is the number of seconds between 2 attempts to send an SNMP notification in the same cycle.

`net.snmp.ntfn.cycles`

A cycle is a set of notification attempts delimited by a successful action delivery or snooze period. This setting is the maximum number of cycles to try per notification action, where one notification action corresponds to one "inform" keyword in an action list for an event.

`net.snmp.ntfn.snooze`

The snooze period measures the time in minutes between two SNMP notification cycles for any one notification action. That is, if you have two events generate informs, each inform will have its own timeouts for retries and cycles, and its own snooze period.

Then set up an event which does an inform action to an SNMP manager or inform receiver. E.g., **`event.sched[1].actions`**`=inform(10.10.5.10).` A T850 with notification capture enabled can serve as an inform receiver. Remember you can't just send an inform to anything: you must send it to a machine capable of replying to the unit with an inform response. Only when the unit gets the inform response will it consider the inform action a success.

Pass-through

Pass-through (also known as “Bypass”) is a bi-directional communication link for a serial, modem, Telnet or SSH connection through the T850 to a device attached to a serial port. Pass-through is useful for configuring or maintaining devices connected to the T850 without having to be in the same physical location.

Pass-through to a serial port is available on TCP ports 210*n* where ‘*n*’ is the number of the serial port.

Pass-through to a serial port is available via from any command processor, including serial, modem, Telnet or SSH connections using the **BYPASS*n*** command where ‘*n*’ is the number of the serial port.

To terminate a pass-through session, press the Escape Key three times.

Following is a table showing what pass-through sub-features/behaviors are applicable to the T850 and a detailed description of each sub-feature below the table.

Sub-feature	T850
Bypass command	**Yes**
Adjustable end sequence pause	**Yes**
End sequence for network pass-through	**3 escapes (via login menu) or 1 escape (via bypass command)**
End sequence for modem pass-through	**1 escape (via bypass command)**
Joinable sessions	**Yes**
Buffered pass-through	**No**
Allow serial break	**Yes**

Bypass command

The command **BYPASS*n***, where '*n*’ is the number of the serial port, is used from any command processor, including serial, modem, Telnet or SSH connections to establish a pass-through connection**.**

Adjustable end sequence pause

This feature means you can control the minimum amount of time between entering escape characters that the unit will register as an authentic escape sequence. That is, you can set this to 1/4 second, meaning that in order to escape pass-through, you must enter the escape sequence with at least 1/4 second between each escape. The point is to make the unit disregard escape sequences that happen from the pass-through data itself, which is assumed to travel across the link without pauses between the escape characters. The **`sys.pt.endpause`** setting controls this.

Joinable sessions

Up to 3 pass-through sessions can be joined in that they all connect to the same serial port. Data arriving on the serial port gets passed through to all parties, and data arriving from any one party gets passed through to the connected serial port as well as the other parties. Joinable pass-through can be enabled/disabled in the General Settings menu, or the **`sys.pt.joinable`** Setting Key.

Buffered pass-through

Buffered pass-through is where upon connecting to a pass-through session, the first thing the unit does is dump all data that has been buffered in that port's database file, instead of connecting to the port right away. Once all data from that file is output then unit connects you to the port. If no data has been buffered (or this feature is turned off) then the unit initially connects you to the port. This option is not available on the T850.

Serial Break

The T850 gives a pass-through client the ability to apply the 'serial break condition’ on any passthrough serial port. A serial break can be a "wake up“ signal to a device connected to any of the T850 serial ports. This feature allows the user to set: the ASCII character to be used for the break, and; the maximum number of times during the current pass-through session the connected device will recognize that character as the break. After that number of times, that character will not be interpreted as a break. This also allows the client to, within the same pass-through session, load binary data files that may include the break character without unintentionally applying the break condition.

Each serial port may be configured independently of the others by use of two Setting Keys:

```
serial[].pt.breakchar (default 0)
serial[].pt.breakcount (default 1)
```

Example:

For example, say you have some device on I/O 6 that requires the serial break condition to wake up. If you access the unit and enter pass-through mode to I/O 6, and you want to enter Ctrl-Break to apply the break condition, and have it do that just once per pass-through session, configure this:

```
serial[6].pt.breakchar=3
serial[6].pt.breakcount=1
```

Ctrl-Break, at least on Windows PCs, sends ASCII character 0x03 down the wire, so this is why you would set the breakchar to 3.

By default the unit provides pass-through access to anyone and can be further defined in the User Profile Settings menus. Various settings control its behavior, as discussed above with each sub-feature.

Call Failure Tracking

Description
Call failure tracking is a feature added for A-tick compliance that limits the number of times the T850 calls any one number that doesn't appear to work. Each number dialed is tracked for how many consecutive failures it has racked up. Each time a call is attempted, this number's failure count is checked before dialing. If the failure count >= 15 then the number will not be dialed for until reset or its blackout period expires. After dialing, if the call is a failure then the called number's failure count is incremented. When it increments to 15 then a blackout timer is set for 2 hours, meaning that this number is forbidden to be dialed for the next 2 hours.

"Call is a failure" means:

- for ppp, ppp was not negotiated
- for other modem calls and alphanumeric pages, carrier was not negotiated.
-

Numeric pages do not fail to dial since nothing is actually negotiated.

After dialing, if the call is successful then called number's failure count is set to 0.

Benefit
This enables the unit to not continually dial a number if the number has been shown to be unresponsive, in order to be a good citizen on the telephone network.

Configuration
There are no settings or UI associated with this feature.

Usage
If a number has reached its failure limit (and thus turned into a forbidden number to dial) then a message is appended to the Audit Log. Any future attempt to dial a forbidden number results in a message appended to the Audit Log. The only way to make the unit dial any forbidden number again is wait until the 2-hour blackout expires for that number or reset the unit (power cycle, **RESTART** command, **RESTART ALL** command, push reset button). When dialing is attempted after the blackout period expires then a message is appended to the Audit Log saying that forbidden number x was granted permission to be dialed again.

RADIUS Security

Description
RADIUS (Remote Authentication Dial In User Service) is feature is used to offload authentication, authorization, and accounting (AAA) work to a RADIUS server, instead of doing that work on the unit. Prior to the introduction of the RADIUS feature, AAA was done on the unit via the User Profiles settings and the Audit Log, although it was never explicitly called AAA in our documentation up to this point. With the introduction of the RADIUS feature, AAA can now be done with a RADIUS server via the RADIUS protocol. A RADIUS server is one instance of a AAA server in that it offers authentication, authorization, and accounting services to client machines, such as the unit. The next few sections go into more detail about how the RADIUS feature works.

Overview
The RADIUS feature is enabled by setting the **sec.mode** Setting Key to RADIUS or setting the Security Settings/Security Mode option to RADIUS. You configure a primary and/or secondary RADIUS server address (or hostname), as well as secrets for each. The secret is for authenticating the network traffic between the unit and the RADIUS server. The unit makes transactions with the RADIUS server in order to:

- authenticate a user (Authentication)
- determine what an authentic user is authorized to do (Authorization)
- log information about when an authentic user started and stopped a login session (Accounting)

Each transaction has a timeout that specifies how long the unit will wait for a response from the server. (This is configured with the **sec.radius.timeout** Setting Key or in the RADIUS Security Settings menu.) "A response from the server" means a response that is authentic; i.e., the response network frame is verified as trusted. If a response is not authentic, it could be due to an attacker, or corrupted network frame, or misconfiguration of the server secret. A server can respond but if the secret is configured wrong then the unit will find it not authentic, and silently discard the response. In this case, it is as if the unit had received no response at all. So from the perspective of the unit, a response from a RADIUS server is one that is both received ***and*** authentic.

If no response arrives after the timeout, or if the unit could not transmit to the server in the first place (the server was unreachable, because, for example, no network link, or no network configured on the unit), the unit can try again, up to a limit as configured with **sec.radius.retries** Setting Key or in the RADIUS Security Settings menu. If the unit exhausts all retries for authentication/authorization transactions, it has three options determined in this order:

1. try the same transaction with the secondary server (if its address/hostname and secret are configured). If the secondary server responds, authentication/authorization will succeed/fail according to that server's response. In any other case (secondary server unconfigured or configured but unreachable), the unit proceeds to step 2.

2. try to authenticate and authorize the user using the local User Profiles configuration (if its configured, when **sec.radius.fallback.mode**=`USER PROFILES`). If the user fails to authenticate with the User Profiles configuration (or if **sec.radius.fallback.mode**=`NONE`) then the unit proceeds to step 3.

3. give up; the unit cannot authenticate the user so the user cannot log in.

If a RADIUS server deems a user authentic then it passes back authorization info to the unit. So authentication and authorization happen in one transaction. Accounting happens in a separate transaction. Once the unit sees that an authentic user is authorized to do what they intend to do, the unit sends a RADIUS accounting start message to the RADIUS server that originally authenticated the user. When the user's session ends, the unit sends an accounting stop message to that same server.

In sum, the RADIUS feature enables the unit do AAA transactions with a RADIUS server in order to:

- determine if a user is actually who they claim to be
- determine if a user is authorized to do what they want to do, and
- log when that user starts and stops their session

The remaining subsections discuss details of each part of AAA.

Authentication

The RADIUS feature enables the unit to offload (and centralize) user authentication responsibilities to a RADIUS server. The unit does this for the following services in Phase 2 implementation:

- Local (console) command processor
- Telnet command processor
- Modem command processor
- Telnet pass-through
- Real-time sockets
- FTP
- Web UI

» **Note:** Phase 3 implementation will support PPP while Phase 4 will support SSH. Neither Phase 3 nor Phase 4 are supported in this version of the T850.

When the unit uses the USER PROFILES security mode, there can be at most 12 users configured, and the unit must be configured with authentication and accounting details. With RADIUS security mode however, as many users can log in to a unit as can be supported on the RADIUS server, and a manner completely independent of the User Profiles configuration on the unit. Additionally, the unit may be just one of many machines that a user would need access to. If all machines supported AAA, user management can be configured more easily and centrally via the RADIUS server, instead of at the unit or other machines configured with their own security mechanisms.

PAP vs CHAP

Authentication can happen via PAP (Password Authentication Protocol) or CHAP (Challenge-Handshake Authentication Protocol). Configured **`sec.radius.chap`**`=ON` for CHAP, or `OFF` for PAP.

PAP is where the user provides a username and password. Both the username and password are transmitted to the unit from the user in clear text (unless protected by the application layer's security, such as SSL (for the web UI) or SSH). The username is transmitted to the RADIUS server from the unit in clear text (the password is not).

CHAP is more complex but more secure because the password is not transmitted to the unit from the user (unlike PAP). Instead, the unit first provides the user with a CHAP challenge. The user provides the username, CHAP ID, and CHAP response (which is generated from both the challenge and the user's password). The user uses some local program to generate a CHAP response based on the user's password, CHAP ID, and CHAP challenge. The CHAP ID is just a number between 0 and 255 that the user chooses and provides to both the unit and the CHAP-response-generating program. The unit passes the challenge, username, CHAP ID, and CHAP response to the RADIUS server, which then authenticates the user based on this data.

When logging in to the command processor, pass-through, Web UI, or real-time sockets, the user is prompted for three things when CHAP is enabled: username, CHAP ID, and CHAP response. When logging in to the FTP server, the UI is more standardized as "username and password" and hence requires some special attention when using CHAP. In the case of logging in to the unit via FTP, enter as the FTP password the concatenation of the ASCII-hex CHAP ID value and CHAP response. For example, if the user chooses CHAP ID 225 and generates CHAP response DD0F3C51116B74CFFEC4379BA6D03507, then the FTP password is 225 in ASCII-hex (which is "E1") concatenated with that response: E1DD0F3C51116B74CFFEC4379BA6D03507.

For all login services, the CHAP challenge is presented as a 32-byte ASCII-hex value, representing 16 bytes of the actual challenge value. This is so the challenge can be a pseudo-random bit sequence of the same size as the RADIUS frame authenticator, and also cut-and-pastable by the user between their login UI and their CHAP-response-generating program.

In sum, PAP is as simple as traditional authentication methods. CHAP is more secure but more complex and requires the user to have a local CHAP-response-generating program. This program is anything that can create a 16-byte MD5 hash of the CHAP ID (as an 8-bit value), user password, and challenge (as a 16-byte value).

Authorization

Once a RADIUS server deems a user is authentic, its necessary to determine what the user is authorized to do. For example, a certain user may be, on the RADIUS server, configured and authorized to log in to the unit via telnet command processor but not via the web UI. So if that user attempts to log in to the unit via the web UI, they will be authenticated by the RADIUS server, but denied access by the unit. This happens because upon authentication, the unit requires the RADIUS server to send it certain authorization data about the user. (If the RADIUS server does not respond with all the required authorization data, the user is not allowed to log in to the unit, even though they were authenticated by the RADIUS server.) The authorization data received by the unit essentially says "this user is not allowed access via the web UI". The unit interprets this data by rejecting the user's web UI login attempt. To remedy, the configuration on the RADIUS server would have to change to allow web UI access for that user. This is an example of just one of the pieces of authorization data that the unit requires. The full set of data is detailed later in this document.

When configuring users for access, be sure to limit their user rights (i.e., authorize them for sub-MASTER rights). MASTER users have enough privilege to change the security settings on the unit, including creating their own user profiles and changing the security mode away from RADIUS. If a user connects via RADIUS and is given MASTER rights, then that user can change the security settings to fit what may be malicious intent. Rights are allocated by the Asentria-User-Rights vendor-specific attribute defined later in this document.

Accounting

When a user is authentic and authorized, the unit sends RADIUS accounting start and accounting stop messages to the RADIUS server that authenticated the user, when that user's login session begins and ends, respectively. If the RADIUS accounting UDP port **`sec.radius.acct.port`** is set to 0 then the unit will not send accounting information. For example, when a user logs in with RADIUS (in PAP mode) to the console port, the unit does the following four things to or for the user:

1. authenticates
2. authorizes
3. sends accounting start information
4. starts a command processor

When the command processor session ends (either by the user explicitly disconnecting or lowering the handshaking on the RS232), then the unit sends accounting stop information to the RADIUS server that authenticated that user (but only if the unit had successfully sent accounting start information for that user when they logged in). Accounting information being "successfully sent" means the unit could reach the RADIUS server and the server responded.

When the unit sends the RADIUS server accounting start and stop messages, it is actually sending RADIUS Accounting-Request frames with the following RADIUS attributes:

- Standard attribute: Acct-Status-Type, which is integer 1 for start or 2 for stop.
- Standard attribute: Acct-Session-Id: the unit uses an RFC 4122 GUID as the value for this attribute; it is used to correlate start and stop messages.
- Standard attribute: User-Name (to specify who logged in or logged out)
- Vendor-specific attribute: Asentria-Service-Type, which is a string that describes the kind of login session the user started.

Limits of support

The unit does not support RADIUS Access-Challenge frame (which the RADIUS server can send in response to an Access-Request frame); the unit interprets Access-Challenge as Access-Reject.

The unit does not support any Accouting-Request frames other than those with Acct-Status-Type set to 1 or 2.

SNMPv3 works only with users specified in the User Profiles configuration when the security mode is set to USER PROFILES; SNMPv3 does not work with RADIUS.

Locking yourself out

Be careful when you are configuring RADIUS, you may lock yourself out of the unit, which means there is no way to gain access to the unit again: you must return it in order for it to be reinitialized at the factory. There are four ways around this:

1. If you are locked out because there is something wrong with the primary RADIUS server (i.e., it is reachable but it is incorrectly rejecting authentication requests), then configure a secondary (redundant) one, if you have the resources for that.

2. The unit attempts to detect an invalid RADIUS configuration, and if it finds it, it automatically authenticates you using User Profiles. An invalid RADIUS configuration is one where (primary server or secret is not configured) and (secondary server or secret is not configured). So if you have misconfigured the unit in this way, you can still get into the unit provided you know the credentials for a MASTER-rights user profile.

3. Configure the unit to fall back to User Profiles (**`sec.radius.fallback.mode`**`=USER PROFILES`). This means when all RADIUS servers configured are unreachable or reachable but unresponsive, the unit will authenticate and authorize the user with its User Profiles configuration. If any RADIUS servers (primary or secondary) are responsive, then when they reject a user, the unit will reject a user and ***not*** fall back to authenticating with User Profiles. On the one hand this is an insurance policy against locking yourself out, but on the other hand it still means you must maintain some local authentication/authorization security configuration of the unit, which erodes the purpose of centralized AAA.

4. If you end up in a situation where you cannot log in to the unit at all, there is one last resort before returning the unit. There is a way to gain access with the button unlock feature. If you tap the reset button a few times (at least 5) until the front panel lights flash, then the unit defaults the following settings, which enables you to log in to the unit via the console port using the default MASTER user profile:
    - **`sec.mode`** (to USER PROFILES)
    - **`sec.consolereq`** (to OFF)
    - **`sec.connectvia`** (to every method of connecting)
    - "admin/password/MASTER" credentials for the user profile appropriate to the product
    - IO2 mode set to COMMAND (if applicable to product)

Note:

- The button unlock feature can only be used if **`sec.button.unlock`**`=ON` (which it is by default). If you do not want the unit to grant access via this feature, then turn it off. However, if you subsequently lock yourself out then there is no way to gain access to the unit: you must return it.

- If you lock yourself out and gain access again with the button unlock feature, remember to reconfigure the settings that were defaulted by the button unlock feature to maintain your prior security configuration!

- "tap the reset button" means press the reset button on the unit (the only button for the current products) until it clicks and then release it, at a frequency of about 1-2 taps per second. Do not hold in the reset button otherwise that will reset the unit, just tap it like you click a mouse button.

RADIUS server configuration

Some configuration for the RADIUS server is vendor-dependent, such as how you configure client machines and users. Likewise there is vendor-independent configuration that tells the RADIUS server what vendor-specific RADIUS attributes should be included in Access-Accept frames. All authorization data is encapsulated by these vendor-specific attributes in a file called the RADIUS dictionary. The Asentria RADIUS dictionary (named dictionary.asentria) is included on the resource CD that ships with the unit, or can be requested from Asentria Technical Support. It is meant to be input into your RADIUS server. The attributes are listed below. When you configure a user on the RADIUS server, you must in some way specify values for these attributes -- this is how you tell the RADIUS server (and the unit) explicitly what a user is authorized to do. The values for each attribute correspond exactly to the traditional settings used on the unit for User Profiles authorization.

Attribute	Allowed values	Corresponding User Profiles Setting	Required by connection method
Asentria-Connect-Via-Local	ON,OFF	sec.user[x].connectvia.local	L
Asentria-Connect-Via-Modem	ON,OFF	sec.user[x].connectvia.modem	M
Asentria-Connect-Via-Telnet	ON,OFF	sec.user[x].connectvia.telnet	TP
Asentria-Connect-Via-FTP	ON,OFF	sec.user[x].connectvia.ftp	F
Asentria-Connect-Via-RTS	ON,OFF	sec.user[x].connectvia.rts	R
Asentria-Connect-Via-SSH	ON,OFF	sec.user[x].connectvia.ssh	N/A in phase 2
Asentria-Log-In-To	COMMAND, PASSTHROUGH, MENU	sec.user[x].loginto	FTMLP
Asentria-Access-File	FILE1, FILE2, ... FILEn	sec.user[x].accessfile	TML
Asentria-PPP-Type	NONE, LOCAL, ROUTING	sec.user[x].ppptype	N/A in phase 2
Asentria-User-Rights	NONE, VIEW, ADMIN1, ADMIN2, ADMIN3, MASTER	sec.user[x].rights	FTMLPW
Asentria-File1-Read-Access	DENY, ALLOW	sec.user[x].file[1].readaccess	FTMLWR
Asentria-File2-Read-Access	DENY, ALLOW	sec.user[x].file[2].readaccess	FTMLWR
Asentria-File3-Read-Access	DENY, ALLOW	sec.user[x].file[3].readaccess	FTMLWR
Asentria-File4-Read-Access	DENY, ALLOW	sec.user[x].file[4].readaccess	FTMLWR
Asentria-File5-Read-Access	DENY, ALLOW	sec.user[x].file[5].readaccess	FTMLWR
Asentria-File6-Read-Access	DENY, ALLOW	sec.user[x].file[6].readaccess	FTMLWR
Asentria-File7-Read-Access	DENY, ALLOW	sec.user[x].file[7].readaccess	FTMLWR
Asentria-File8-Read-Access	DENY, ALLOW	sec.user[x].file[8].readaccess	FTMLWR
Asentria-File9-Read-Access	DENY, ALLOW	sec.user[x].file[9].readaccess	FTMLWR
Asentria-File10-Read-Access	DENY, ALLOW	sec.user[x].file[10].readaccess	FTMLWR
Asentria-File11-Read-Access	DENY, ALLOW	sec.user[x].file[11].readaccess	FTMLWR
Asentria-File12-Read-Access	DENY, ALLOW	sec.user[x].file[12].readaccess	FTMLWR
Asentria-File13-	DENY, ALLOW	sec.user[x].file[13].readaccess	FTMLWR

Read-Access			
Asentria-File14-Read-Access	DENY, ALLOW	sec.user[x].file[14].readaccess	FTMLWR
Asentria-File15-Read-Access	DENY, ALLOW	sec.user[x].file[15].readaccess	FTMLWR
Asentria-File16-Read-Access	DENY, ALLOW	sec.user[x].file[16].readaccess	FTMLWR
Asentria-Events-Read-Access	DENY, ALLOW	sec.user[x].events.readaccess	FTMLWR
Asentria-Audit-Read-Access	DENY, ALLOW	sec.user[x].audit.readaccess	FTMLWR
Asentria-File1-Write-Access	DENY, ALLOW	sec.user[x].file[1].writeaccess	FTMLWR
Asentria-File2-Write-Access	DENY, ALLOW	sec.user[x].file[2].writeaccess	FTMLWR
Asentria-File3-Write-Access	DENY, ALLOW	sec.user[x].file[3].writeaccess	FTMLWR
Asentria-File4-Write-Access	DENY, ALLOW	sec.user[x].file[4].writeaccess	FTMLWR
Asentria-File5-Write-Access	DENY, ALLOW	sec.user[x].file[5].writeaccess	FTMLWR
Asentria-File6-Write-Access	DENY, ALLOW	sec.user[x].file[6].writeaccess	FTMLWR
Asentria-File7-Write-Access	DENY, ALLOW	sec.user[x].file[7].writeaccess	FTMLWR
Asentria-File8-Write-Access	DENY, ALLOW	sec.user[x].file[8].writeaccess	FTMLWR
Asentria-File9-Write-Access	DENY, ALLOW	sec.user[x].file[9].writeaccess	FTMLWR
Asentria-File10-Write-Access	DENY, ALLOW	sec.user[x].file[10].writeaccess	FTMLWR
Asentria-File11-Write-Access	DENY, ALLOW	sec.user[x].file[11].writeaccess	FTMLWR
Asentria-File12-Write-Access	DENY, ALLOW	sec.user[x].file[12].writeaccess	FTMLWR
Asentria-File13-Write-Access	DENY, ALLOW	sec.user[x].file[13].writeaccess	FTMLWR
Asentria-File14-Write-Access	DENY, ALLOW	sec.user[x].file[14].writeaccess	FTMLWR
Asentria-File15-Write-Access	DENY, ALLOW	sec.user[x].file[15].writeaccess	FTMLWR
Asentria-File16-Write-Access	DENY, ALLOW	sec.user[x].file[16].writeaccess	FTMLWR
Asentria-Events-Write-Access	DENY, ALLOW	sec.user[x].events.writeaccess	FTMLWR
Asentria-Audit-Write-Access	DENY, ALLOW	sec.user[x].audit.writeaccess	FTMLWR

Asentria-Port1-PT-Access	DENY, ALLOW	sec.user[x].port[1].ptaccess	TMLWP
Asentria-Port2-PT-Access	DENY, ALLOW	sec.user[x].port[2].ptaccess	TMLWP
Asentria-Port3-PT-Access	DENY, ALLOW	sec.user[x].port[3].ptaccess	TMLWP
Asentria-Port4-PT-Access	DENY, ALLOW	sec.user[x].port[4].ptaccess	TMLWP
Asentria-Port5-PT-Access	DENY, ALLOW	sec.user[x].port[5].ptaccess	TMLWP
Asentria-Port6-PT-Access	DENY, ALLOW	sec.user[x].port[6].ptaccess	TMLWP
Asentria-Port7-PT-Access	DENY, ALLOW	sec.user[x].port[7].ptaccess	TMLWP
Asentria-Port8-PT-Access	DENY, ALLOW	sec.user[x].port[8].ptaccess	TMLWP
Asentria-Port9-PT-Access	DENY, ALLOW	sec.user[x].port[9].ptaccess	TMLWP
Asentria-Port10-PT-Access	DENY, ALLOW	sec.user[x].port[10].ptaccess	TMLWP
Asentria-Port11-PT-Access	DENY, ALLOW	sec.user[x].port[11].ptaccess	TMLWP
Asentria-Port12-PT-Access	DENY, ALLOW	sec.user[x].port[12].ptaccess	TMLWP
Asentria-Port13-PT-Access	DENY, ALLOW	sec.user[x].port[13].ptaccess	TMLWP
Asentria-Port14-PT-Access	DENY, ALLOW	sec.user[x].port[14].ptaccess	TMLWP
Asentria-Port15-PT-Access	DENY, ALLOW	sec.user[x].port[15].ptaccess	TMLWP
Asentria-Port16-PT-Access	DENY, ALLOW	sec.user[x].port[16].ptaccess	TMLWP
Asentria-Service-Type	LOCAL, MODEM, TELNET, PASSTHROUGH, FTP, RTS, WEB, PPP, SSH	N/A	N/A

The final column, "Required by connection method", lists the connection methods that require the attribute. Here is what the letters mean for this column:

- **F**=FTP
- **T**=Telnet command processor
- **M**=Modem command processor
- **L**=Local (console) command processor
- **W**=Web UI
- **R**=Real time sockets
- **P**=Telnet pass-through (to port 210x)

For example, Asentria-Access-File has "TML", which means if you configure a user on the RADIUS server that you intend to connect by Telnet, Modem, or Local, then you ***must*** configure this attribute to be returned to the unit upon successful authentication, otherwise the unit cannot authorize the user, and will therefore reject the user's login even though they are authentic.

The Asentria-Service-Type attribute is N/A for the last two columns because it does not deal with authorization -- it is used in accounting RADIUS transactions only.

Note that the Asentria-Filex-* and Asentria-Portx-* attributes are required for only however many serial ports on the unit. For example, if you have a unit with only 2 ports, then only Asentria-File1-*, Asentria-File2-*, Asentria-Port1-*, and Asentria-Port2-* attributes are required by that unit for the given connection method.

Note that "N/A in phase 2" means that this attribute is not used in phase 2 of the RADIUS feature (phase 2 supports everything except PPP and SSH).

Benefit

In a typical application environment for these units, there is hardware from other vendors too, and each piece of hardware probably has its own way of doing AAA operations. As the number of disparate machines rises, so does the administration headache of maintaining AAA for each machine for each user. If all machines use a standard, centralized AAA architecture however, then that simplifies administration of all of them and makes each one fit more easily in into the entire application environment. Therefore, having a unit support AAA (via RADIUS, one of the most-deployed and most-mature of AAA servers) makes it easier for organizations to fit units into their environments.

Configuration

To configure RADIUS on the unit (minimum required configuration) enter the Setting Key values as shown below, or onfigure using the RADIUS Security Settings menu:

```
sec.mode=RADIUS
sec.radius.server[1]=<address or hostname>
sec.radius.server[1].secret=<secret>
```

To configure other parts of RADIUS (optional):

```
sec.radius.server[2]=<address or hostname>
sec.radius.server[2].secret=<secret>
sec.radius.fallback.mode=<NONE or USER PROFILES>
sec.radius.auth.port=<UDP port that server uses for authentication/authorization>
sec.radius.acct.port=<UDP port that server uses for accounting, or 0>
sec.radius.chap=<ON or OFF>
sec.radius.timeout=<timeout in seconds, 1 to 30>
sec.radius.retries=<number of retries, 0 to 30>
```

Example

Say you want to configure user "bob" to access the unit's modem command processor via RADIUS. First configure "bob" on the RADIUS server. He may already be configured on your RADIUS server because his duties may include administering other RADIUS-supporting machines besides the unit. Either way, you must configure the following attributes for "bob" on the RADIUS server (this list is generated by looking at the table above and seeing which attributes are required by the "T" method (telnet command processor). (Say the unit has only 2 serial ports to minimize the File/Port authorization attributes listed here.)

```
Asentria-Connect-Via-Telnet = ON
Asentria-Log-In-To = COMMAND
Asentria-Access-File = FILE1
Asentria-User-Rights = ADMIN3
Asentria-File1-Read-Access = ALLOW
Asentria-File2-Read-Access = ALLOW
Asentria-File1-Write-Access = ALLOW
Asentria-File2-Write-Access = ALLOW
Asentria-Events-Read-Access = ALLOW
Asentria-Audit-Read-Access = ALLOW
Asentria-Events-Write-Access = DENY
Asentria-Audit-Write-Access = DENY
Asentria-Port1-PT-Access = ALLOW
Asentria-Port2-PT-Access = ALLOW
```

This list of attributes for user "bob" on the RADIUS server specifies that he can access the unit's Telnet command processor with ADMIN3 rights, the access file set to FILE1 and all files/ports readable and writable except that he cannot write the Events and Audit files.

Also configure a user for yourself that gives you MASTER rights to the unit should you need access to it.

Then configure RADIUS on the unit according to the Configuration section above, verify the unit can reach the RADIUS server by pinging it, and then log out. Then try logging in to test the RADIUS setup. If you or "bob" cannot log in then you have locked yourself out of the unit. If the reason you cannot log in cannot be attributed to a configuration error on the RADIUS server then you must use the unit's fallback options for getting access to the unit again: the RADIUS fallback mode or the button unlock feature. From there troubleshooting steps can be taken to see why login failed.

Please contact Asentria Technical Support for assistance in troubleshooting RADIUS connection problems.

Data Events

This section offers a brief tutorial on how to set up a functional data event that will send an SNMP trap when the word "test" is received over a data port. Full details on how to configure data alarm equations are available in the next section, Configuring Data Alarm Equations.

Set Up a Data Event

1. From the command prompt, access the Setup menu. Select "Alarm/Event Definitions", "Data Alarm/Filter Settings", and then "Data Alarm Field Settings". The following menu allows a user to define up to 16 data event fields to be used when scanning for event data. Below is an abbreviated example of this menu:

```
TeleBoss 850 - Data Alarm Field Definition Table
                     Start    Length   Line     Type       Name
A) Definition A      0        0        0        [Alpha]
B) Definition B      0        0        0        [Alpha]
   ...
O) Definition O      0        0        0        [Alpha]
P) Definition P      0        0        0        [Alpha]
```

2. Select field A. The menu in the following example will be displayed.

```
TeleBoss 850 - Data Alarm Field Definition
Data Field: A
A) Start Position                       [0]
B) Field Length                         [0]
C) Field Name                           []
D) Field Line Number                    [0]
E) Field Type                           [Alpha]
```

3. Select Start Position. When prompted to enter a new value, enter "1" and press <Enter>.
4. Select Field Length. When prompted to enter a new value, enter "4" and press <Enter>.
5. Select Event Name and enter **TEST_FIELD**, then press <Enter>.
6. Press <Enter> to return to the Field definition Table. If configured properly, the data event field should appear in this menu.
7. Press <Enter> to return to the Data Alarm/Filter Settings menu. From here, select the Data Alarm Settings menu, Alarm/Filter Page 1, then Alarm/Filter 1. The following menu will be displayed:

```
TeleBoss 850 - Settings For Data Alarm/Filter 1
A) Alarm/Filter Enable                  [OFF]
B) Alarm/Filter Mode                    [ALARM]
C) Alarm/Filter Name                    []
D) Alarm/Filter Equation                []
E) Threshold                            [1]
F) Auto-Clear when Threshold Reached    [ON]
G) Alarm Counter Clear Interval         [12 HOURS]
H) Alarm Counter Reset Time             [00:00]
I) Actions                              []
J) Class                                [Info]
K) Data Alarm Trap Number               [503]
L) Clear This Alarm Counter Now
```

8. Press “A” to toggle Alarm/Filter Enable to ON.
9. Alarm/Filter Mode should be set to ALARM. If it is set to FILTER, press “B”.
10. Select Alarm/Filter Name and enter **Test Event 1**.

11. Select Alarm/Filter Equation and enter **TEST_FIELD="test”**. This will cause an event to occur any time the word "test" is received.
12. Select Actions and enter "**TRAP(1)**" to cause this data event to send a trap to SNMP manager #1, as configured below in the Hostname/IP Address menu.

Other Setup

1. Return to the Main Setup Menu, select “Action Definitions”, select “Hostname/IP Address 1” and enter either the hostname or IP address of the SNMP Manager where the trap will be sent.
2. Go to the Serial Setup Menu for serial port I/O 1 (or whichever port incoming data will be monitored) and set the Data Alarm Enable setting to ON.
3. Press <CTRL> + C to return to the command processor.

Testing

Connect to the unit serially on I/O 1 and type the word **test** followed by <Enter>. This should trigger the above data event, and an SNMP trap should be sent to SNMP Manager #1. If this is not the case, double check the network and data event settings and then call Asentria Technical Support.

» **Note:** There will be a 30 second delay in alarming if the terminal emulator being used does not send a LF with the CR. This may be circumvented by pressing <CTRL + J> to generate a LF.

Configuring Data Alarm Equations

The equation is the heart of any data event. The following are a few examples event equations:

- `alarm_code = "L31"`
- `ext >= "A 600" AND exit_code = "DN"`
- `(alarm_code > "1051" OR exit_code = "1Ow74x") AND switch = " 001.1.9*.**"`
- `@ = "CRITICAL"`

Here are a few tips to help you create your own data event equations:

- Multiple field references are acceptable, as long as both fields are the same length. For example, d=c is a valid equation if the fields that both 'd' and 'c' represent are two characters long
- Variable names are case sensitive
- Equation literals (the data contained within quotation marks) are case sensitive
- If any rule is violated in a equation, an alarm will not be generated, nor will an error be presented

Note: There may be times when two or more fields are necessary to analyze one piece of data. For example, if a time is represented in hh:mm format, some calculations may require two different fields. Other times, wildcards will do the job of masking out non-important characters just fine.

The data alarm equations used in the T850 are standard Boolean-type operators. The following table outlines each of the supported operators and their function.

Operator	**Function**
>	Greater Than
<	Less Than
>=	Greater Than or Equal to
<=	Less Than or Equal to
! or <>	Not Equal to
=	Equal to
*	Single character wildcard (matches any character or space)
()	Parenthesis used to combine operations
OR	Logical OR
AND	Logical AND
@	Positional wildcard (used in place of a field name to match anywhere within an incoming record)

Data Alarm Macros

Data alarm macros provide a way to define up to 100 equations that can be used in one or more data alarm equations. Each macro consists of an equation and an associated name that can be used to reference the macro in a data alarm equation. They simplify the creating of data alarm events, particularly where more than one event uses the same expression in its equation. Also, since the macro expression is evaluated only once per record, it improves the efficiency of alarm processing.

Data alarm macros can be configured using the setup menu or setting keys:

Menu

Setup -> Alarm/Event Definitions -> Data Alarm/Filter Settings -> Data Alarm Macro

Settings Keys

```
event.macro[].name
event.macro[}.equation
```

The macro equation is entered the same way as a data alarm equation. A macro equation cannot refer to another macro; in such a case, the expression involved will always evaluate to FALSE. The macro equation can be up to 160 characters in length.

The macro name is the name by which the macro is referenced in any data alarm equation, and can be up to 16 characters in length. Macro names are subject to these restrictions:

- Macro names and data field names are not case sensitive; therefore DLT35 and Dlt35 are equivalent.
- A macro cannot be given the same name as a data field or another macro.
- The following names are reserved and should not be used as macro names or data field names:

°IOx (where x is a number)	°FALSE
°IPRC	°AND
°TRAP	°OR
°FTP	°IS
°TRUE	°ISNOT

Using a macro name or data field name that starts with AND or OR will cause that part of the expression to always evaluate to FALSE.

Macro names and data field names cannot start with $.

When used in a data alarm equation, macros are always compared to TRUE or FALSE. Any other comparison yields a result of FALSE.

Example

Settings

- `event.data[1].enable=ON`
- `event.data[2].enable=ON`
- `event.data[1].equation=m1=true`
- `event.data[2].equation=m1 = true and f2 = "0"`
- `event.field[1].start=7`
- `event.field[2].start=6`
- `event.field[1].length=1`
- `event.field[2].length=1`
- `event.field[1].name=f1`
- `event.field[2].name=f2`
- `event.macro[1].name=m1`
- `event.macro[1].equation=f1="1"`

Incoming records

```
0000001 N 019 00 DN1042  T001034         02/25 09:21 00:00:50 A 5558481677
0000002 N 020 00 DN5280  T001033         02/25 09:22 00:00:08 A 5551377443
0000003 N 021 00 T002014 DN6502          02/25 09:22 00:00:10
0000004 N 022 00 T007002 DN5700          02/25 09:19 00:02:36
0000005 E 023 00 T002024 DN1006          02/25 09:22 00:00:58
0000006 N 024 00 T002042 DN6000          02/25 09:21 00:00:46
0000007 N 025 00 DN5154  T001035         02/25 09:04 00:17:50 A 5558451000
0000008 N 026 00 DN1192  T001031         02/25 09:22 00:01:10 A 5558406776
0000009 N 027 00 DN1048  T001034         02/25 09:23 00:00:26 A 5556426898
0000010 N 028 00 DN1197  T001020         02/25 09:19 00:04:30 A 5552550948
0000011 N 029 00 DN6063  T001033         02/25 09:23 00:00:16 A 5557458535
0000012 N 030 00 T002019 DN6447          02/25 09:23 00:00:10
```

Alarm records

```
0000001 N 019 00 DN1042  T001034         02/25 09:21 00:00:50 A 5558481677 (DA 1)
0000001 N 019 00 DN1042  T001034         02/25 09:21 00:00:50 A 5558481677 (DA 2)
0000011 N 029 00 DN6063  T001033         02/25 09:23 00:00:16 A 5557458535 (DA 1)
```

- The first record matches data alarm 1, because macro 'm1' is true. Macro 'm1' is true any time the character in the 7th position is '1'.
- The first record also matches data alarm 2, because macro 'm1' is true and field 'f2' contains a '0' character.
- The eleventh record matches data alarm 1, again because macro 'm1' is true. It does not match data alarm 2 because field 'f2' does not contain a '0' character.

Action List

An action list is a text string that specifies what the unit should do upon an event. It's comprised of a list of keywords and parameters separated by semicolon. Each keyword specifies a certain action and has its own parameter set, which is enclosed in parentheses.

» **Note:** Not all actions on the Action List may be available in this product. Check with Asentria Tech Support if you have questions concerning this.

For example, the keyword *trap* has a parameter <*ipaddress or index*>, and has syntax *trap(ipaddress or index)* in an action list. This keyword means send an SNMP trap to the specified parameter. If the parameter is an IP address then that address is the trap destination. If the parameter is an index then it uses the address specified in the corresponding index # for Hostname/IP Address in the Action Definitions menu. (This IP action setting list is **`action.ip`**, so trap(1) means send a trap to the address in setting **`action.ip[1]`**.)

- Cancel: cancel(*idname*)
  Cancel any running action list identified by *idname*.

- Dialup Pager: dpage(index)
  Send a pager callout via modem; index is the phone number configured with **`action.page.number`**

- Dispatcher: dispatch(*phone#* or *index*)
  Send a Dispatcher alarm via modem; *index* is the phone number configured with **`action.call.number`**.
  E.g., **`action.call.number[`*`index`*`]`.**

- Email: email(*email* or *index*)
  Send an email to the address specified by *email*; *index* is the email address configured with **`action.email`**

- Group: group(*groupname*)
  Identify this action list as part of a group identified by *groupname*; not currently used. In a future version this will be used to cancel or postpone groups of action lists.

- ID: id(*idname*)
  Identify this action list by *idname*.

- Inform: inform(*ipaddress or index*)
  Send an SNMP inform to a specific IP address or *index* which refers to an IP address or host name configured in the Action Definitions menu.

- Malert: malert(*phone# or index*)
  Send an malert (Asentria Alarm via modem); the parameters are the same as for the dispatch keyword.

- Modem: modem(*phone# or index*)
  Make the unit dial a phone number and start a login session (to the unit's command processor) with the answering machine. The parameters are the same as for the dispatch keyword.

- Postpone: postpone(*idname, seconds*)
  Postpone an already-running action list identified by *idname* for a duration specified by *seconds*.

- Pause: pause(*seconds*)
  Pause operation for a duration specified by *seconds*.

- Relay: relay(action, EventSensor, point)
  Put a relay in a certain state specified by *action*.
  - *action*: one of the following two words, by case-insensitive exact match or partial unambiguous match: *active* or *inactive*. "Active" always means to energize the relay.
  - *EventSensor*: the number of the EventSensor that has the specified relay, where it is the same as that referred to by the index in an EventSensor key (e.g., 1 in **`event.sensor[1].*`** for the first external EventSensor) as well as that referred to by the SNMP esIndex object.

- *point*: the number of the relay (1-based) on the specified EventSensor. E.g., this is the same number x in "**event.sensor[1].relay[x].***"

- Script: script(*action, name or number*)
  Start or stop a script
  - *action* is the case-insensitive exact match of *exec* or *kill*.
  - *name* is the registered name of the script
  - *number* is the number of the registered script

- SMS: sms(*phone# or index*)
  Send an SMS message to a specific phone number or *index* which refers to a phone number configured in the Actions Definition menu.

- Talert: talert(*ipaddress or index*)
  Send a talert (Asentria Alarm via TCP).
  - *ipaddress* is the destination machine;
  - *index* is the IP address configured with **action.ip**. E.g., **action.ip[*index*].**

- Trap: trap(*ipaddress or index*)
  Send an SNMP trap. The parameters are the same as for the talert keyword. In order to send a trap there must be a route for it. Since a trap is an unacknowledgable action, the way the unit knows if a trap is successful is if it was able to leave the unit. In order for a trap to leave the unit there must be an IP route to its host. A trap action without a route to its host is considered a failure. "Without a route" means, for example, that:
  - if the host is meant to be on a local net but cannot be ARPed
  - if the host is meant to be off all local nets but the router cannot be ARPed
  - if the above two conditions exist and PPP cannot be raised as a backup route.

- Stop if any/all actions OK: okstop(any|all)
  Conditionally stop action list processing based on the outcome of actions prior to this keyword in the action list. The parameter specifies how much of the prior actions for this even must be successful in order for the unit to stop processing the action list: any action or all actions
  Examples:
  - "inform(1);okstop(any);sms(1)" would send the sms only if the inform failed.
  - "inform(1);okstop(all);sms(1)" would send the sms only if the inform failed.
  - "inform(2);inform(1);okstop(any);sms(1)" would stop if any of the informs succeeded. I.e., it would send the sms if neither inform succeded.
  - "inform(2);inform(1);okstop(all);sms(1)" would stop if all the informs succeeded. I.e., it would send the sms if any inform failed

- Continue: continue(*id*)
  Continue any event identified by *idname* that has either paused or postponed its action processing.

Each action can take a varying amount of time depending on what's going on in the unit. E.g., a trap may take less than a second to send if there is a route for it on a network interface that is already up (like Ethernet). Otherwise, if the unit is configured to bring up PPP in case the trap cannot be sent on an already-up interface, then the trap may take a minute to send while the unit brings up PPP.

The unit starts all actions up to the first pause keyword at the same time. E.g., if you have an action list like *trap(1);email(1);modem(1);pause(60);trap(2)* then the unit will start the first 3 actions, pause for a minute, then start the last action.

Wherever you can configure an event you can configure its actions. Generally this is with the ***.actions** setting key that applies to the event you want to monitor. You can also configure email actions (in the action list syntax) for a user profile's login challenge destination (e.g., **sec.user.challenge.telnetsendto**). Not all actions are applicable to all events: relay actions can be caused only by sensor events and data events.

Clearing Actions in the Event Queue

There may be a need to clear all event actions that are in the events queue that have not yet completed. In that case, set the `event.mgmt.clear` key to any value (e.g., 0) to delete any event that has been triggered but has not yet completed its action delivery. This is a function key only. Reading this key (**sk event.mgmt.clear**) simply returns a blank value.

Types of Alarm Notices

When alarms are detected by the T850 and a notification event is warranted, you have a choice of number of different alarm methods. Specifically these are:

- SNMP Trap
- Email Alarms
- Asentria Alarms
- SMS Alarms (requires wireless modem)
- Pager Alarms (requires dialup modem)

The following section describes these messages and how to use them.

SNMP Traps

SNMP Traps are alarm notices which are sent using TCP/IP and which conform to the requirements of the SNMP protocol. In essence, the SNMP Trap is a TCP/IP alarm message using the SNMP protocol, which contains a number of name/value pairs in its payload. In this payload the “name” is an SNMP Object ID and the “value” is the value of that OID.

In the case of the T850 product, there are two defined SNMP traps that you can choose from. These traps are defined in the SNMP MIB which is provided with the T850 product (or which is available through the Asentria website or Asentria Technical Support).

The first trap is a ‘Standard’ SNMP trap. This is the original SNMP trap format supported by Asentria products. In this trap there are two name/value pairs in the trap payload; `‘siteName’` which is the sitename of the device sending the trap and `‘stockTrapString’` which is a string value which is the standard concatenated alarm message string used for this and other alarms messages in the T850.

The stockTrapString message format looks like this:

```
Date Time :: SiteName :: Sensor Pod/Bank name  :: Sensor Point Name :: Alarm Alias
```

For example, the stockTrapString might actually look like this

```
10/24 06:43 :: San Diego Site #12 :: Sensor Pod 12 :: Cabinet Temp :: Temperature Very High
```

For users familiar with SNMP, the actual SNMP MIB definintion of the Standard SNMP looks like this:

```
t850StockTempTrap TRAP-TYPE
    ENTERPRISE t850
    VARIABLES { siteName, stockTrapString }
    DESCRIPTION
        "A stock temperature trap is issued when a temperature event
        happens."
    ::= 120
```

The other kind of SNMP trap which you can use what we call a ‘User Defined Trap’. In this trap we provide for a series of traps which each have an individual “Trap number”. This can be easier to integrate with management systems because the manager can have rules setup to kick in when you get “trap # 1000” or “trap # 1001” or so on. When using User Defined Traps, the trap number to use is assigned as part of the Event Definition Setup. In the case of User Defined Traps, the payload of the trap contains a number of OID variables, essentially anything that might be relevant to the particular alarm being transmitted. If the variable is not relevant for the alarm being transmitted then that variable is null.

For users familiar with SNMP, the actual trap definintion in the SNMP MIB looks like this:

```
    t850UserTrap1000 TRAP-TYPE
        ENTERPRISE t850
        VARIABLES { siteName, esIndex, esName, trapEventTypeNumber,
            trapEventTypeName, esIndexPoint, esPointName, esID,
            clock, trapIncludedValue, trapIncludedString,
            trapEventClassNumber, trapEventClassName }
        DESCRIPTION
            "This user-defined trap is issued when an event happens that causes a
            trap with specific trap type 1000."
        ::= 1000
```

In the above there are various alarm values in this trap including the trapIncludedString referenced in the Standard Trap.

Email Alarms

Email alarms contain a concatenated alarm string which follows the format of:

```
Date Time :: SiteName :: Sensor Pod/Bank name  :: Sensor Point Name :: Alarm Alias
```

For example, a typical Email notification for a temperature alarm might look like the following:

```
  From: Asentria TeleBoss 850
  Sent: Friday, October 24, 2008 3:59 PM
  To: support@Asentria.com
  Subject: Event

  10/24 15:59 :: San Diego Site #12 :: Sensor Pod 12 :: Cabinet Temp :: Temperature Very High
```

Asentria Alarms

Version 1.1 (default) for TCP

An Asentria Alarm sent via TCP is called a Notice. An notice is a piece of data formatted in printable ASCII: a set of lines delimited by CRLF. Each line is of the format <field>: <data>CRLF. The first line has <field> = "ID" (without the quotes). The last line has <field> = "TEXTx" (without the quotes, where x is some number between 1 and 30). The particular format the describes the alarm, and is one of the actions that can be configured for each alarm. A notice that rides on TCP/IP is called a "talert", short for "TCP alert". Talerts are delivered according the the Asentria Alarm Protocol, which over tcp is just a specification of message format.

Notices ride on an IP network. The IP network is facilitated by broadband internet connection or PPP in this model. When riding on a network from a unit to SitePath, it is assumed that a notice is normally tunneled over a VPN via a VPNG. In situations where the VPN is unavailable, the notice rides on a PPP link to SitePath via the PPPG. When riding on a network from a VPNG to the notice receiver (or on a network from a PPPG to the notice receiver), a notice travels in plaintext (i.e., not encrypted).

The format below is common to all events that can trigger a notice:

```
<Answer string (i.e., the value of sys.answer)>
<Sitename (i.e., the value of sys.sitename)>
Asentria Alarm Notice ver. 1.1

ID : 00
Date : mm/dd/yy
Time : hh:mm:ss
TargetPort:
TargetName:
AlarmType :
AlarmMsg :
Severity : {as specified by class/severity}
AlarmNum : {the value of the trap number setting for the triggering event}
Threshold :
Current :
```

```
Text1 :



Hardware: (the value of sys.hardware)
Product: (the value of sys.product)
Version: (the value of sys.version)
Build: (the value of sys.build)
Serial #: (the value of sys.serial)
```

Note: There are 3 blank lines before "Hardware:" and 2 blank lines after "Serial #:".

Other more specific types of Asentria Alarm Notice formats are: (contact Asentria Technical Support for sample format)

- Data Alarm notice
- No-data Alarm notice
- CPE Down Alarm notice
- VPN Down Alarm notice
- VPNG Down Alarm notice

Version 1.0 for modem dialout
An Asentria Alarm can also be sent over dialup modem when the Asentria Alarm Version is set to 1.0. Details of this alarm follow:

When an Asentria Alarm is initiated, the box dials into the callout number specified by the action. Once connected, it sends a header and waits for a specific response. If the T850 receives a specific response to the header, it delivers alarms in CRC mode; otherwise, alarms are delivered in non-CRC mode. In CRC mode, each Asentria Alarm is transmitted with some extra control characters and a CRC, and the remote host is required to acknowledge each alarm in a certain format.

After all Asentria Alarms have been delivered, the box waits for 20 seconds for any type of keystroke. If a keystroke is detected, the box will present a login menu.

Initial header

Note: Please see the Control Characters appendix for more information about special characters used within this section.

Upon dialing into the receiver, the T850 will send a message similar to the following:

```
  TeleBoss 850
  Server Room B
  Asentria Alarm Notice ver. 1.00
  (CR/LF)(ENQ)
```

The first line of the output is the T850's answer string.
The second line is the T850's unit ID.
The third line indicates the version of Asentria Alarm.
The final line is the (ENQ) control code.

Non-CRC Mode
After sending the initial header, the T850 pauses for 10 seconds to wait for an ACK from the receiver. Non-CRC mode requires the Require Asentria Alarm ACKs setting to be turned off. If the T850 sees no response or the receiver replies with:

```
  (ACK)00(ACK)
```

then non-CRC mode is assumed and the sender will transmit the alarms. The control characters (SOH), (SOT), and (ETX) are not transmitted in non-CRC mode.

CRC Mode

CRC mode exists to ensure that event notifications are delivered intact. Asentria Alarms delivered in CRC mode have extra control characters and a 16-bit CRC included in each alarm to allow for error detection by the receiver. Additionally, CRC mode causes the T850 to store and later retry each alarm until a proper acknowledgement is received from the receiver.

If Require Asentria Alarm ACKs is enabled, the T850 will require a positive CRC mode response or it will disconnect and retry the call. To enable CRC, the receiver must respond with the following after the header is received:

```
(ACK)01(ACK)
```

Once CRC mode is enabled, each alarm must be acknowledged by a message in the following format:

```
(ACK)XX(ACK)
```

XX represents the alarm ID to acknowledge. The ID can be found in the first line of each record sent by the T850.

Alarm Transmission

After successfully initiating a session, alarms are delivered in the following format:

```
(SOH)ID=XX(SOT)
Date=12/25/07
Time=10:30:02
TargetPort=
TargetName=
AlarmType=Data Alarm
AlarmName=Test Alarm
Threshold=0
Severity=Critical
Text1=text record line
Text2=text record line
(ETX)XX
(CR/LF)
(CR/LF)
```

The alarm ID indicates the index number of each alarm delivered during a call. This number restarts at 1 for each new call.
The severity line represents the Class value defined for this alarm.
Up to twelve lines of Text*n* may be sent.
XX represents the 16-bit CRC if CRC mode is enabled. If not, this line will contain two spaces.

If additional alarms are queued to send in the same transmission, the above output is repeated, and the ID incremented with each alarm. When non-CRC alarm transmission is selected, alarms are sent with a 5 second delay between each. When all alarms and been transmitted, then T850 sends the following:

```
(EOT)
(CR/LF)
(CR/LF)
```

At this point, the T850 waits 20 seconds for the receiver to send any input, and then hangs up. If any commands are received, a command prompt is established and the connection will remain active.

Action Definition

Asentria Alarm actions are designated by "Modem" in action definitions. The numbers correspond to callout numbers.
Example: Modem(1), Modem(2), etc

SMS Alarms

» **Note:** SMS Messaging is only supported with an EDGE wireless modem installed in the T850.

SMS alarm messages contain a concatenated alarm string which follows the format of:

```
Date Time :: SiteName :: Sensor Pod/Bank name  :: Sensor Point Name :: Alarm Alias
```

For example, a typical SMS notification for a temperature alarm might look like the following:

```
10/24 15:59 :: San Diego Site #12 :: Sensor Pod 12 :: Cabinet Temp :: Temperature Very High
```

SMS alarm messaging has the following limitations:

- The user cannot specify the order of event message items
- The user CAN specify which items are included in event message using the existing mechanism
- The event class is not included
- If the event message is too large to fit into the allowed SMS message size, it will be broken up into multiple SMS messages

Pager Alarms

» **Note:** requires dialup modem

Pager alarm messages contain a concatenated alarm string which follows the format of:

```
Date Time :: SiteName :: Sensor Pod/Bank name  :: Sensor Point Name :: Alarm Alias
```

For example, a typical Pager notification for a temperature alarm might look like the following:

```
10/24 15:59 :: San Diego Site #12 :: Sensor Pod 12 :: Cabinet Temp :: Temperature Very High
```

EventSensor™ Configuration Setup

The T850 can be ordered with any of the following different internal I/O devices or can be connected to a number of external EventSensor devices as described in this section. The setup menus are the same regardless of whether the device is internal or external to the T850.

Input
Contact closure
Temperature
Humidity
Voltage and Current

Output
Relays

Virtual (via Scripting only)
Virtual EventSensors

```
EventSensor ID: ESI000999
Name: unnamed
Contact Closure States:
 01 unnamed                    Open
 02 unnamed                    Open
 03 unnamed                    Open
 04 unnamed                    Open
 05 unnamed                    Open
 06 unnamed                    Open
 07 unnamed                    Open
 08 unnamed                    Open
```

Above is a representative Internal Events Menu showing an ES-8C Type 2 EventSensor that features 8 contact closures. Descriptions of temperature, humidity, voltage and relays will follow.

Contact Closure n displays the menu for configuring each of the contact closure points.

Contact Closure Setup

```
TeleBoss 850 - Internal Contact Closure Event 1
A) Sensor Name                                      [unnamed]
B) Contact Closure Enabled                          [OFF]
C) Event State                                      [OPEN]
D) Threshold                                        [2]
E) Event State Actions                              []
F) Return to Normal Actions                         []
G) Event State Class                                [Info]
H) Return to Normal Class                           [Info]
I) Event Trap Number                                [110]
J) Return to Normal Trap Number                     [110]
K) Active Alarm Alias                               []
L) Inactive Alarm Alias                             []
```

Contact closures (CC) sense the state of a circuit. A weak voltage is applied to the source pin and if pulled to ground by a connection on the circuit, the sensor reports a "closed" state. If it remains high, the sensor reports an "open" state. All of the CCs share a common ground. The contact closures may be configured to alarm in either the open or closed state, depending on the needs of the attached devices.

Sensor Name is an alphanumeric field that allows you to name this contact closure. (Max length 16 chars)

Contact Closure Enabled is an ON/OFF toggle to enable this contact closure. Default setting is OFF.

Event State is an OPEN/CLOSED toggle that determines whether an event will be triggered when the contact closure circuit is opened or closed. The default state is OPEN.

Threshold is the number of seconds (0-255) the sensor must remain in the event state before an actual event occurs. Default threshold is 2.

Event State / Return to Normal Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Event State / Return to Normal Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Event / Return to Normal Trap Number sets the trap number which can be useful when using SNMP trap managers that employ a trap numbering system to help identify incoming traps. The default trap number for Contact Closure Events is 110, but any number in the alternate range of 1000 – 1199 can be used.

Active Alarm Alias is a special sensor name used when reporting active events for this sensor.

Inactive Alarm Alias is the same as Active Alarm Alias, but used with Return to Normal events.

Temperature Sensor Setup

```
TeleBoss 850 - Internal Temperature Event
A) Temperature Sensor Enabled           [OFF]
B) Sensor Values Represented in         [FAHRENHEIT]
C) Temperature Deadband                 [3]
D) Very High Event Settings             [100] []          [120] [Info]
E) High Event Settings                  [80]  []          [120] [Info]
F) Return to Normal Settings            [-]   []          [120] [Info]
G) Low Event Settings                   [50]  []          [120] [Info]
H) Very Low Event Settings              [30]  []          [120] [Info]
```

Temperature Sensor Enabled is an ON/OFF toggle to enable the temperature sensor. Default setting is OFF.

Sensor Values Represented In toggles either FAHRENHEIT or CELSIUS for the desired temperature scale.

Temperature Deadband is the range, in degrees, on either side of a temperature setting that prevents the alarm from repeatedly going in and out of the "alarm state" as the actual temperature fluctuates above and below the temperature setting.

Very High / High / Low / Very Low Event Settings display a menu where the temperature at each level can be configured to alarm along with the action(s) to occur, trap number, and class. In the case of Very High or High levels, the alarm will occur as the temperature rises above the setting. In the case of Low or Very Low, the alarm will occur as the temperature drops below the setting.

Return to Normal Settings displays a menu where the actions to occur when the temperature returns to normal (drops below the High/Very High settings, or rises above the Low/Very Low settings) can be configured.

Very High / High / Low / Very Low Event Settings Setup

```
TeleBoss 850 - Internal Temperature Event Settings
A) Very High Event Temperature                      [100]
B) Very High Event Actions                          []
C) Very High Event Trap Number                      [120]
D) Very High Event Class                            [Info]
```

The menu for setting Very High Temperature settings is shown. Menus for High/Low/Very Low are identical.

Very High Event Temperature sets the temperature at which the Very High Event Actions will be triggered.

Very High Event Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Very High Trap Number sets the trap number which can be useful when using SNMP trap managers that employ a trap numbering system to help identify incoming traps. The default trap number for Temperature Events is 120, but any number in the alternate range of 1000 – 1199 can be used.

Very High Event Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Return to Normal Settings Setup

```
TeleBoss 850 - Internal Temperature Event Settings
A) Return to Normal Event Actions                   []
B) Return to Normal Event Trap Number               [120]
C) Return to Normal Class                           [Info]
```

Return to Normal Event Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Return to Normal Event Trap Number sets the trap number which can be useful when using SNMP trap managers that employ a trap numbering system to help identify incoming traps. The default trap number for Temperature Events is 120, but any number in the alternate range of 1000 – 1199 can be used.

Return to Normal Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Humidity Sensor Setup

```
TeleBoss 850 - Internal Humidity Event
A) Humidity Sensor Enabled              [OFF]
B) Humidity Deadband                    [3]
C) Very High Event Settings             [90]  []          [130] [Info]
D) High Event Settings                  [80]  []          [130] [Info]
E) Return to Normal Settings            [-]   []          [130] [Info]
F) Low Event Settings                   [20]  []          [130] [Info]
G) Very Low Event Settings              [10]  []          [130] [Info]
```

Humidity Sensor Enabled is an ON/OFF toggle to enable the humidity sensor. Default setting is OFF.

Humidity Deadband is the range on either side of a humidity setting that prevents the alarm from repeatedly going in and out off the "alarm state" as the actual humidity fluctuates above and below the humidity setting.

Very High / High / Low / Very Low Event Settings display a menu where the humidity at each level can be configured to alarm along with the action(s) to occur, trap number, and class. In the case of Very High or High levels, the alarm will occur as the humidity rises above the setting. In the case of Low or Very Low, the alarm will occur as the humidity drops below the setting.

Return to Normal Settings displays a menu where the actions to occur when the humidity returns to normal (drops below the High/Very High settings, or rises above the Low/Very Low settings) can be configured.

Very High/High/Low/Very Low Event Settings Setup

```
TeleBoss 850 - Internal Humidity Event Settings
A) High Event Humidity                              [80]
B) High Event Actions                               []
C) High Event Trap Number                           [130]
D) High Event Class                                 [Info]
```

The menu for setting High Humidity settings is shown. Menus for Very High/Low/Very Low are identical.

High Event Humidity sets the humidity at which the High Event Actions will be triggered.

High Event Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

High Trap Number sets the trap number which can be useful when using SNMP trap managers that employ a trap numbering system to help identify incoming traps. The default trap number for Humidity Events is 130, but any number in the alternate range of 1000 – 1199 can be used.

High Event Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Return to Normal Settings Setup

```
TeleBoss 850 - Internal Humidity Event Settings
A) Return to Normal Event Actions                   []
B) Return to Normal Event Trap Number               [130]
C) Return to Normal Event Class                     [Info]
```

Return to Normal Event Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Return to Normal Event Trap Number sets the trap number which can be useful when using SNMP trap managers that employ a trap numbering system to help identify incoming traps. The default trap number for Humidity Events is 130, but any number in the alternate range of 1000 – 1199 can be used.

Return to Normal Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Analog Voltage / Current Sensor Setup

Below is a representative Events Menu showing an 8V Expanson Card to monitor 8 analog voltage inputs. Analog current inputs, such as on an 8mA Expansion Card use an identical menu. Analog sensors can be used in various applications, from monitoring a power supply to verifying RS232 voltage levels.

```
TeleBoss 850 - External Events Menu
Device Number: 2         Device ID: 20020000
A) Device Name                                      [unnamed]
B) Analog Input 1
. . .
I) Analog Input 8
J) EventSensor Reporting Enabled                    [OFF]
K) Clear Settings for This EventSensor

Enter your Selection:
```

Analog voltage sensors provide individual voltage sensing for ranges from –60/+60VDC.
Analog current sensors provide individual voltage sensing for ranges from 4-20mA.

Note: Effective with firmware version 2.05.840, the analog value has changed from 16-bit to 32-bit, which increases range of unit value to +2147483647 and –2147483648. Now all analog voltage slot cards, including 4VP, can handle +-60 volts and still maintain accuracy and resolution at lower voltage inputs.

Device Name is the option name given to the sensor. Default is unnamed.

Analog Input n displays a menu where each analog voltage sensor can be configured.

Event Sensor Reporting Enabled is an ON/OFF toggle to enable the Event Sensor Reporting feature. See the Event Sensor Reporting section in the Features chapter for more information.

Clear Settings for This EventSensor when selected will immediately clear all of the configured settings for this sensor and remove it from the Sensor Events menu (except for Internal Sensors). If "Confirmation Prompt“ in General Settings is ON, then there will be a confirmation prompt (`Are you sure (y/n)?`) displayed before clearing the configured settings. Return to the Sensor Events menu to assign it a new slot, if desired, and reconfigure it.

Analog Input *n*

```
TeleBoss 850 Internal Analog Input Event 1
Device Number: 1         Device ID: 20020000       Device Name:
A) Analog Input Enabled                [OFF]
B) Name                                [ ]
C) Input Polarity                      [POSITIVE]
D) Deadband                            [30]
E) Very High Event Settings            [600]    []        [140]  [Info]
F) High Event Settings                 [600]    []        [140]  [Info]
G) Return to Normal Settings           [-]      []        [140]  [Info]
H) Low Event Settings                  [0]      []        [140]  [Info]
I) Very Low Event Settings             [0]      []        [140]  [Info]
J) Unit Conversion Settings            [Volts]
```

Analog Input Enabled is an ON/OFF toggle to enable this analog sensor. Default setting is OFF.

Name is an alphanumeric field that allows you to name this analog input. (Max length 16 chars)

Input Polarity indicates to the unit whether the input polarity will be positive or negative.

Deadband is the range on either side of an analog setting that prevents the alarm from repeatedly going in and out off the "alarm state" as the actual voltage or current fluctuates above and below the setting.

Very High / High / Low / Very Low Event Settings displays a menu where the voltage or current at each level can be configured to alarm along with the action(s) to occur, trap number, and class. In the case of Very High or High levels, the alarm will occur as the voltage or current rises above the setting. In the case of Low or Very Low, the alarm will occur as the voltage or current drops below the setting.

Return to Normal Settings displays a menu where the optional action definition for alarms as they return to a normal state can be configured.

Unit Conversion Settings displays a menu where “real world” values can be configured.

Very High / High / Low / Very Low Analog Input Event Settings

```
TeleBoss 850 Internal Analog Input Event Settings
Device Number: 1         Device ID: 20020000       Device Name:
A) Very High Event Value                            [600]
B) Very High Event Actions                          []
C) Very High Event Trap Number                      [140]
D) Very High Event Class                            [Info]
```

The menu for setting Very High Event Value settings is shown. Menus for High/Low/Very Low are identical.

Very High Event Value sets the voltage or current (in tenths) at which the Very High Event Actions will be triggered.

Very High Event Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Very High Event Trap Number sets the trap number which can be useful when using SNMP trap managers that employ a trap numbering system to help identify incoming traps. The default trap number for Analog Events is 140, but any number in the alternate range of 1000 – 1199 can be used.

Very High Event Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Return to Normal Settings

```
TeleBoss 850 Internal Analog Input Event Settings
Device Number: 1         Device ID: 20020000       Device Name:
A) Return to Normal Event Actions                   []
B) Return to Normal Event Trap Number               [140]
C) Return to Normal Event Class                     [Info]
```

Return to Normal Event Actions displays the Actions List, a menu where the action string for the event is configured. This field will be empty [ ] if no actions have been configured, and will show [*SET*] if one or more actions have been configured. Refer to Action List in the Features chapter for more information.

Return to Normal Event Trap Number sets the trap number which can be useful when using SNMP trap managers that employ a trap numbering system to help identify incoming traps. The default trap number for analog events is 140, but any number in the alternate range of 1000 – 1199 can be used.

Return to Normal Event Class sets the class for the alarm. When this option is selected, a list of the classes previously defined in the Class Table is displayed, from which you can select one to be assigned to this event.

Unit Conversion Settings

```
TeleBoss 850 Analog Input Event Unit Conversion
Device Number: 1         Device ID: 20020000       Device Name:
A) Unit Name                                        [Volts]
B) Low Voltage Amount (tenths)                      [0]
C) Low Unit Amount (tenths)                         [0]
D) Low Unit Sign                                    [POSITIVE]
E) High Voltage Amount (tenths)                     [600]
F) High Unit Amount (tenths)                        [600]
G) High Unit Sign                                   [POSITIVE]
```

Relay Output Setup

```
TeleBoss 850 - Internal Relay Event Settings
A) Device Name                                          []
B) Relay 1                                              []
C) Relay 2                                              []
D) Relay 3                                              []
E) Relay 4                                              []
F) Relay 5                                              []
G) Relay 6                                              []
H) Relay 7                                              []
I) Relay 8                                              []
J) Clear Settings for This EventSensor
```

Internal relay outputs provide electrical output that can open or close an external circuit. Typically this is used with devices that would not otherwise be able to interface with a host product, like audio alarms, LEDs, custom circuitry, and an almost limitless number of other applications.

Device Name is the option name given to the relay module.

Relay *n* displays a menu where each relay output can be configured.

Clear Settings for This EventSensor when selected will immediately clear all of the configured settings for this relay and remove it from the Sensor Events menu (except for Internal Sensors). Return to the Sensor Events menu to assign it a new slot, if desired, and reconfigure it.

Relay *n*

```
TeleBoss 850 - Internal Relay Event 1
A) Relay Name                                           []
B) Relay Active State                                   [CLOSED]
```

Relay Name is a text-entry field that allows you to name this relay.

Relay Active State toggles CLOSED/OPEN to set whether the relay will close or open when activated. Default setting is CLOSED.

» **Note:** There are two types of optional relay Expansion Cards available for use in the T850.

- a) **8R (or xx4R)** which is an electro-mechanical relay, rated as follows:
  - a. Max switched VDC: 60V peak (AC or DC)
  - b. Max switched power: 30W (DC)
  - c. Max switched current: 0.6A
- b) **8SR (or xx4SR)** which is a solid state relay, rated as follows:
  - a. Max switched VDC: 60V peak (AC or DC)
  - b. Max switched power: 90W (DC)
  - c. Max switched current: 01.5A

With the use of solid state relays, mechanical components and switch contacts are eliminated, thus having less potential issues with arcing, terminal degradation or fused contacts.

There is no way to differentiate one type of relay from the other in the user interface so use this Setting Key **event.sensor[x].output[y].type**, which will return the type of output for a given point on the card.

For example, an **8SR** card would display like this:

```
event.sensor[1].output[1].type = SSR
event.sensor[1].output[2].type = SSR
event.sensor[1].output[3].type = SSR
```

```
   event.sensor[1].output[4].type = SSR
   event.sensor[1].output[5].type = SSR
   event.sensor[1].output[6].type = SSR
   event.sensor[1].output[7].type = SSR
   event.sensor[1].output[8].type = SSR
```

For example, an **8R** card would display like this:

```
   event.sensor[1].output[1].type = Relay
   event.sensor[1].output[2].type = Relay
   event.sensor[1].output[3].type = Relay
   event.sensor[1].output[4].type = Relay
   event.sensor[1].output[5].type = Relay
   event.sensor[1].output[6].type = Relay
   event.sensor[1].output[7].type = Relay
   event.sensor[1].output[8].type = Relay
```

Contact Asentria Technical Support if you have any questions about the type of relay card installed in your T850.

Relays as Alarm Action

Relays can be used to open or close part of a circuit of your design or part of another product. You can use the T850 internal relays to control these devices. Relays can be toggled based on sensor readings, data events, or even remotely by SNMP.

Caution: Do not exceed maximum ratings for relays. T850 relays are only designed to switch relatively low voltages and amps, and are not intended to switch AC powered devices. Only a certified electrician should work with and connect AC Voltage to the T850. Improper use outside the guidelines of this manual could cause injury or death.

8R electro-mechanical relay	**8SR solid state relay**
Max switched voltage: 60V	Max switched voltage: 60V
Max switched power: 30W	Max switched power: 90W
Max switched current: 1A	Max switched current: 1.5A

Remember Ohm's law: W = V x A (watts = volts x amps)
30W = 1A x 30V
30W = .5A x 60V

Note: Be aware of the inrush (startup) current of the device you are connecting to the relays. A device drawing 1A while powered up can draw many times that upon power up. This is especially true with capacitive or inductive circuits.

Action Definition
Relays actions are defined in the Action List and below. Relay definitions are somewhat more complicated than other sensors in that they must declare the action to perform, which sensor the relay is on, and which relay on that sensor to switch.

Relay actions are declared with the following syntax:

- relay(action, EventSensor, point)
  Put a relay in a certain state specified by *action*.
  - *action*: one of the following two words, by case-insensitive exact match or partial unambiguous match: *active* or *inactive*. “Active” always means to energize the relay.
  - *EventSensor*: the number of the EventSensor that has the specified relay, where it is the same as that referred to by the index in an EventSensor key (e.g., 1 in **`event.sensor[1].*`** for the first external EventSensor) as well as that referred to by the SNMP esIndex object.
  - *point*: the number of the relay (1-based) on the specified EventSensor. E.g., this is the same number x in "**`event.sensor[1].relay[x].*`**".

EventSensor Reporting

EventSensor Reporting (formerly known as Contact Mirroring) is the feature where a unit can transmit/receive EventSensor (ES) data to/from other units. When transmitting, you can select which physical ES's should report their data, and one IP address to report to. When receiving, you can configure the unit to monitor an ES as if it were attached to the unit with a cable, when it is actually attached to the unit only with a TCP connection. Put simply, this feature allows a device in one location to affect an action at another location even though the two devices are not physically connected.

A unit can monitor data from EventSensors on any medium that can carry a TCP connection: Ethernet, ADSL, POTS/Wireless modems, SitePath, etc.

In addition to the menu option you saw on the Sensor Events Menu, there is this menu option in the Networking Settings menu:

G) EventSensor Reporting Settings

```
TeleBoss 850 EventSensor Reporting Settings
A) EventSensor Report To IP                        [0.0.0.0]
B) EventSensor Report To Port                      [4000]
C) Enable EventSensor Reporting Host               [OFF]
D) EventSensor Reporting Host Port                 [4000]
```

Options A & B are configured on the client unit. A is where you enter the IP address of the host T850 and B is where you select a TCP port to use.

Options C & D are configured on the host T850. C enables it to receive EventSensor reports from the client unit, and D is where you select the TCP port it should be listening on.

Obviously Option B on the client unit should match Option D on the host T850.

When everything is properly configured, the sensor at the client (Site A) will appear in the Sensor Events Menu of the host (Site B), with (REMOTE) following the Alive indicator for that sensor:

Setting Keys

There are 4 global settings that control TCP transmitting/receiving:

```
net.esreporting.listen.enable
net.esreporting.listen.port
net.esreporting.connect.server
net.esreporting.connect.port
```

There is one per-ES setting that controls whether the ES reports its data:

```
event.sensor.reporting.enable
```

Type2 EventSensor™ Setup

The T850 SensorJack port supports up to 16 Type2 EventSensors. Type2 EventSensors are different than the Type1 EventSensors sold by Asentria but support similar and expanded monitoring capabilities. Type2 EventSensors work only with the SiteBoss and TeleBoss line of Asentria products. Data-Link and SNMP-Link products use only the Type1 Event Sensors. (The two Types are not compatible.) However, configuration of Type2 EventSensors within the EventSensor Device Settings menu is identical to how Type1 EventSensors are configured.

Connections

Type2 EventSensors connect to the host unit and each other via an RJ45/9-pin Mini DIN cable. The 9-pin Mini DIN cable end of the EventSensor cable plugs in to the SensorJack port on the back panel of the T850. The RJ45 end of that cable plugs in to the Type2 EventSensor RJ45 port labeled Control. Additional Type2 EventSensors are chained together using Cat-5 straight-thru cable from the Sensor port on the first EventSensor, to the Control port on the next EventSensor. Be sure to set the DIP switches for each additional EventSensor so that each occupies it's own slot as per the chart below.

Different configuration arrays of Type2 Event Sensors are fully described with graphics in the EventSensor Datasheet which is available from either Asentria Sales (sales@asentria.com) or Tech Support (support@asentria.com).

DIP Switch Settings

Defines up to 16 address locations. Note that the DIP switch is numbered from left to right, 1 through 4. The Most Significant Bit (MSB) is switch location 1.

1 = DIP Switch up 0 = DIP Switch down

DIP SW	Slot	DIP SW	Slot	DIP SW	Slot	DIP SW	Slot
0000	= 1	0100	= 5	1000	= 9	1100	= 13
0001	= 2	0101	= 6	1001	= 10	1101	= 14
0010	= 3	0110	= 7	1010	= 11	1110	= 15
0011	= 4	0111	= 8	1011	= 12	1111	= 16

Configuration

Refer to the EventSensor Configuration section for configuration instructions.

Calibration of Temperature and Humidity Sensors

Temperature and humidity settings can be calibrated in ES-T and ES-TH Type2 EventSensors via Setting Keys (no menu options available to do this). This gives a user the ability to make calibration adjustments to fine-tune the accuracy of the reported reading, if desired. This process is transparent and provides temperature and humidity readings that are consistent with other devices that measure temperature and relative humidity in the same environment. This mechanism uses two calibration points to set up a slope and offset that is used to adjust the measured reading.

This feature is enabled by changing Setting Key values only; the text menu and web interface do not provide access to these keys. The default Setting Keys are:

```
event.sensor[x].humid[y].callowin=0
event.sensor[x].humid[y].callowout=0
event.sensor[x].humid[y].calhighin=100
event.sensor[x].humid[y].calhighout=100
event.sensor[x].temp[y].callowin=0
event.sensor[x].temp[y].callowout=0
event.sensor[x].temp[y].calhighin=100
event.sensor[x].temp[y].calhighout=100
```

Example calibration procedure for humidity sensor:

1) Place the ES-TH in a controlled-humidity environment along with an accurate humidity reference.
2) Set the humidity to some level toward the low end of the range, like 10-20%, and wait for it to stabilize.
3) Write down the humidity as indicated by the reference, and the humidity as indicated by the ES-TH.
4) Repeat the previous two steps, except set the range toward the high end, like 70-90%.
5) Enter the values that were written down in the appropriate settings:

```
event.sensor[x].humid[y].callowin = <low indicated value>
event.sensor[x].humid[y].callowout = <low reference value>
event.sensor[x].humid[y].calhighin = <high indicated value>
event.sensor[x].humid[y].calhighout = <high reference value>
```

For example, if the eventsensor 1 indicated 23% RH when the reference indicated 30% R, and the eventsensor indicated 84% RH when the reference indicated 90% RH, then the following values should be entered:

```
event.sensor[1].humid[1].callowin = 23
event.sensor[1].humid[1].callowout = 30
event.sensor[1].humid[1].calhighin = 84
event.sensor[1].humid[1].calhighout = 90
```

A similar procedure is used for temperature calibration.

Contact Asentria Technical Support if you have any questions concerning this.

Virtual EventSensor Setup

Via Scripting, the T850 supports a "virtual EventSensor". This in an allocation in RAM with 48 analog sensor values. The sensors on this Virtual EventSensor can be read with the usual keys. There can be only one Virtual EventSensor on a T850. A Virtual EventSensor behaves like a 'real' or physical EventSensor with regards to events, actions, and SNMP. Unlike a physical EventSensor, a Virtual EventSensor's sensor value keys are writable.

The Virtual EventSensor allows a script to populate sensor values based on some custom functionality, like querying a thrid part sensor via serial port or network. With some scripting customization, a third party sensor can be treated as one of the T850's EventSensors, bringing it under the umbrella of consistent SNMP access and event/action management.

The script calls a_lib.InitVirtualES. This allocates the Virtual EventSensor. From there, the script can write sensor values to it (and read them). Configure events and actions for it as you normally would for any other kind of EventSensor.

Customizable Command Prompt

This feature allows the prompt in the command processor to be customized, and includes the ability to embed one or more settings values in the prompt. A customized command prompt can help simplify administration of units, particularly where multiple units are involved.

The command prompt setting is available in the General setup menu section, and via the Setting Key `sys.prompt`. The setting can contain up to 64 characters, but the prompt itself is limited to 30 characters; any additional characters are truncated.

In addition to specifying plain text to be included in the command prompt, setting values can be embedded using a special syntax: $(setting_key_name). If this construct is used, the value of the specified Setting Key replaces the construct. If the Setting Key is not accessible for any reason (invalid key, insufficient user access level, etc), "ERROR" is displayed instead.

Note: T850 only supports the `sys.sitename` Setting Key; all others return "ERROR".
To make the system prompt blank, set `sys.prompt` to a null value (i.e. "`sk sys.prompt` = ").

Examples:

```
Set prompt to be “>”
        Via Setup menu:   Enter new prompt:  >
        Via Setting Key:  sk sys.prompt = >

Set prompt to be “Site Name”
        Via Setup menu:   Enter new prompt: TeleBoss (or whatever the site name is)
        Via Setting Key:  sk sys.prompt = “$(sys.sitename) “

Set prompt to be “System Date and Time>”
        Via Setup menu:   The date and time entered via the Prompt option do not
                          change as the actual date and time progress.  What you enter
                          here will always be displayed as the prompt, until you change
                          it.  If you want the date/time prompt to change with the
                          system clock, then change it via the Setting Key entry
                          describe below.
        Via Setting Key:  sk sys.prompt = $(sys.clock.date) $(sys.clock.time)>
```

IP Record Collection (IPRC)

The T850 supports the following IP Record Collection protocols/IP-enabled switches:

Generic Server
Avaya – Reliable Session Protocol
Alcatel OmniPCX
CCM4 (Cisco CallManager version 4.x)
Generic Client

- Siemens HiPath 4000

Intecom Telari
Nortel BCM
Syslog
NEC NEAX2400
CCM5 (Cisco CallManager version 5.x)

Generic Server

Definition
Generic Server is plain text record collection that offers no handshaking or quality control above that of the TCP/IP protocols. Therefore, this method of record collection is not specific to Avaya Definity in that there is no application-layer protocol. Plain Text IPRC data is received on TCP port 5000 (user-adjustable).

Commands

Command	**Function**
IPRC IPRC STATUS IPRC ?	Displays a status report of the active IPRC mode.

Status Display
The IPRC command brings up a status report similar to the following report for Generic Server:

```
iprc
Record Collection Server
Status: Listening on port 5000
```

This report simply indicates the status of the RCS. The TCP port is displayed for informational purposes only.

Avaya – Reliable Session Protocol

Definition

Reliable Session Protocol (RSP) is Avaya's solution to the problem of lost connections while transferring valuable call record data. This protocol is used on both the client (PBX) and server (Data-Link) sides to ensure that if the data connection breaks, no records are lost. This is accomplished by the devices repeatedly checking in with one another. If the connection is lost, an alarm sent out by the Data-Link (and the PBX if so configured), and the PBX will begin to buffer its own data until the connection is restored.

Commands

Command	Function
IPRC or IPRC STATUS or IPRC ?	Displays a status report of the active IPRC mode.
IPRC PORT n	Changes the TCP port on which to listen for RSP connections.
IPRC RESET	Manually disconnects the current session (if connected), closes the socket (if established), and reinitializes the server.

Status Display

The IPRC command brings up a status report similar to the following report for RSP:

```
RSP Server
Status: Listening on port 9000
SAMs tx        : 0
ACKs tx        : 0
SDMs tx        : 0
SCMs rx        : 0
New data rx    : 0
Blocks rx      : 0
Dup. blocks rx : 0
```

IPRC Terms

The following terms are used in the status display accessed by the IPRC command for RSP:

Term	Meaning
SAM (Session Accept Message)	A message transmitted by the Data-Link to acknowledge the client's Session Connect Message.
ACK (session Acknowledgement message)	A response transmitted by the Data-Link to acknowledge data blocks.
SDM (Session Disconnect Message)	A command sent from the T850 to terminate the current session. This happens when the T850 encounters an anomaly in the protocol or the user resets the server.
SCM (Session Connect Message)	A request transmitted from the client to establish a session with the T850's IPRC server.
New Data	The number of non-duplicate bytes received by the server.
Blocks	Represents the number of blocks (including duplicates) received by the unit.
Dup. Blocks	The number of duplicate blocks received by the unit. If this number is high relative to the number of blocks received, either the SPDU Response Timer (ST2) on the switch needs to be increased or the Data-Link is full and needs to be polled.

Alcatel OmniPCX 4400

Definition

The T850 supports the Alcatel OmniPCX 4400 ticket system of IPRC. This method involves receiving large data packets or tickets via TCP. These tickets contain many different data fields that may not be useful to a system administrator. The T850 allows an administrator to use a configuration file that selects exactly which records to store in the CDR database.

Commands

Command	Function
SK SET X	Initiates a settings key file upload via Xmodem.
SK SET A	Initiates a settings key file upload via plaintext ASCII.
SK LOG	Displays results of uploading the settings key file.
IPRC START	Opens a connection to the PBX if not already open. This is required if the unit was unable to connect to the PBX at boot because of improper settings.
IPRC STOP	Places the client into an idle state. Closes any open connection.
IPRC or IPRC STATUS or IPRC ?	Displays a status report of the active IPRC mode.
IPRC FIELDS	Displays the list of compiled output fields.
IPRC DEBUG ON	Show the ticket data as it is parsed.
IPRC DEBUG OFF	Disables showing ticket data as it is parsed.

Status Display

The following is an example status display for Alcatel OmniPCX IPRC:

```
Alcatel OmniPCX 4400 Real Time Client
Status: Idle
Last Error: No fields active (0/00:00:06 ago)
Seconds until next state: 0
Tickets processed: 00000000
```

The Configuration File

The ticket parsing functionality is configured via a configuration file. This configuration file is a list of setting keys, where a setting key is a "<setting> = <value>" statement. <setting> is a period-delimited string of keywords. These keys can name all of the setup variables of the product. These include the generic operational parameters of the box such as these below, as well as specialized parameters such as those for the OmniPCX:

```
net.ip=192.168.100.32
net.subnet=255.255.255.0
net.router=192.168.100.100
net.snmp[1]=192.168.100.36
net.snmp[2]=0.0.0.0
net.snmpcomm=public
```

The unit assembles output fields into records defined by their end-of-line characters. Using this method we can specify output fields using the specific ticket field numbers (1-48) or by character start position and length within the ticket structure. For example, if the user wants to create an output record which contains these fields:

Call Type, Start Date Time, End Date Time, Effective Call Duration (converted from seconds to HH:MM:SS format), Acting Extension Number, Trunk 1 and the user wants to specify the record using TICKET FIELD NUMBERS, the setup would look like this:

```
alcatel.ip=22.23.212.12
alcatel.port = 2533
alcatel.timeout = 30
alcatel.field[1]=10,2,L // Call Type
alcatel.field[2]=40,17,L // Start Date Time
alcatel.field[3]=12,17,L // End Date Time
alcatel.field[4]=38,10,L,STOHMS // Effective Call Duration
alcatel.field[5]=41,25,R // Acting Extension Number
alcatel.field[6]=16,5,5,L // Trunk Identity
alcatel.field[7]=9,30,L // Calling Number
alcatel.field[8]=2,30,L,0D0A // Called Number
```

In the above, the field definition arguments are:

<field>=<ticket field # , length of that field to take, justification[,end of line chars][,conversion]>

The 0D0A terminator on field 8 tells the unit to store all assembled output fields up to and including that output field (in ascending order of field definition number) as 1 record. Note that the 0D0A optional value places the end-of-line characters on the last field, but you could include the EOL characters at other fields also so as to make multiple records. If the final field definition does not have any EOL characters, then the unit stores whatever it has assembled so far as 1 record, appended with either:

1. the first EOL character set found in any other field definition starting with the first field, or
2. CRLF, if no other field definitions have EOL characters.

If we wanted to use explicit character position values, the setup would look like this:

```
alcatel.ip=22.23.212.12
alcatel.port = 2533
alcatel.timeout = 30
alcatel.field[1]=166,2,2,R // Call Type
alcatel.field[2]=441,17,17,L // Start Date Time
alcatel.field[3]=169,17,17,L // End Date Time
alcatel.field[4]=430,10,10,R,STOHMS // Effective Call Duration
alcatel.field[5]=458,30,20,L // Acting Extension Number
alcatel.field[6]=211,5,5,R // Trunk Identity
alcatel.field[7]=136,30,30,L // Calling Number
alcatel.field[8]=5,30,30,L,0D0A // Called Number
```

In the above, the field definition arguments are:

<field>=<start pos, how long the field is, length of that field to take, justification[,end of line chars][,conversion]>

Once a configuration file is uploaded to the unit, the T850 indicates that it is processing the data. It returns "COMPLETE" when all settings are processed. The unit gives no other progress or status feedback to the user while it is processing the file. Instead, it logs feedback to a file that the user can view after processing is complete. If there were any problems, the unit will display an error message after processing is complete.

To view the log, enter the **SK LOG** command. This will display which settings, if any, it failed to process because of bad value, key name, or syntax. Bear in mind, this upload process does not attempt to error check the output field definitions, it only stores them. Instead, the real time client verifies these field definitions when it is started. If the client is idle (you can tell the client's state by entering the **IPRC STATUS** command), you must start the client in order to tell it to compile the settings (**IPRC START** command).

Limits of field definitions
There is room for up to 3 EOL characters for each field definition. Null is an invalid EOL character. There are 2 available conversion options, if conversion is desired, for each field definition: STOHMS and MTOHMS. STOHMS assumes the input data from the ticket is a value represented in seconds, and it will convert this value to hh:mm:ss format in the output field. MTOHMS works like STOHMS except it assumes the value to be converted is in minutes. The maximum output field length is 300 characters. The maximum record length is 520 characters.

Aside from the up to 48 output fields, there are 6 other items to configure:

Setting	Function
alcatel.ip	IP address of PBX.
alcatel.port	TCP port of PBX real time interface. Default is 2533.
alcatel.timeout	Timeout (in seconds) used for waiting for packets and connection retries. Default is 30.
alcatel.delim	Output field delimiter. This is a 1-byte value, expressed as ASCII-HEX. If it is non-zero, then this byte is appended to each unterminated output field. For example, to separate each output field with a space, assign this key the value of "20". Default is "00".
net.iprc.mode = ALCATEL OMNIPCX	Selects the client as the active IPRC service.
Net.iprc.file	Selects the database file used for record storage.

Standard Ticket Fields
There are 47 ticket fields to choose from when paring down which data you would like to keep from the incoming tickets. This section covers each of the fields, their location and size in the ticket, and the alignment of the data within the field. When specifying an output field using TICKET FIELD NUMBERS, the unit uses this standard ticket format:

Note: Ticket structure is subject to change by Alcatel. You should refer to the latest Alcatel documentation if there is any problem or question.

Field	Name	Position	Size	Alignment
1	Ticket Version	0-4	5	L
2	Called Number	5-34	30	L
3	Charged Number	35-64	30	L
4	Charged User Name	65-84	20	L
5	Charged Cost Center	85-94	10	L
6	Charged Company	95-110	16	L
7	Charged Party Node	111-115	5	R
8	Charged Party Subaddress	116-135	20	L
9	Calling Number	136-165	30	L
10	Call Type	166-167	2	R
11	Cost Type	168	1	NA
12	End Date-Time	169-185	17	NA
13	Charge Units	186-190	5	R
14	Cost Info	191-200	10	R
15	Duration	201-210	10	R
16	Trunk Identity	211-215	5	R
17	Trunk Group ID	216-220	5	R
18	Trunk Node	221-225	5	R
19	Personal/Business Call	226	1	NA
20	Access Code	227-242	16	L
21	Specific Charge Info	243-249	7	NA
22	Bearer Capability	250	1	NA
23	High Level Compatibility	251-252	2	R
24	Data Volume	253-262	10	R
25	User To User Volume	263-267	5	R
26	External Facilities	268-307	40	NA
27	Internal Facilities	308-347	40	NA
28	Call Reference	348-357	10	R

29	Segments-Rate 1	358-367	10	R
30	Segments-Rate 2	368-377	10	R
31	Segments-Rate 3	378-387	10	R
32	Com Type	388	1	NA
33	X25 In Flow Rate	389-390	2	R
34	X25 Out Flow Rate	391-392	2	R
35	Carrier	393-394	2	R
36	Initial Dialed Number	395-424	30	L
37	Waiting Duration	425-429	5	R
38	Effective Call Duration	430-439	10	R
39	Redirected Call Indicator	440	1	NA
40	Start Date-time	441-457	17	NA
41	Acting Extension Number	458-487	30	L
42	Called Number Node	488-492	5	R
43	Calling Number Node	493-497	5	R
44	Initial Dialed Number Node	498-502	5	R
45	Acting Extension Number Node	503-507	5	R
46	Transit Trunk Group ID	508-512	5	R
47	EndOfLine (0x0A)	513	1	NA

The Real Time Client

The client is idle whenever there are no configured OmniPCX settings. After you upload a configuration file for the first time, type "**IPRC START**" to start the client. Then type "**IPRC STATUS**" or "**IPRC ?**" to check the current status. It will either indicate a working status (e.g., "Established - awaiting packet") or if something is wrong (e.g., unable to connect to the OmniPCX, or a certain output field definition doesn't make sense). All output field definitions must compile correctly in order for the unit to accept the configuration and attempt to connect to the OmniPCX.

Once the client has accepted a valid configuration, it will attempt to connect to the OmniPCX whenever the unit is reset. If the user manually stops the client with the **IPRC STOP** command, then the client will remain in the idle state until either the unit is reset or the user enters the **IPRC START** command.

If a new configuration is uploaded while the client is connected to the PBX, then it will:

1. Disconnect from the PBX if the configuration is invalid.
2. Stay connected to the PBX using the new configuration, if the configuration is valid and the PBX IP address or TCP port did not change.
3. Disconnect from the PBX and reconnect using the new configuration, if the configuration is valid and the PBX IP address or TCP port changed.

CCM 4 (Cisco CallManager version 4.x)

Definition

The T850 supports the Cisco CallManager 4.x software. This method involves querying the Cisco CallManager database using SQL commands. The database contains many different fields that may not be useful to a system administrator. The T850 allows an administrator to use a configuration file that selects exactly which fields to retrieve from the Cisco CallManager database.

Commands

Command	Function
SK SET X	Initiates a settings key file upload via Xmodem.
SK SET A	Initiates a settings key file upload via plaintext ASCII.
SK LOG	Displays results of uploading the settings key file.
IPRC START	Causes immediate connection to CallManager to retrieve any new records, followed by automatic connection at the interval specified by the connection interval setting. When the T850 starts up in Cisco CallManager IPRC mode, and a non-zero connection interval is set, automatic connection is enabled. This command is only required if automatic connection was previously stopped using the IPRC STOP command, or the connection interval was changed from zero to a non-zero value.
IPRC STOP	Disables automatic connection to CallManager, and terminates any open connection. Automatic connection is re-enabled if the T850 is restarted.
IPRC NOW [value]	Causes the T850 to connect to CallManager immediately and retrieve any available new records. If value is specified, it will only retrieve that many records even if more are available.
IPRC DBINFO	Causes the T850 to connect to CallManager immediately and retrieve and display the total number of records present, and the date/time stamp of the first and last records.
IPRC or IPRC STATUS or IPRC ?	Displays a status report of the active IPRC mode.
IPRC FIELDS	Displays the list of compiled output fields.
IPRC LOG	Shows any messages returned by the CallManager server during the last non-interactive connection attempt. This information can be useful for troubleshooting.
IPRC INTERACTIVE	Causes the T850 to connect to CallManager, and then present an interface for entering SQL commands to be sent to CallManager. The results of any SQL commands are displayed on-screen, and are not stored in the T850 database. Field settings do not apply in interactive mode.

Status Display

The following is an example status display for Cisco CallManager IPRC:

```
Cisco CallManager IPRC Status
State: Waiting
Last result: Retrieved 5 records (connected 0/00:00:23 ago for 00:00:07)
Time until next connection: 00:09:36
Records processed: 00000730
```

Configuration File

The record retrieving functionality is configured via a configuration file. This configuration file is a list of setting keys, where a setting key is a "<setting> = <value>" statement. <setting> is a period-delimited string of keywords. These keys can name all of the setup variables of the product. These include the generic operational parameters of the box such as these below, as well as specialized parameters such as those for the Cisco CallManager:

```
net.ip=192.168.100.32
net.subnet=255.255.255.0
net.router=192.168.100.100
net.snmp[1]=192.168.100.36
net.snmp[2]=0.0.0.0
net.snmpcomm=public
```

The unit queries the CallManager database and, for each available record, retrieves the values (columns) specified in the field table. The retrieved values are assembled into records as defined in the field table. Using this method we can specify output fields using the specific database column numbers (shown in the tables below), or by specifying the exact name of the database column.

Values can be retrieved from two CallManager tables: CallDetailRecord (CDR), and CallDetailRecordDiagnostic (CMR). When CMR values are specified, values are retrieved only from CMR records that are related to CDR records included in the query. When specifying fields, each field name/number is prefixed by "cdr." or "cmr." depending on which table the field is coming from.

For example, if the user wants to create an output record which contains these fields: cdr.dateTimeDisconnect, cdr.originalCalledPartyNumber, cdr.finalCalledPartyNumber, cdr.dateTimeOrigination (converted to MM/DD/YYYY HH:MM:SS format), cdr.origIPAddr (converted to 4-dot notation), cdr.duration, cmr.jitter, and cmr.latency, and the user wants to specify the fields using COLUMN NUMBERS, the field setup would look like this:

```
iprc.field[1]=cdr.38,10,R // Date/time disconnect (integer format)
iprc.field[2]=cdr.26,25,R // Original called party number
iprc.field[3]=cdr.28,25,R // Final called party number
iprc.field[4]=cdr.5,17,R,NTOD // Date/time origination (date/time format)
iprc.field[5]=cdr.10,15,R,NTOIP // Orig IP address (4-dot notation)
iprc.field[6]=cdr.39,10,R // Duration
iprc.field[7]=cmr.13,10,R // Jitter
iprc.field[8]=cmr.14,10,R,0D0A // Latency
```

In the above, the field definition arguments are:

```
<field>=<column# , length of that value to take, justification[,end of line chars][,conversion]>
```

If the specified length is greater than the length of the returned value, then the returned value is padded with spaces and justified within the output field based on the justification specification. 'L' means the value is left-aligned, 'R' means the value is right-aligned, and 'N' means the output field retains the size of the returned value and is not padded with spaces.

The 0D0A terminator on field 8 tells the unit to append CRLF to the end of that field. Note that in this example the 0D0A optional value places the end-of-line characters on the last field, but you could include the EOL characters at other fields also so as to break the record into multiple lines. If the final field definition does not have any EOL characters specified, then the unit appends CRLF automatically.

If we wanted to use explicit column names (if, for example, a column is desired that is not in the COLUMN NUMBER table), the setup would look like this:

```
iprc.field[1]= cdr.dateTimeDisconnect,10,R // Date/time disconnect (integer
format)
iprc.field[2]= cdr.originalCalledPartyNumber,25,R // Original called party
number
iprc.field[3]= cdr.finalCalledPartyNumber,25,R // Final called party number
iprc.field[4]= cdr.dateTimeOrigination,17,R,NTOD // Date/time origination
(date/time format)
iprc.field[5]= cdr.origIPAddr,15,R // Orig IP address (4-dot notation)
iprc.field[6]= cdr.duration,10,R // Duration
iprc.field[7]= cmr.jitter,10,R // Jitter
iprc.field[8]= cmr.latency,10,R,0D0A // Latency
```

In the above, the field definition arguments are:

```
<field>=<column name, length of that field to take, justification[,end of line
chars][,conversion]>
```

Once a configuration file is uploaded to the unit, the T850 indicates that it is processing the data. It returns "COMPLETE" when all settings are processed. The unit gives no other progress or status feedback to the user while it is processing the file. Instead, it logs feedback to a file that the user can view after processing is complete. If there were any problems, the unit will display an error message after processing is complete.

To view the log, enter the **SK LOG** command. This will display which settings, if any, it failed to process because of bad value, key name, or syntax. This upload process does not attempt to error check the output field definitions, it only stores them. Instead, the fields are verified when a connection attempt is made to the CallManager server.

Limits of field definitions
There is room for up to 3 EOL characters for each field definition. Null is an invalid EOL character. There are 2 available conversion options, if conversion is desired, for each field definiton: NTOD and NTOIP. NTOD assumes the value is a coordinated universal time (UTC) value that represents the number of seconds since midnight (00:00:00) Jan. 1, 1970, and it will convert this value to "mo/dd/year hh:mm:ss" format in the output field. NTOIP assumes the value is a 32-bit representation of an IP address with the bytes reversed, so that the high-order byte contains the low-order IP address octet, and so on; the value is converted to a standard 4-dot IP address representation. The maximum output field length is 160 characters. The maximum total record length is 800 characters.

Aside from the up to 48 output fields, there are some other items to configure:

Setting	**Function**
iprc.mode	Selects the client as the active IPRC service.
iprc.file	Selects the database file used for record storage.
iprc.ccm.database	The name of the CallManager database containing call detail records.
iprc.ccm.username	The username for logging into the CallManager server.
iprc.ccm.password	The password for logging into the CallManager server.
iprc.ccm.interval	Determines how often the T850 connects to the CallManager server to retrieve new records, in minutes. Setting this value to 0 effectively disables automatic connection.
iprc.ccm.delimiter	Output field delimiter. This is a 1-byte value, expressed as ASCII-HEX. If it is non-zero, then this byte is appended to each unterminated output field. For example, to separate each output field with a space, assign this key the value of "20". Default is "00".
iprc.ccm.startdate	The date and time, in “MM/DD/YYYY HH:MM:SS” format, that determines which records in the CCM database are considered new records. By default, when CCM IPRC is enabled for the first time, the T850 retrieves records that are time stamped on or after midnight the day before, according to the T850 system clock. After each non-interactive connection to the CCM server, this setting is updated to reflect the last “new record” date/time.

There are 67 columns to choose from in the CallManager database – 50 in the CDR table, and 17 in the CMR table. When specifying an output field using COLUMN NUMBERS, the unit uses these standard CallManager columns:

Note: The CallManager database structure is subject to change by Cisco. You should refer to the latest Cisco documentation if there is any problem or question.

CallDetailRecord Fields

Field	Name	Max Length*	Data Type
1	cdrRecordType	10	Number
2	globalCallID_callId	10	Number
3	globalCallID_callManagerId	10	Number
4	origLegCallIdentifier	10	Number
5	dateTimeOrigination	10/19	Number
6	origNodeId	10	Number
7	origSpan	10	Number
8	callingPartyNumber	25	Text
9	origIpPort	10	Number
10	origIpAddr	10/15	Number
11	originalCallingPartyNumberPartition	50	Text
12	origCause_location	10	Number
13	origCause_value	10	Number
14	origMediaTransportAddress_IP	10/15	Number
15	origMediaTransportAddress_Port	10	Number
16	origMediaCap_payloadCapability	10	Number
17	origMediaCap_maxFramesPerPacket	10	Number
18	origMediaCap_g723BitRate	10	Number
19	lastRedirectDn	25	Text
20	lastRedirectDnPartition	50	Text
21	destLegIdentifier	10	Number
22	destNodeId	10	Number
23	destSpan	10	Number
24	destIpAddr	10/15	Number
25	destIpPort	10	Number
26	originalCalledPartyNumber	25	Text
27	originalCalledPartyNumberPartition	50	Text
28	finalCalledPartyNumber	25	Text
29	finalCalledPartyNumberPartition	50	Text
30	destCause_location	10	Number
31	destCause_value	10	Number
32	destMediaTransportAddress_IP	10/15	Number
33	destMediaTransportAddress_Port	10	Number
34	destMediaCap_payloadCapability	10	Number
35	destMediaCap_maxFramesPerPacket	10	Number
36	destMediaCap_g723BitRate	10	Number
37	dateTimeConnect	10/19	Number
38	dateTimeDisconnect	10/19	Number
39	duration	10	Number
40	origDeviceName	129	Text
41	destDeviceName	129	Text
42	origCallTerminationOnBehalfOf	10	Number
43	destCallTerminationOnBehalfOf	10	Number
44	origCalledPartyRedirectOnBehalfOf	10	Number
45	lastRedirectRedirectOnBehalfOf	10	Number
46	origCalledPartyRedirectReason	10	Number
47	lastRedirectRedirectReason	10	Number
48	joinOnBehalfOf	10	Number
49	destConversationId	10	Number
50	globalCallId_ClusterID	50	Text

Note: Max Length specifies the number of characters to represent the maximum possible value. Where two numbers are supplied, the second number specifies the number of characters after performing the usual conversion on that particular type of value.

CallDetailRecordDiagnostic Fields

Field	Name	Max Length*	Data Type
1	cdrRecordType	10	Number
2	globalCallID_ callManagerId	10	Number
3	globalCallID_callId	10	Number
4	nodeId	10	Number
5	directoryNum	50	Text
6	callIdentifier	10	Number
7	dateTimeStamp	10/19	Number
8	numberPacketsSent	10	Number
9	numberOctetsSent	10	Number
10	numberPacketsReceived	10	Number
11	numberOctetsReceived	10	Number
12	numberPacketsLost	10	Number
13	jitter	10	Number
14	latency	10	Number
15	directoryNumPartition	50	Text
16	globalCallId_ClusterID	50	Text
17	deviceName	129	Text

CallManager Operation

After the T850 is reset, or Cisco CallManager IPRC mode is selected, the unit attempts to connect to the CallManager server using the settings provided. Once successfully connected, the unit will retrieve any new records and store them into the specified T850 database file, and then disconnect from the CallManager. This operation is repeated at the interval specified in the settings, regardless of whether the previous connection attempt was successful. If a record retrieval session is in progress when the interval expires (that is, either automatic or via **IPRC NOW** command), the interval timer is reset and the next connection is deferred until the next interval expires.

The IPRC status command (**IPRC**, **IPRC STATUS**, or **IPRC ?**) provides information about the current state, as well as the result of the last connection attempt. Additional information may be available via the **IPRC LOG** command.

When a connection is made to the CallManager server, the settings in effect at the beginning of that session are used; IPRC settings changes that are made during the session are ignored.

Generic Client

Definition

Generic Client IPRC is a TCP/IP client that runs on the T850 and attempts connections to a specified host to download records. This connection is a clear text telnet protocol, typically over port 1752.

Commands

Command	Function
IPRC IPRC STATUS IPRC ?	Displays a status report of the active IPRC mode.

Status Display

The IPRC command brings up a status report similar to the following report:

```
Record Collection Client
Status: Waiting to open connection
Last error: None
```

Siemens HiPath 4000

The Siemens HiPath 4000 uses the Generic Client protocol in the T850. Setup is as described below:

```
TeleBoss 850 - IP Record Collection (IPRC) Setup
A) IP Record Collection                  [GENERIC CLIENT]  <<<<<<
B) Store Collected Data In               [FILE1]
C) Data Alarm/Filter Enable              [OFF]
D) Target Name                           [IPRC 1]
E) Hostname/IP Address                   [192.0.2.3]  <<<<<<
F) Port                                  [1201]       <<<<<<
G) Time Stamping                         [OFF]
H) Multiline Record Enable               [OFF]
```

The HiPath sends CDR via Plain text Telnet. Use **Generic Client** in the T850 to connect to the PBX "Atlantic" Port - an Ethernet port that is dedicated for CDR only. It is always set to 192.0.2.3.

Port is 1201 by default.

The T850 ETH1 IP Address MUST be set to 192.0.2.x.

Use of a Default Router is also very difficult w/ this IP setup; it is best to leave it blank.

To setup a T850 for the HiPath, one merely needs to configure the T850 Ethernet IP address as directed by the Siemens Tech, and configure IP Record Collection for Generic Server as shown above.

Polling via network (FTP push, FTP “get”, Real Time Sockets) can be accomplished using the 2nd Ethernet Port on the T850.

Intecom Telari

Definition
Intecom Telari is IPRC from EADS (f.k.a. Intecom) E and Telari switches. In this method of IP record collection, a TCP/IP client on the unit attempts connections and accepts CDR via the connection. This method of IPRC differs from Generic Client in that it employs a proprietary application-layer protocol to transmit records.

Configuration:

```
TeleBoss 850 - IP Record Collection (IPRC) Setup
A) IP Record Collection                [INTECOM TELARI]
B) Store Collected Data In             [FILE1]
C) Data Alarm/Filter Enable            [OFF]
D) Target Name                         [IPRC 1]
E) Hostname/IP Address                 []
F) Port                                [8186]
G) Connection Interval (minutes)       [1]
H) Time Stamping                       [OFF]
```

IP Record Collection sets the protocol to be used to Intecom Telari.

Store Collected Data In toggles the FILE to which all incoming Syslog data will be stored. Options are FILE1, FILE2, AUX1, AUX2, and AUX3. Default setting is FILE1.

Data Alarm/Filter Enable is an ON/OFF toggle to set whether configured Data Alarms or Filters will be applied to the incoming data. Default setting is OFF.

Target Name is the name used to identify the switch when an IPRC Connection Lost Alarm is sent via an AsentriaAlarm. Default setting is IPRC 1.

Hostname/IP Address sets the hostname or IP Address of the Telari Record Collection Server (RCS).

Port set the TCP port used by the Telari RCS. Default setting is port 8186.

Connection Interval (minutes) sets the number of minutes (1 – 65535) to wait before disconnecting an idle connection. Default setting is 1.

Time Stamping is an ON/OFF toggle to set whether each individual call record is stamped with the Date and Time received in the T850. Default setting is OFF.

Commands

Command	**Function**
IPRC IPRC STATUS IPRC ?	Displays a status report of the active IPRC mode.
IPRC Connect	Forces the client to connect from a state where it's waiting to connect.
IPRC Start	Causes immediate connection to the server to retrieve any new records and to resume regular checking. This command is only required if automatic connection was previously stopped using the IPRC STOP command.
IPRC Stop	Disables automatic connection and terminates any open connection. Automatic connection is re-enabled if the T850 is restarted.

Status Display
The IPRC command brings up a status report similar to the following report:

```
Intecom CDR Client
Status:       Idle
Time now:     12/16 12:24:10
COMPLETE
```

Nortel BCM

The Nortel Business Communications Manager (BCM) sends call records to the T850 using FTP. Therefore, the T850 must be configured to allow an incoming FTP connection from the BCM, including logging in with a user name and password. To do this, there are three things to configure – two on the T850 and one on the BCM.

On the T850:

1) Configure IPRC for Nortel BCM as shown:

```
TeleBoss 850 - IP Record Collection (IPRC) Setup
A) IP Record Collection                 [NORTEL BCM]
B) Store Collected Data In              [FILE1]
```

2) Configure any unused user with User Name: **bcm**, Password: **bcm**, Allow User Connection via **FTP**, and Upon Login the Go To **COMMAND**. The remaining menu options do not matter.

```
TeleBoss 850 - User Setup Menu
A) Enable This User Access              [ON]
B) User Name                            [bcm]
C) Password                             [********]
D) User Profile Expiration Date/Time    []
E) Allow User Connection via            [F]
F) Upon Login then Go To                [COMMAND]
G) Set Access/Pass-through Pointer To   [FILE1]
H) Pass-through Permissions
I) After PT, ESC Takes User To          [MENU]
J) PPP Connection                       [LOCAL]
K) Setup/Status Rights                  [MASTER]
L) File Release Permissions
M) File Delete Permissions
N) Additional Authentication Options
```

On the BCM:

The user should consult with the Nortel BCM technical personnel for exactly how to configure the BCM, but here is a brief outline of the Data File Transfer parameters that must be configured:

Transfer Type: (your preference)

- Push – Daily
- Push – Weekly
- Push – Monthly
- None

IP Address: ***<the IP address of the T850>***

Remote User: **bcm**

Remote Password: **bcm**

Compress File Before Transfer: **NO**

Other settings on the BCM are your preference and Asentria cannot give advice as to how any of those should be set.

Syslog

The Syslog IP Record Collection protocol allows the T850 to receive syslog messages from any Cisco voice-enabled router, including **Cisco CallManager Express**.

```
TeleBoss 850 - IP Record Collection (IPRC) Setup
A) IP Record Collection                    [SYSLOG]
B) Store Collected Data In                 [FILE1]
C) Data Alarm/Filter Enable                [OFF]
D) Target Name                             [IPRC 1]
E) TCP Port                                [1468]
F) UDP Port                                [514]
G) Time Stamping                           [OFF]
H) Multiline Record Enable                 [OFF]
I) Division Target 1                       []
J) Division Target 2                       []
```

Syslog IP Record Collection protocol is based on the BSD Syslog protocol. Messages are typically a single line of text, however, they are occasionally longer than one line of text (> 506 bytes) so the T850 features an option to break the oversize record into multiple lines, and assemble the component single lines into one multiline record. The impact of this is that the user has to take this into account when defining data alarms. To make it more predictable to the user where the unit divides an oversize message, there are additional settings called ***division targets*** (strings up to 8 characters). If the unit needs to divide an oversize message, it tries to make it so that the division target is the beginning of the remainder piece. The BSD syslog protocol specifies that a message can be 1024 bytes. So the worst case is that the unit must store a 1024-byte single-line record. The minimum number of divisions necessary to break a 1024-byte message into records of acceptable size is 2. Therefore there are 2 division target settings. If the division targets fail to work through misconfiguration then the unit divides the message such that the 1st, 507th, and 1013th bytes are the first bytes of each of the new records.

IP Record Collection sets the protocol to be used to Syslog.

Store Collected Data In toggles the FILE to which all incoming Syslog data will be stored. Options are FILE1, FILE2, AUX1, AUX2, and AUX3. Default setting is FILE1.

Data Alarm/Filter Enable is an ON/OFF toggle to set whether configured Data Alarms or Filters will be applied to the incoming data. Default setting is OFF.

Target Name is the name used to identify the switch when an IPRC Connection Lost Alarm is sent via an AsentriaAlarm. Default setting is IPRC 1.

TCP Port sets the TCP port used by the sending Cisco device. Default setting is port 1468.

UDP Port sets the UDP port used by the sending Cisco device. Default setting is port 514.

Time Stamping is an ON/OFF toggle to set whether each individual call record is stamped with the Date and Time received in the T850. Default setting is OFF.

Multiline Record Enable displays the Multiline Record Settings menu.

Division Target 1 / 2 are eight charaters text strings used to designate the beginning of a section of a divided oversize record. Default settings are blank.

Multiline Record Settings

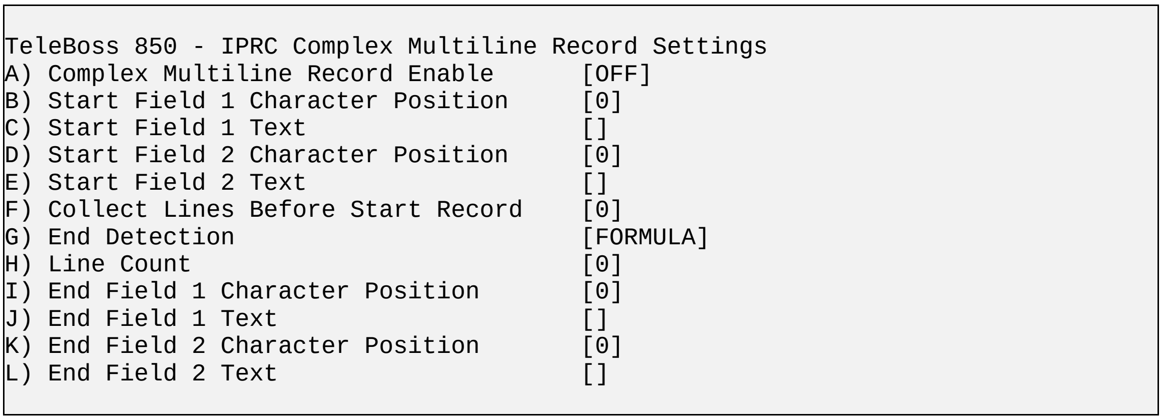
```
TeleBoss 850 - IPRC Multiline Record Settings
A) Multiline Record Enable               [OFF]
B) Blank Line Count                      [0]
C) Complex Multiline Detection           [OFF]
```

The T850 has the ability to monitor incoming Syslog CDR for multi-line records (individual records that are broken into multiple lines with carriage returns). If the records are separated by a specific number of blank lines, this basic configuration menu will suffice. If a more complex delineation scheme is used, enable Complex Multiline Detection.

Multiline Record Enable is an ON/OFF toggle to enable multiline record detection. Default setting is OFF.

Blank Line Count sets the number of blank lines that must come between records. Default setting is 0.

Complex Multiline Detection displays settings for detecting more complex multiline records. Default setting is OFF.



```
TeleBoss 850 - IPRC Complex Multiline Record Settings
A) Complex Multiline Record Enable      [OFF]
B) Start Field 1 Character Position     [0]
C) Start Field 1 Text                   []
D) Start Field 2 Character Position     [0]
E) Start Field 2 Text                   []
F) Collect Lines Before Start Record    [0]
G) End Detection                        [FORMULA]
H) Line Count                           [0]
I) End Field 1 Character Position       [0]
J) End Field 1 Text                     []
K) End Field 2 Character Position       [0]
L) End Field 2 Text                     []
```

Complex Multiline Record Enable is an ON/OFF toggle to enable advanced multiline detection. Default setting is OFF.

Start Field *n* Character Position sets the character position used to define the beginning of the multiline field. This option is used with "Count" method record end detection.

Start Field *n* Text sets the text used to determine the beginning of the multiline field. This option is used with "Formula" method record end detection.

Collect Lines Before Start Record sets the number of blank lines that are between each record.

End Detection toggles between FORMULA, COUNT, and BLANKS to set the method of detecting the end of each record. Default setting is FORMULA.

Line Count is the number of lines to meter each record at. This option is used with "BLANKS" record end detection.

End Field *n* Text/Character Position is the counterpart to start the text or character position option. This option sets the end delimiter for multiline records.

NEC NEAX2400

The T850 collects data from the NEC NEAX2400 by opening a socket on a specific port. Generally, only the Hostname or IP Address of the switch is all that needs to be configured on the T850. Two other settings on the T850 that have the same default values as the corresponding settings in the switch: Port and Device Number. In certain cases where the switch is not configured to default port and device number, you may have to adjust these either on the switch or on the T850 to get IPRC running. The Device Number ranges from 0 to 3 (default 0 on the unit) and controls what kind of data the unit retrieves from the switch; refer to the NEAX2400 SMDR reference manual for details.

```
TeleBoss 850 - IP Record Collection (IPRC) Setup
A) IP Record Collection                 [NEC NEAX2400]
B) Store Collected Data In              [FILE1]
C) Data Alarm/Filter Enable             [OFF]
D) Target Name                          [IPRC 1]
E) Hostname/IP Address                  []
F) Port                                 [60010]
G) Request Period (seconds)             [5]
H) Device Number                        [0]
I) Time Stamping                        [OFF]
```

CCM 5 (Cisco CallManager version 5.x)

Cisco CallManager version 5.x sends call records to the T850 using FTP. Therefore, the T850 must be configured to allow an incoming FTP connection from the CCM, including logging in with a user name and password. To do this, there are three things to configure – two on the T850 and one on the CCM.

On the T850:

1) Configure IPRC for CCM 5 as shown:

```
TeleBoss 850 - IP Record Collection (IPRC) Setup
A) IP Record Collection                [CCM 5]
B) Store Collected Data In             [FILE1]
```

2) Configure any unused user with User Name: **ccm**, Password: **ccm**, Allow User Connection via **FTP**, and Upon Login the Go To **COMMAND**. The remaining menu options do not matter.

```
TeleBoss 850 - User Setup Menu
A) Enable This User Access             [ON]
B) User Name                           [ccm]
C) Password                            [********]
D) User Profile Expiration Date/Time   []
E) Allow User Connection via           [F]
F) Upon Login then Go To               [COMMAND]
G) Set Access/Pass-through Pointer To  [FILE1]
H) Pass-through Permissions
I) After PT, ESC Takes User To         [MENU]
J) PPP Connection                      [LOCAL]
K) Setup/Status Rights                 [MASTER]
L) File Release Permissions
M) File Delete Permissions
N) Additional Authentication Options
```

On the CCM:

The user should consult with the Cisco technical personnel for exactly how to configure the CCM, but here is a brief outline of the Data File Transfer parameters that must be configured:

Transfer Type: (your preference)

- Push – Daily
- Push – Weekly
- Push – Monthly
- None

IP Address: ***<the IP address of the T850>***

Remote User: **ccm**

Remote Password: **ccm**

Compress File Before Transfer: **NO**

Other settings on the CCM are your preference and Asentria cannot give advice as to how any of those should be set.

Scripting

Scripting provides the ability to easily customize the operation of an Asentria device. Scripts are written in the Lua scripting language, with access to Asentria-specific functionality via a rich set of library functions. Scripts can read or change any setting on the unit, and can also create custom settings that can be accessed via Setting Keys. Scripting capabilities open up all sorts of possibilities that would previously require custom factory programming.

This chapter covers the configuration and management of scripts in the Asentria T850, and assumes a level of scripting knowledge that may not be applicable to all T850 users who wish to use scripting. For a basic primer in scripting in the T850, titled “Scripting 101”, please contact Asentria Tech Support to have this document emailed to you.

Configuration

General - the steps for using a script are:

- Write the script code in a text editor.
- Transfer the script to the unit.
- Configure the script (can be done any time before running the script).
- Invoke the script (if not scheduled to start automatically).
- The details of these steps are given in later sections.

Requirements and Limitations

Scripts can be created in any text editor as long as they are saved in pure text format. Both DOS and Unix end-of lines are supported.

A maximum of 20 scripts can be used on the unit. This includes both scripts that are loaded and scripts that are running.

Running Scripts

Before a script can be run, it must be transferred to the unit and then configured. Simply putting the script file on the unit will not allow it to be run.

Getting the Script Onto the Unit

The **SCRIPT GET** command can be used to transfer a script to the unit via XMODEM, YMODEM, ZMODEM, or TFTP or via an SK set operation. There is also a Setup menu item for this. A simple script could be created directly on the unit using the **SCRIPT EDIT** command or the equivalent Setup menu item.

Script Configuration

A script must be configured to tell the unit when the script should run, and provide any parameters required by the script. Scripts are configured via the settings described in a following section.

Script Management

In addition to scripting settings, scripts are managed via a group of commands that are available in any command processor. Here is a list of the commands with a brief description:

SCRIPT [HELP]	Display list of script commands.
SCRIPT LIST	Display a list of configured scripts.
SCRIPT START	<script> [<args>...] Start a script.
SCRIPT STATUS	<script> Display detailed status of a script.
SCRIPT STOP	<script> Stop a running script.
SCRIPT RECORDS [CLEAR]	Show/clear pending script records.
SCRIPT DEVICES	Show script device allocations.
SCRIPT GET/PUT	<file> [<args>...] Transfer script file to/from the unit.
SCRIPT DELETE	<file> Delete a script file.
SCRIPT EDIT	<file> Edit a script file (using VI editor).
SCRIPT DIR	List script file directory.
SCRIPT SHOW	<file> Display script file.
SCRIPT TEST	<script> Enter interactive script interpreter.

The following sections describe the above commands in detail.

HELP - Display Help Information

Displays the command information shown above. The command list is also displayed if an invalid command is entered.
Usage
SCRIPT or **SCRIPT HELP**

LIST - Display List of Configured Scripts

Scripts that are present on the box but not configured are not shown in this list.
Usage
SCRIPT LIST
Displays a list of configured scripts and their current status like this:

```
Scripts Status - All Scripts
Name      State       Schedule                     Arguments
========  ==========  ===========================  =================
Script 1  Not loaded  Always                       argument 1
Script 2  Not loaded  On startup, at 14:30         argument 2
```

START - Start a Script Manually

Starts a script using the specified arguments. If no arguments are specified, then the arguments associated with the script via the configuration settings are used. Attempting to start a script that is already running produces an error message, with no ill effects.
Usage
SCRIPT START <scriptname> [<arguments>...]

STATUS - Display Detailed Status of a Script

Displays detailed status of a script, including its current state, schedule, configured arguments, and information about the last time the script ran. The display looks like this:

```
TELEBOSS - Script 1 Status

Current state: Stopped
Name: Hello World
File Name: hello.lua
Schedule: Manual
Arguments: Argument 1
Open devices:
Last run time:  10/24/07  09:38:01
Last stop time:  10/24/07  09:38:32
Last exit code:  0
```

Usage
SCRIPT STATUS <scriptname>

STOP - Stop a Running Script

Causes a running script to stop. The script may not stop right away, depending on how often it checks its messages, and how long it takes to perform any shutdown tasks. If a script is in some loop where it doesn't handle the shutdown request or is otherwise crashed, it will not stop for about 20 seconds until the system shuts it down.
Usage
SCRIPT STOP <scriptname>

RECORDS - Display/Clear Pending Script Records

Displays the number of records in the special DB files dedicated to scripts, AUX1, AUX2, and AUX3. This number is the aggregate of the records in all three files. The CLEAR option deletes the records in all three files at once.
Usage

SCRIPT RECORDS [CLEAR]

DEVICES - Display Device Allocations

Displays a list of IO devices that are currently allocated to scripts. For example, if a script reserves IO1 for i/o activity, it will appear in this list.

Usage

SCRIPT DEVICES

GET/PUT - Transfer Script File

Transfers a script file to or from the unit. Type the command without any arguments for usage information.

Usage

```
SCRIPT action method <script file name> [host]
  Available actions: GET, PUT
  Available methods: X[MODEM], Y[MODEM], Z[MODEM], T[FTP]
  Script file name is required (including .lua extension)
  Host is required for tftp

  Script file name is case-sensitive, other items are not.
```

DELETE - Delete a Script File

Deletes a script file, which must be specified with the .lua extension.

Usage

SCRIPT DELETE <script file name>

EDIT - Edit a Script File

Opens a script file for editing in the VI text editor. If the script file does not exist, a new one is created. The script file name specified must have the .lua extension.

Usage

SCRIPT EDIT <script file name>

DIR - Display Script File Directory

Displays a list of the script files on the unit.

Usage

SCRIPT DIR

SHOW - Display Script File

Displays the contents of a script file. The script file name must be specified with the .lua extension.

Usage

SCRIPT SHOW <script file name>

TEST - Test a Script in Interactive Mode

Opens the script in the interactive OmniLua interpreter.

Usage

SCRIPT TEST <script name>

Uploading/Downloading/Deleting Scripts using the Setting Keys function

Scripts can be uploaded to and downloaded from the T850 using Setting Key commands. This enables settings and scripts to be configured in one operation. MASTER-level security requirement is enforced when transferring scripts with an SK file.

Uploading - Scripts can be uploaded onto the unit by doing an ‘SK set’ operation. These can be in a text file with just scripts, or in a text file with both Setting Keys and scripts.

Downloading - Scripts can be downloaded from the unit by doing any of the ‘SK get’ operations decribed here:

sk get script of **sk r**	Dumps all scripts
sk get or **sk g**	Dumps all Setting Keys followed by scripts. Setting Keys are wrapped with <keys>…</keys> XML-like header and footer text.

Deleting - Scripts can be deleted by inserting the special tag <deleteAllScripts> on a line by itself in the SK file.

Script Settings

There is a group of standard settings that control various aspects of scripting. These settings are available in the Setup menu as shown here:

```
TeleBoss 850 - Main Setup Menu
A) Network Settings
   ...
J) Scripting Settings

Enter your Selection: j

TeleBoss 850 - Scripting Settings
A) Enable Scripting                      [OFF]
B) Clear Pending Records                 [0]
C) DTR Override Ports
D) List Allocated Devices
E) List Scripts
F) Manage Script Files
```

The individual settings are detailed in the following sections.

Enable Scripting is on ON/OFF toggle that controls whether scripts are allowed to run on the unit at all. If scripting is disabled, then scripts cannot be started either automatically or manually, and other scripting functionality such as record collection and DTR override will not happen regardless of the related settings. If scripting is disabled while scripts are running, they will be issued the **STOP** command which could take up to 20 seconds to complete. If re-enabled, scripting will not function until after the previous scripting session is completely shut down (i.e. all scripts are stopped).

Clear Pending Records displays the number of script records pending, and when selected will clear them, setting the counter back to 0.

DTR Override Ports displays a menu that toggles ON/OFF to specify IO ports where DTR handling will be under script control. Normally the state of the DTR output pin on the IO ports is kept high. On these ports, after a power-cycle or reset, DTR will stay low until a script changes it to the high state.

List Allocated Devices displays a list of I/O devices that are currently allocated to a running script.

List Scripts displays the menu that lists of all of the 20 script entries, including the name, current state, and configured arguments. Selecting a script opens up a submenu with detailed settings and status for that script.

Manage Script Files displays the menu that allows the user to manage script files.

Script List

```
TeleBoss 850 - Script 1 Settings
A) Enable                                [OFF]
B) Name                                  []
C) File Name                             []
D) Run Always                            [OFF]
E) Run At Startup                        [OFF]
F) Run At Scheduled Time                 [OFF]
G) Repeat Interval (minutes)             [0]
H) Arguments                             []
I) Start Script Now
J) Stop Script Now
K) Detailed Status                       [Disabled]
```

Enable is an ON/OFF togle that enables/disables the script. If disabled, the script will not run on schedule, and cannot be run manually. Default setting is OFF.

Name sets the name of script. This is the name that is used when referring to the script, and should not be confused with the name of the script file associated with the script.

File Name sets the name of the script file associated with this script. The same script file can be used with any number of scripts.

Run Always is an ON/OFF toggle where if enabled, the script starts after the unit starts up, and is restarted automatically if it stops for any reason. Default setting is OFF.

Run At Startup is an ON/OFF toggle where if enabled, the script starts after the unit starts up. If it stops for any reason, it is not restarted unless the unit itself is restarted. Default setting is OFF.

Run At Scheduled Time is an ON/OFF toggle where if enabled, the script is run at the specified time each day. Default setting is OFF.

Repeat Interval sets the time in minutes of how often the script is repeated. If a non-zero value is entered, the script is run at the specified interval, measured from the last time the script was started on a schedule. Default setting is 0.

Arguments sets the specified arguments that are passed to the script when it is invoked on a schedule, manually from the setup menu, or via the **SCRIPT START** command with no arguments specified. Note: Arguments do not work when running scripts interactively.

Start Script Now when selected immediately starts the script using the configured arguments. This item has no effect if the script is already running.

Stop Script Now when selected immediately stops the script if it is running. This may take up to 30 seconds before it actually stops the script. If Run Always is set then the script will restart immediately after ending.

Detailed Status displays detailed information about the script (example shown below).

```
TeleBoss 850 - Script 1 Status

Current state: Disabled
Name:  Goodbye World
File Name: goodbye.lua
Schedule: Manual
Arguments:  argument 2
Open devices:
Last start time: <never>

Press a key to continue...
```

Manage Script Files

```
TeleBoss 850 - Manage Script Files
A) List Script Files
B) View Script File
C) Edit Script File
D) Delete Script File
E) Download Script File to Unit
F) Upload Script File From Unit
```

List Script Files displays a list of all script files contained on the unit. Equivalent to the **SCRIPT DIR** command.

View Script File displays the contents of the selected script file. Equivalent to the **SCRIPT SHOW** command.

Edit Script File bring up the selected script file, or a new blank one, in the VI text editor. Equivalent to the **SCRIPT EDIT** command.

Delete Script File deletes the selected script file. Equivalent to the **SCRIPT DELETE** command.

Download Script File to Unit transfers a script file to the unit. Equivalent to the **SCRIPT GET** command.

Upload Script File From Unit transfers a script file from the unit. Equivalent to the **SCRIPT PUT** command.

Script Programming Guide

Scripts are written using the Lua programming language. In addition to built-in Lua constructs and the standard Lua libraries, the 'omni' library provides an interface to Asentria-specific functionality.

A complete list of Asentria-specific functions can be found in OmniLua Function List. Additional OmniLua scripting information can be found in the Scripting FAQ.

Scripting FAQ

1. How do I post records to a database file using a script?
2. How do I retrieve or change a setting?
3. How do I create a custom setting?
4. Which functions should a well-behaved script contain?
5. How do I save frequently-changing data so it survives a power cycle?
6. How can I generate an event?

How do I post records to a database file using a script?

In order to post a record to the database, the script creates a table that defines the records, then passes it to the a_lib.DBPostRecord() function.

How do I retrieve or change a setting?

Settings are accessed from a script using their corresponding Setting Key, via the a_lib.AccessSetting() function. This function returns the value of the setting in text format. Simply pass the name of the Setting Key by itself to read the value. To modify the value, append "= <value>" to the Setting Key, where <value> is the desired value for the setting.

How do I create a custom setting?

There are 200 settings keys available to the write of a script to store values.

- One hundred of these keys are for nonvolatile settings - settings that do not change often and need to survive resets. Those keys are:

  ```
  scripting.nvstring[x]
  scripting.nvint[x]
  ```

  Where X can range from 1 to 50. Obviously there is a group of settings for string values and a group of settings for integer values. Examples of this would be things like sitenames or IP addresses.

- One hundred settings are also available for volatile settings that do not have to survive a reset.

  ```
  scripting.vstring[x]
  scripting.vint[x]
  ```

  Strings can be up to 64 characters long. Integers can go from 0 to 2,147,483,647. These settings would be used to store values that can change often such as a signal strength or temperature.

Additionally these settings are tied to SNMP variables so that they are available to any SNMP based network management system available. This allows the script to be able to send and receive data via SNMP and thereby creating the possibility of our units acting as a true SNMP proxy for another device.

```
scripting.nvstring[x] = SNMP object: scrNonVolatileString.x
scripting.nvint[x] = SNMP object: scrNonVolatileInt.x
scripting.vstring[x] = SNMP object: scrVolatileString.x
scripting.vint[x] = SNMP object: scrVolatileInt.x
```

Through the use of a_lib.AccessSetting( ) a script can read and write SNMP values. This allows a script to actually and easily act as a proxy for a device that is not already SNMP compatible. This has very broad range application. Some simple examples are:

1. Displaying the last 50 lines of a log file.
2. Displaying the current single strength of a radio that isn't SNMP compatible
3. Allowing the SNMP manager to issue an SNMP set and having that translated into a serial command on a remote device.
4. Having a set of a number value issued by a Network Management System and having a value set to a certain level on a remote non-networked device.

Which functions should a well-behaved script contain?

Any script that might run for more than a few seconds should watch for a shutdown message from the system. This is done using the a_lib.CheckMessages() function. The purpose of this function is to allow communication between scripts, and to allow a script to watch for system messages.

When the script receives the Terminate message (Message ID 13808) it should stop whatever it is doing and shut down gracefully, using the a_lib.Exit() function. If a script does not check for and respond to the Terminate message, it will be shut down by the system about 20 seconds after the message is initially sent.

If a script uses a device, such as an IO port, for more than a couple of seconds, it should likewise listen for the Yield Device message (Message ID 13819). This way, if a passthrough session is attempted to that port, the script could potentially close the device so the passthrough session can use it.

How do I save frequently changing data so it survives a power cycle?

Using settings or even a temporary file to save frequently-changing non-volatile data is not a good idea, as it could result in premature failure of the flash memory.

You can use the a_lib.DBDeleteAndPost() function to save up to 500 characters of text (or any other data that can be stored in a Lua string). This stores the data in a special area of the database, which has provisions for being maintained regardless of power loss. The data can be retrieved using the normal methods (i.e. a_lib.DBGetRecord(), where the file is specified as AUX1 or whatever).

Note: DIR ALL displays a directory of all records in the unit in all files.

How can I generate an event?

A script can generate an event using the function a_lib.PostEvent(). This function allows the script to specify event message text, actions, class, and trap number (in case one of the specified actions is a trap). The actions are not specified directly; rather, they are specified by passing the Setting Key of any action setting that specifies the desired group of actions. It doesn't matter which action setting is used, as long as its setting reflects the desired action

OmniLua Function List

IO Functions

a_lib.CloseDevice
a_lib.Flush
a_lib.GetCharacter
a_lib.GetLine
a_lib.OpenDevice
a_lib.SendString
a_lib.SetDTR
a_lib.TCPClose
a_lib.TCPConnect
a_lib.UDPClose
a_lib.UDPListen
a_lib.UDPOpen
a_lib.UDPReceive
a_lib.UDPSend
a_lib.WaitForString
a_lib.CTSHigh

Database Functions

a_lib.DBDeleteAndPost
a_lib.DBDeleteRecords
a_lib.DBGetRecord
a_lib.DBGetRecordCount
a_lib.DBLockFile
a_lib.DBUnlockFile
a_lib.DBPostRecord

Miscellaneous Functions

a_lib.AccessSetting
a_lib.CheckMessages
a_lib.Decrypt
a_lib.Encrypt
a_lib.Exit
a_lib.HashFinalize
a_lib.HashInit
a_lib.HashUpdate
a_lib.InitEncryption
a_lib.PostAudit
a_lib.PostEvent
a_lib.ReadDIPs
a_lib.Relay
a_lib.SendMessage
a_lib.SetLED
a_lib.Sleep
a_lib.SNMPGet
a_lib.SNMPSet
a_lib.GetModbusValue
a_lib.SetModbusValue
a_lib.MODBUSReadRegisters
a_lib.MODBUSWriteRegisterSingle
a_lib.MODBUSWriteRegisterMultiple
a_lib.InitVirtualES

IO Functions

a_lib.CloseDevice

Description

Closes an IO device (IO port or modem) that has been previously opened, freeing it for use by other scripts or processes.

Syntax

```
a_lib.CloseDevice(handle)
a_lib.CloseDevice(name)

handle:                 The handle obtained when the device was opened
name:                   The name of the device (i.e. "com1", "mdm1", etc)
```

Returns

None.

Example

```
a_lib.CloseDevice(name)

name:                The name of the device (i.e. "com1", "mdm1", etc)
```

a_lib.Flush

Description

Removes any characters waiting to be read.

Syntax

```
a_lib.Flush(handle)

handle:              The handle obtained when the device or socket was opened
```

Returns

None.

Example

```
-- flush waiting characters
a_lib.Flush(handle)
-- send a command
a_lib.SendString(handle, "yazzo\n")
-- look for some response
response = a_lib.GetLine(handle, 2000)
```

a_lib.GetCharacter

Description

Reads a single character from the device or socket. Returns immediately whether a character is available or not.

Syntax

```
a_lib.GetCharacter(handle)

handle:              The handle obtained when the device or socket was opened
```

Returns

Character that was read, or 'nil' if no character available.

Example

```
-- prompt user
a_lib.SendString(handle, "Are you sure (y/n)? ")
-- wait for a character
repeat
   char = a_lib.GetCharacter(handle)
until char ~= nil
-- if response is Yes
if char == 'y' or char == 'Y'
   -- do stuff.....
end
```

a_lib.GetLine

Description

Reads a line of text from the device or socket. Returns when a line is received that is terminated by CR, or when the timeout, if specified, is exceeded. The maximum size of the received line must be less than 128 characters.

Syntax

```
a_lib.GetLine(handle)
a_lib.GetLine(handle, timeout)

handle:              The handle obtained when the device or socket was opened
timeout:             How many milliseconds to wait for the incoming line;
                     will wait forever if timeout is not specified
```

Returns

Returns a string containing line that was read, not including the terminating CR. Returns 'nil' if a complete line was not received before the timeout, or if more than 128 characters were received without a terminating CR.

Example

```
-- get line of input, timeout in 2 seconds
string = a_lib.GetLine(handle, 2000)
-- if we got something
if string ~= nil then
   -- do stuff with it...
end
```

a_lib.OpenDevice

Description

Opens an IO device (IO port or modem), and prevents it from being used by other scripts or processes.

Syntax

```
a_lib.OpenDevice(name)

name:                The name of the device (i.e. "com1", "mdm1", etc)
```

Returns

Returns a handle if successful, 'nil' if not.

Example

```
-- open the device
handle = a_lib.OpenDevice("com1")
if handle ~= nil then
   -- do some stuff....
   -- then close the device
   a_lib.CloseDevice(handle)
end
```

a_lib.SendString

Description

Sends a string via the specified device or socket.

Syntax

```
a_lib.SendString(handle, string)

handle:              The handle obtained when the device or socket was opened
string:              The text string to be sent
```

Returns

Returns the number of bytes sent if successful, 'nil' if not.

Example

```
-- send status message to remote terminal
a_lib.SendString(handle, "Connected successfully.\n")
```

a_lib.SetDTR

Description

Sets the state of the DTR pin on the specified IO port. The port must be configured via the 'scripting.dtrcontrol.portenable' setting for DTR to be under scripting control.

Syntax

```
a_lib.SetDTR(port, state)

name:                The name of the port (i.e. "com1", etc)
state:               Value representing the desired state (0 = low, anything
else = high)
```

Returns

Returns 1 if successful, 'nil' if not.

Example

```
-- set DTR high on io3
a_lib.SetDTR("com3", true)
```

a_lib.TCPClose

Description

Closes a TCP connection that was opened previously using a_lib.TCPConnect().

Syntax

```
a_lib.TCPClose(handle)

handle:              The handle obtained when the connection was established
```

Returns

None.

Example

```
-- connect to remote host using telnet port
handle = a_lib.TCPConnect("192.168.168.3", 23, 10, "telnet")
-- if connection successful
if handle ~= nil then
   -- wait for a prompt
   if a_lib.WaitForString(handle, "READY", 5000) ~= nil then
      -- do some stuff...
   end
   -- close the connection
   a_lib.TCPClose(handle)
end
```

a_lib.TCPConnect

Description

Establish a TCP connection with a remote host.

Syntax

```
a_lib.TCPConnect(ip_address, port)
a_lib.TCPConnect(ip_address, port, timeout)
a_lib.TCPConnect(ip_address, port, options ...)
a_lib.TCPConnect(ip_address, port, timeout, options ...)
```

```
ip_address:              The IP address of the remote host
port:                    The TCP port to connect to
timeout:                 Connection timeout, in seconds (optional - default is
30)
options:                 Strings representing socket options:
                         "telnet" - use telnet option negotiation
```

Returns

Returns a handle if successful, 'nil' if not.

Example

```
-- connect to remote host using telnet port
handle = a_lib.TCPConnect("192.168.168.3", 23, 10, "telnet")
-- if connection successful
if handle ~= nil then
   -- wait for a prompt
   if a_lib.WaitForString(handle, "READY", 5000) ~= nil then
      -- do some stuff...
   end
   -- close the connection
   a_lib.TCPClose(handle)
end
```

a_lib.UDPClose

Description

Closes a UDP socket that has previously been opened using a_lib.UDPListen().

Syntax

```
a_lib.UDPClose(handle)
handle:       The handle obtained when the socket was opened
```

Returns

Returns 1 if successful, 'nil' if not.

Example

```
See example for a_lib.UDPReceive.
```

a_lib.UDPListen

Description

Opens a UDP socket to listen for incoming frames on the specified port.

Syntax

```
a_lib.UDPListen(port number)

port number:            UDP port number to listen on
```

Returns

Returns a handle if successful, 'nil' if not.

Example

```
a_lib.UDPListen(port number)
```

a_lib.UDPOpen

Description

Opens a UDP socket for sending using a_lib.UDPSend().

Syntax

```
a_lib.UDPOpen()
```

Returns

Returns a handle if successful, 'nil' if not.

Example

```
See example for a_lib.UDPReceive.
```

a_lib.UDPReceive

Description

Receives a frame on UDP socket that has previously been opened using a_lib.UDPListen().

Syntax

```
a_lib.UDPReceive(handle, max size, timeout)

handle:       The handle obtained when the socket was opened
max size:     Maximum length of data to return; data is truncated
              to this value or 512 bytes, whichever is smaller
timeout:      How many milliseconds to wait for a frame to arrive
```

Returns

If successful, returns received data, length of received data, and remote IP address. If no frame was received, returns 'nil'.

Example

```
-- open a socket for listen on port 12345
handle = a_lib.UDPListen(12345)
-- if socket opened successfully
if handle ~= nil then
   -- loop around doing stuff
   done = false
   while not done do
      -- do some stuff....
      -- check for received frame
      data = a_lib.UDPReceive(handle, 128, 500)
      if data ~= nil then
         -- check for frame telling us to stop
         if string.find(data, "quit") then
            -- now we're done
            done = true
         end
      end
   end
   -- when done, close the connection
   a_lib.UDPClose(handle)
end
```

a_lib.UDPSend

Description

Send a frame on UDP socket that has previously been opened using a_lib.UDPOpen().

Syntax

```
a_lib.UDPSend(handle, data, target_ip, target_port)

handle:              The handle obtained when the socket was opened
data:                A string containing the data to send
target_ip:           IP address of the destination host
target_port:         Port number
```

Returns

If successful, returns number of bytes sent. If unsuccessful, returns 'nil' and error message.

Example

```
See example for a_lib.UDPReceive.
```

a_lib.WaitForString

Description

Waits for the specified string to be received on the specified handle.

Syntax

```
a_lib.WaitForString(handle, string)
a_lib.WaitForString(handle, string, timeout)

handle:                 The handle obtained when the device or socket was
opened
string:                 The text string to be sent
timeout:                How many milliseconds to wait for the string to
arrive;
                        waits forever if timeout not specified
```

Returns

Returns 1 if successful, 'nil' if not.

Example

```
-- wait 5 seconds for prompt
if a_lib.WaitForString(handle, "READY", 5000) then
   -- send a command
   a_lib.SendString(handle, "ver\n")
end
```

a_lib.CTSHigh

Description

Returns the state of the CTS pin on the specified IO port.

Syntax

```
a_lib.CTSHigh(port)

port:                   The name of the port (i.e. "com1", etc)
```

Returns

Returns 1 if the CTS pin is high, 0 if low, and 'nil' if port is not physically present.

Example

```
-- check CTS on io2
print(a_lib.CTSHigh("com2"))
1
```

Database Functions

a_lib.DBDeleteAndPost

Description

Posts data to an auxiliary database file, deleting the existing file first. The purpose of this function is to provide a way for a script to store information that needs to be preserved across resets and power-cycles of the unit. Since the file is deleted first, the data is never written to flash (which would slow things down and possibly cause premature failure of the flash part). Only the 'AUXx' files can be posted to with this function.

Syntax

```
a_lib.DBDeleteAndPost(table)

table:                Contains the following items:
numlines:             How many lines the record contains (required)
dest:                 The destination file ("aux1", "aux2", etc) (required)
1..n:                 Indexed record data as ASCIIZ data (required)
```

Returns

Returns 'true' if the record was successfully posted, or 'nil' if not.

Example

```
-- create a table
a = {}
-- two-line record
a.numlines = 2
-- fill in line 1
a[1] = timestamp
-- fill in line 2
a[2] = record_hash
-- post the record
a_lib.DBDeleteAndPost(a)
```

a_lib.DBDeleteRecords

Description
Deletes records from a database file. Oldest records are always deleted first.

Syntax

```
a_lib.DBDeleteRecords(file, numrecs)
a_lib.DBDeleteRecords(file, "all")

file:                   The file that records are to be deleted from
numrecs:                How many records to delete from the file
"all":                  Specifies that all records are to be deleted
```

Returns

Returns how many records were deleted, or 'nil' if the function was unsuccessful.

Example

```
-- get count of records in file
print(a_lib.DBGetRecordCount("file1"))
1281
-- delete 10 records
count = a_lib.DBDeleteRecords("file1", 10)
print(count)
10
-- delete all records
count = a_lib.DBDeleteRecords("file1", "all")
print(count)
1271
```

a_lib.DBGetRecord

Description

Retrieves a record from a database file.

Syntax

```
a_lib.DBGetRecord(file, recnum)

file:                   The file that records are to be deleted from
recnum:                 Zero-based number of the record to retrieve
```

Returns

Returns the record data as a string, and the record length; returns 'nil' if unsuccessful or a record is not available.

Example

```
-- get the first record in FILE1
rec, length = a_lib.DBGetRecord("file1", 0)
-- print record
print(rec)
   100300039                 5718  385               7004    0
print(length)
65
```

a_lib.DBGetRecordCount

Description

Retrieves the count of records in a database file, or the entire database.

Syntax

```
a_lib.DBGetRecordCount(file)
a_lib.DBGetRecordCount()

file:              The file to return the record count for; if not specified,
                   returns record count for entire database
```

Returns

Returns the record count, or 'nil' if unsuccessful.

Example

```
-- get count of records in file
print(a_lib.DBGetRecordCount("file1"))
1281
-- get count of records in entire database
print(a_lib.DBGetRecordCount())
13112
```

a_lib.DBLockFile, a_lib.DBUnlockFile

Description

Locks or unlocks a database file. Locking a database file prevents records from being polled or deleted by another script or process.

Syntax

```
a_lib.DBLockFile(file)
a_lib.DBUnlockFile(file)

file:                   The file to be locked or unlocked
```

Returns

Returns 1 if successful, or 'nil' if unsuccessful.

a_lib.DBPostRecord

Description

Posts a record to the database. The record can be posted directly to the database, or it can be routed through the data filters and/or data alarms.

Syntax

```
a_lib.DBPostRecord(table)

table:             Contains the following items:
numlines:          How many lines the record contains (required)
dest:              The destination file ("file1", "alarms", etc) (required)
time:              Timestamp (seconds since Epoch) (optional)
serial:            Serial number (optional)
do_filter:         Use data filters (true/false) (optional, default is false)
do_alarm:          Use data alarms (true/false) (optional, default is false)
1..n:              Indexed record data as ASCIIZ data (required)
```

Returns

Returns 'true' if the record was successfully posted, or 'nil' if not.

Example

```
-- create a table
a = {}
-- two-line record
a.numlines = 2
-- destination is file 2
a.dest = 2
-- fill in the timestamp
a.time = os.time()
-- fill in line 1
a[1] = "this is line 1, jack\r\n"
-- fill in line 2
a[2] = "and here is line 2, jill\r\n"
-- post the record
a_lib.DBPostRecord(a)
```

Miscellaneous Functions

a_lib.AccessSetting

Description

Reads or modifies a setting.

Syntax

```
a_lib.AccessSetting(setting_key)

setting_key:    Setting key name, plus equals sign and new value if modifying
```

Returns

Returns a string containing the value of the setting if successful, 'nil' if not.

Example

```
-- read a setting
print(a_lib.AccessSetting("sys.sitename"))
Data-Link
-- modify a setting
print(a_lib.AccessSetting("sys.sitename = Yakkity Yack"))
Yakkity Yack
```

a_lib.CheckMessages

Description

Checks for IPC message from the system or another script.

Syntax

```
a_lib.CheckMessages()
```

Returns

Returns the message ID of the received message, and the message string if any. Returns 'nil' if no message was received.   There are several messages that could be sent to a script from the system. A well-behaved script should call CheckMessages() periodically and handle these messages appropriately:

Message ID	Purpose
13808	Terminate. The script should do any necessary cleanup and then shut down gracefully using the a_lib.Exit() function.
13819	Yield device. A passthrough session is being initiated to a port that is currently allocated by the script. The script can close the device or terminate to allow the passthrough session to proceed, but it is not required to.

Example

```
-- check for message
message_id = a_lib.CheckMessages()
-- if it is shutdown request
if message_id == 13808 then
   -- do graceful shutdown...
end
```

a_lib.Decrypt

Description

Decrypts ciphertext that was generated using the a_lib.Encrypt() function.

Syntax

```
a_lib.Decrypt(ciphertext, length)
ciphertext:             String containing the ciphertext
length:                 Length of ciphertext string
```

Returns

Returns the decrypted data as a string, and the data length.

Example

```
a-- initialize encryption
my_key = "yaddayadda"
a_lib.InitEncryption(my_key, #my_key)
-- encrypt a message
message = "Hello world!"
ciphertext, ciphertext_length = a_lib.Encrypt(message, #message)
-- print the encrypted message
print(ciphertext)
<this will yield a bunch of garbage characters>
-- decrypt the message
decrypted_message = a_lib.Decrypt(ciphertext, ciphertext_length)
-- print it out
print(decrypted_message)
Hello world!
```

a_lib.Encrypt

Description

Encrypts data using the Blowfish encryption algorithm.

Syntax

```
a_lib.Encrypt(plaintext, length)
plaintext:              String containing the data to encrypt
length:                 Length of plaintext string
```

Returns

Returns the encrypted data as a string, and the data length.

Example

```
a-- initialize encryption
my_key = "yaddayadda"
a_lib.InitEncryption(my_key, #my_key)
-- encrypt a message
message = "Hello world!"
ciphertext, ciphertext_length = a_lib.Encrypt(message, #message)
-- print the encrypted message
print(ciphertext)
<this will yield a bunch of garbage characters>
-- decrypt the message
decrypted_message = a_lib.Decrypt(ciphertext, ciphertext_length)
-- print it out
print(decrypted_message)
Hello world!
```

a_lib.Exit

Description

Terminates the script.

Syntax

```
a_lib.Exit(result)
result:         Result code to indicate the exit status of the script
```

Returns

None.

Example

```
-- if some operation failed
if SomeOperation() ~= true then
   -- terminate with error (user-defined error code)
   a_lib.Exit(3)
end
-- otherwise do some other stuff...
-- and then terminate normally (0 is the usual 'normal' result)
a_lib.Exit(0)
```

a_lib.HashFinalize

Description

Finalizes MD5 hash value.

Syntax

```
a_lib.HashFinalize()
```

Returns

Returns a 32-character ASCII string containing the hash value.

Example

```
-- initialize hash function
a_lib.HashInit()
-- traverse string array 'lines'
for i,l in pairs(lines) do
   --- update hash with current line
   a_lib.HashUpdate(l)
end
-- finalize hash
hash = a_lib.FinalizeHash()
-- print it out
print(hash)
971CCDF7813648A532D8682B39A60CF9
```

a_lib.HashInit

Description

Initializes MD5 hashing function.

Syntax

```
a_lib.HashInit()
```

Returns

None.

Example

```
-- initialize hash function
a_lib.HashInit()
-- traverse string array 'lines'
for i,l in pairs(lines) do
   --- update hash with current line
   a_lib.HashUpdate(l)
end
-- finalize hash
hash = a_lib.FinalizeHash()
-- print it out
print(hash)
971CCDF7813648A532D8682B39A60CF9
```

a_lib.HashUpdate

Description

Updates MD5 hash.

Syntax

```
a_lib.HashUpdate(string)

string:        String value to update hash with; as a Lua string, it can
contain binary values
```

Returns

None.

Example

```
-- initialize hash function
a_lib.HashInit()
-- traverse string array 'lines'
for i,l in pairs(lines) do
   --- update hash with current line
   a_lib.HashUpdate(l)
end
-- finalize hash
hash = a_lib.FinalizeHash()
-- print it out
print(hash)
971CCDF7813648A532D8682B39A60CF9
```

a_lib.InitEncryption

Description

Initialize encryption/decryption functions with a user-specified key.

Syntax

```
a_lib.InitEncryption(key, key_length)
key:                String value to use as encryption/decryption key
key_length:         Length of key string
```

Returns

Returns 1 if successful, nil if not.

Example

```
              <<<  See example for a_lib.Encrypt function.  >>>
```

a_lib.PostAudit

Description

Posts a message to the audit log.

Syntax

```
a_lib.PostAudit(string
)
string:                 Message to post
```

Returns

None.

Example

```
-- post message to audit log
a_lib.PostAudit("Script 'Jabba the Hut' posting to audit log.")
```

a_lib.PostEvent

Description

Posts an event to the event-handling queue.

Syntax

```
a_lib.PostEvent(message_text, actions_key)
a_lib.PostEvent(message_text, actions_key, class_offset)
a_lib.PostEvent(message_text, actions_key, class_offset, trap_number)

message_text:           Event message
actions_key:            Setting key that specifies event actions
class_offset:           Event class (1-12)
trap_number:            Trap number to use if trap action is specified
```

Returns

Returns 1 if successful, 'nil' if not.

Example

```
-- set message text
msg = "Cannot communicate with switch."
-- set actions key to use (hijacking data event 100 setting in this ;Example)
key = "event.data[100].actions"
-- set event class to 3 (major)
class = 3
-- post the event
a_lib.PostEvent(msg, key, class)
```

a_lib.ReadDIPs

Description

Read the state of a bank of DIP switches.

Syntax

```
a_lib.ReadDIPs(bank)

bank:                   Which DIP switch bank to read (1-based);
                        number of DIPs varies by platform, most have
                        one bank or none
```

Returns

Returns string representing DIP switch state, from left-to-right, where '1' represents UP and '0' represents DOWN. Returns nil and error message if argument invalid or DIPs not present on platform.

Example

```
print(a_lib.ReadDIPs(1))
01001111
print(a_lib.ReadDIPs(2))
nil     Invalid DIP switch number.
```

a_lib.Relay

Description

Sets a relay output to a given state.

Syntax

```
a_lib.Relay(slot, relay, state)
a_lib.Relay(slot, relay, state, duration)

slot:           Eventsensor slot (0-16; 0 for internal)
relay:          Relay number on eventsensor (1-10)
state:          State to set relay to: "Closed", "Open", "Active", "Inactive"
duration:       Number of seconds to hold active state (optional, only
applies to active state)
```

Returns

Returns 1 if the command is successful, or 'nil' if not.

Example

```
-- set internal relay 3 to 'closed' state for 5 seconds

print(a_lib.Relay(0, 3, "closed", 5)
```

a_lib.SendMessage

Description

Posts a message to another script.

Syntax

```
a_lib.SendMessage(target_script, message_id)
a_lib.SendMessage(target_script, message_id, message)

target_script:        Target script name, or its 1-based settings index
message_id:           Message ID (used-defined) (message id's greater than
10000 are reserved)
message:              String containing message data; as a Lua string, it can
contain binary values
```

Returns

Returns 1 if the target script is running, or 'nil' if not. Note that there is no guarantee the target script will actually process the message.

Example

```
-- set target script by name
name = "Switch Handler"
-- set message id
msg_id = 5
-- send some text as the message
msg_text = "Manual override"
-- send the message
a_lib.SendMessage(name, msg_id, msg_text)
```

a_lib.SetLED

Description

Controls the state of front panel LEDs on the SL85. Supported LEDs can be set to off, steady on, or flashing at 1 cycle per second.

Syntax

```
a_lib.SetLED(led, state)

led:                 Which led to control ("alert1", "alert2", or "alert3")
state:               State to set the LED to ("off", "on", or "flash")
```

Returns

None.

Example

```
-- turn on the alert1 LED
a_lib.SetLED("alert1", "on")
-- make alert2 LED flash
a_lib.SetLED("alert2", "flash")
```

a_lib.Sleep

Description

Suspend script activities for specified amount of time.

Syntax

```
a_lib.Sleep(time_in_milliseconds)

time_in_milliseconds:   How much time to sleep, in milliseconds
```

Returns

None.

Example

```
-- wait for 5 seconds before further actions
a_lib.Sleep(5000)
```

a_lib.SNMPGet

Description

Gets and SNMP object from a host.

Syntax

```
a_lib.SNMPGet(host, community name, oid)

host:                   Host name or IP address
community name:         Community name string
oid:                    Object identifier string
```

Returns

Returns a string containing the value of the object if successful. If not successful, returns 'nil' and an error message.

Example

```
-- read time from an T850 unit
timeobj, err = a_lib.SNMPGet("10.10.5.137", "public",
"1.3.6.1.4.1.3052.9.2.8.1")
if timeobj == nil then
   print(err)
   return
end
```

a_lib.SNMPSet

Description

Sets an SNMP object to an SNMP agent.

Syntax

```
a_lib.SNMPSet(agent, community name, oid, type, value)

agent:                  Agent name or IP address
community name:         Community name string
oid:                    Object identifier string
type:                   A case-insensitive string that defines the type of
                        the value
                        Allowed types: "Int", "Integer", "String", "IP",
                        "IPAddress"
value:                  The value to be set.
```

Returns

Returns 1 if successful, otherwise returns nil and an error message.

Example

```
-- set a contact closure to event state on an S571
Return_val, err = a_lib.SNMPSet("10.10.5.137", "public",
"1.3.6.1.4.1.3052.14.2.2.2.1.5", "Int", "1" )

if Return_val == nil then
   print(err)
   return
end
```

a_lib.GetModbusValue

Description

Request a register value from a Modbus device.

Syntax

```
a_lib.GetModbusValue(handle, address, type, register)

handle:      The handle obtained when the serial port device was opened
address:     Address of the Modbus device (1-63)
type:        The type of data to return ("int16", "uint16", "int32",
"uint32", "float")
register:    The register on the Modbus device from which to obtain the value
```

Returns

Returns a string containing the requested value if successful. If not successful, returns nil plus a string containing an error message.

Note: The serial port device must already be set to the required baud rate and data format.

Example

```
-- open serial device
handle = a_lib.OpenDevice("com1")
-- request a floating point value from register 0x338 of device at address 1
value, error = a_lib.GetModbusValue(handle, 1, "float", 0x338)
print (value, error)
handle = a_lib.CloseDevice(handle)
```

a_lib.SetModbusValue

Description

Set a register value on a Modbus device.

Syntax

```
a_lib.SetModbusValue(handle, address, type, register, value)

handle:       The handle obtained when the serial port device was opened
address:      Address of the Modbus device (1-63)
type:         The type of data to store ("int16", "uint16", "int32", "uint32",
"float")
register:     The register on the Modbus device to write the value to
value:        Value to write to the reigster
```

Returns

Returns 1 if successful, otherwise returns nil plus a string containing an error message.
» **Note:** The serial port device must already be set to the required baud rate and data format.
» **Note:** It is assumed that the device stores the values high-word-first, high-byte-first, and that floating point values are stored in 32-bit IEEE format.

Example

```
-- open serial device
handle = a_lib.OpenDevice("com1")
-- set an unsigned 32-bit integer value at register 3 of device at address 2
to value -456.4
result = a_lib.SetModbusValue(handle, 2, "float", 3, -456.4)
handle = a_lib.CloseDevice(handle)
```

a_lib.MODBUSReadRegisters

Description

Read one or more registers from a MODBUS device

Syntax

```
a_lib.MODBUSReadRegisters(handle, address, register, register_count)

handle:          The handle obtained when the serial port device was opened
address:         Address of the Modbus device (1-63)
register:        The first register on the Modbus device to read
register_count:  How many 16-bit registers to read
```

Returns

If successful, returns Lua string containing register values as an array. If not successful, returns nil plus a string containing an error message.

Note: The serial port device must already be set to the required baud rate and data format.

Example

```
-- read a 32-bit unsigned integer (assumes high-word first, high-byte first)
v, error = a_lib.MODBUSReadRegisters(handle, address, register, 2)
if v ~= nil then
   value = (string.byte(v, 1) * 0x1000000) + (string.byte(v, 2) * 0x10000) +
(string.byte(v, 3) * 0x100) + string.byte(v, 4)
end

-- read a 16-bit signed integer (assumes high-word first, high-byte first)
v, error = a_lib.MODBUSReadRegisters(handle, address, register, 1)
if v ~= nil then
   -- get unsigned value
   value = (string.byte(v, 1) * 0x100) + string.byte(v, 2)
   -- adjust if negative
   if (value >= 0x8000) then
      value = -(0x8000 - (value - 0x8000))
   end
end

-- read a string that is stored in 8 16-bit registers
v = a_lib.MODBUSReadRegisters(handle, address, register, 8)
-- print the string
print(v)
Blahblahblah
-- print the string length (this corresponds to registers * 2, regardless of
ASCIIZ length)
print(#v)
16
```

a_lib.MODBUSWriteRegisterSingle

Description

Write a value to a single 16-bit register on a MODBUS device.

Syntax

```
a_lib.MODBUSWriteRegisterSingle(handle, address, register, value)

handle:          The handle obtained when the serial port device was opened
address:         Address of the Modbus device (1-63)
register:        The register on the Modbus device to write to
value:           Lua string containing value as byte array
```

Returns

Returns 1 if successful, otherwise returns nil plus a string containing an error message.

Note: The serial port device must already be set to the required baud rate and data format.

Example

```
-- write a 16-bit unsigned integer (stored with high-byte first)
value = 12345
-- populate value array (string)
v = string.char(value / 0x100, value % 0x100)
-- set the register
a_lib.MODBUSWriteRegisterSingle(handle, address, register, v)
```

a_lib.MODBUSWriteRegisterMultiple

Description

Write a value to a series of 16-bit registers on a MODBUS device.

Syntax

```
a_lib.MODBUSWriteRegisterMultiple(handle, address, register, register_count, 
value)

handle:          The handle obtained when the serial port device was opened
address:         Address of the Modbus device (1-63)
register:        The first register on the Modbus device to write to
register_count:  How many 16-bit registers to write to
value:           Lua string containing value as byte array
```

Returns

Returns 1 if successful, otherwise returns nil plus a string containing an error message.

Note: The serial port device must already be set to the required baud rate and data format.

Note: The length of the value string must be equal to or larger than register_count * 2. If the value being written is shorter, it must be padded to fulfill this requirement.

Example

```
-- write a 32-bit unsigned integer (stored with high-word first and high-byte 
first)
value = 123456
-- populate value array (string)
v = string.char(value / 0x1000000, value / 0x10000, value / 0x100, value % 
0x100)
-- set the registers
a_lib.MODBUSWriteRegisterMultiple(handle, address, register, 2, v)

-- write an ASCIIZ string that is stored in 8 registers
v = string.format("Test\0\0\0\0\0\0\0\0\0\0\0\0")
a_lib.MODBUSWriteRegisterMultiple(handle, address, register, 8, v)
```

a_lib.InitVirtualES

Description

Initialize virtual EventSensor.

Syntax

```
a_lib.InitVirtualES(setting_slot)

setting_slot:   Which setting slot to associate with the virtual eventsensor
```

Returns

Returns 1 if successful, nil plus error message if not.

Example

```
No example, but refer to EventSensor Configuration Setup section in the
Features chapter.
```

Command Reference

User Interface Commands

Note: The HELP command can give helpful context sensitive information for most commands.

Command	Summary	Syntax	Description
BYE	Disconnect from unit	BYE	Disconnect a processor session.
EXIT	Exit command processor	EXIT	Ends the console session.
HELP	Show help menu	HELP [*command*]	Displays a list of commands or context sensitive help for a specific command.
LOGOFF	Ends a processor session	LOGOFF	Ends a processor session without terminating the connection.
PING	Ping IP address	PING target_address	Performs a standard network ping function on the specified IP address.
RESTART	Restart unit	RESTART	Reset the system, same as pressing the physical reset button.
SENSORS or !	Display status of internal or external sensors	SENSORS or !	Display the status of internal or external sensors
STATUS or **?**	Display status screen	STATUS or ?	Display the status screen
STATUSW or **STATUS WIRELESS** or **?WIRE** or **?WIRELESS**	Display status of wireless modem	STATUSW or STATUS WIRELESS or ?WIRE or ?WIRELESS	Display the status of the wireless modem

Setup Commands

Command	Summary	Syntax	Description
BYPASS	Access serial ports	BYPASS [port_number]	Provide pass-through terminal access between the user and the input port.
SK	Set/get key	SK [KEY[=*value*]]	Set or get a single key See Setting Keys for more information.
SK GET	Read keys	SK GET [X|A [CUSTOM] [*filter*]]	SK GET initiates a download of Setup menu options. See Setting Keys for more information.
SK HERE	Manage individual keys	SK HERE	SK HERE allows you to set or get individual keys interactively. See Setting Keys for more information.
SK LOG	Show SK error log	SK LOG	SK LOG outputs a list of any errors generated during an SK set. See Setting Keys for more information.
SK SET	Set keys	SK SET [X|A]	SK SET puts the unit in bulk settings key upload mode. See Setting Keys for more information.
SETUP	Enter setup menu	SETUP	Opens the setup menu.

Data Release Commands

Command	Summary	Syntax	Description
ACCESS	Set file access	ACCESS [file_number]	Access a specific file for data release
CLEAR	Clear released records	CLEAR	Deletes all the records released via one of the RL commands. Records not released are not deleted.
COMPRESS	Release compressed records	COMPRESS [ON\OFF]	When ON, releases records in compressed format where spaces are replaced by an alpha character; when OFF, released records exactly as received.
ESC/ESC/ESC	Escape command	ESC/ESC/ESC	Stops the release of data. If used when releasing data in LINE mode, records not yet displayed are still considered released and will be deleted when the CLEAR command is issued.
NEXT	Sends the next record or block of records	NEXT	When WAIT is ON, the NEXT command causes records to be released according to the previously issued RELEASE command.
PRT	Set data partition	PRT	Sets a partition to separate existing data in a file from any new data
RL	Release data	RL RLn RL@nnn RLn@nnn	Releases the data stored in a data file. RL release all data in the partition; RL*n* releases just *n* number of records; RL@*nnn* release all records beginning a record number *nnn*; RL*n@nnn* release *n* number of records beginning at record number *nnn*.
RLMODE	Set the release mode	RLMODE RLMODE LINE RLMODE CBB RLMODE XMODEM	By itself, RLMODE displays what the current release mode is. Followed by LINE, CBB, or XMODEM, it sets the RLMODE to one of those methods.
TAG	Prepend a line number to records	TAG [ON|OFF]	By itself, TAG displays the current setting. When set to ON, TAG will prepend a six-digit line number (beginning with 000001) to call records as they are released.
TRIM	Delete records in a data file.	TRIM *n*	TRIM followed by a number of records will delete that many records from the beginning of a data file, whether they have been released already or not. To delete the entire file, enter TRIM followed by a number that is greater than the number of records in the file.
WAIT	Sets the "wait for next" mode	WAIT [ON|OFF]	By itself, WAIT displays what the current wait mode is. When set to ON, data release will not start after the RL command is given, but instead will "wait" until the NEXT command is sent.
ZERO	Deletes records from all data files.	ZERO	After issuing this command, the user is prompted with "Are you sure (y/n)?". Selecting "y" will delete all data in all data files (including the Audit and Event logs).

System Commands

Command	Summary	Syntax	Description
COLDSTART	Cold boot unit	COLDSTART	Restores all settings to defaults, deletes all record data, and reboots the unit.
DEFAULT	Restore MOST settings to factory defaults	DEFAULT	Resets most settings to factory default values, except for the following: • IP address • Subnet mask • Router address • Serial port baud rate and data format • Data alarm fields • Data alarm settings • Action queue Does not affect record data
DEFAULT ALL	Restore ALL settings to factory defaults	DEFAULT ALL	Restores all settings to defaults, but does not affect record data, and does not reboot the unit.
DOALARM	sends a test Asentria Alarm via TCP/IP	DOALARM [IP ADDRESS or HOST NAME]	Useful in quickly diagnosing problems and verifying setup of SitePath. If used without arguments then the DOALARM command sends a test alarm to all configured action IP hosts (**action.host[]**). If you supply an argument then the unit interprets it as a specific host (IP or DNS name) to which you want one test alarm sent.
DOMAIL	Test emails	DOMAIL	Sends a test email to all defined email addresses.
DOPAGE	Test pagers	DOPAGE	Sends a test page to all defined pagers.
DOSMS	Test SMS	DOSMS	Sends a test SMS message to each phone number configured in the Actions settings.
DOSMS [<phone #> <message>]	Test SMS to a specific phone number with message	DOSMS [<phone #> <message>]	Sends a test SMS message to a specific phone number.
DOTRAP	Test traps	DOTRAP	Sends a test trap to all defined trap managers.
PUSHNOW	Initiate an immediate FTP push of data	PUSHNOW	Initiates an immediate FTP push of data
PUSHTEST	Test connectivity to the FTP server	PUSHTEST	Tests connectivity to the FTP server
TYPE	Display file contents	TYPE [EVENTS|AUDIT]	Displays the contents of the Events or Audit file.
VER	Display unit version	VER	Displays unit hardware and software versions as well as the product and version build.

Numeric Commands

The T850 supports numeric (Ctrl-B) commands as follows:

Numeric (Ctrl-B)	Word
00	PRT (partition)
01	RL (release)
02	NEXT
06	RESEND
09	BYE
20	COUNT
21	FREE
25	CLEAR
29	BYPASS
39	ZERO
50	DEFAULT
53	COMPRESS
53,1	COMPRESS ON
53,2	COMPRESS OFF
54	WAIT
54,1	WAIT OFF
54,2	WAIT ON
59	TAG
59,1	TAG OFF
59,2	TAG ON
62	RLMODE
62,1	LINE
62,3	CBB
62,4	XMODEM
63	CRC
63,1	CRC OFF
63,2	CRC ON
68	DUPLEX
68,1	DUPLEX HALF
68,2	DUPLEX FULL

Usage Commands

Usage for certain functions (SCRIPT, SK, SSH, SSHC, TCPDUMP, TELNET, TRACEROUTE, VWB, and XF) can be displayed by simply entering the function command without any arguements, as shown below:

SCRIPT

```
>SCRIPT
Script Commands:

SCRIPT [HELP]                         Display list of script commands.
SCRIPT LIST                           Display a list of configured scripts.
SCRIPT START <script> [<args>...]     Start a script.
SCRIPT STATUS <script>                Display detailed status of a script.
SCRIPT STOP <script>                  Stop a running script.
SCRIPT RECORDS [CLEAR]                Show/clear pending script records.
SCRIPT DEVICES                        Show script device allocations.
SCRIPT GET/PUT <file> [<args>...]     Transfer script file to/from the unit.
SCRIPT DELETE <file>                  Delete a script file.
SCRIPT EDIT <file>                    Edit a script file (using VI editor).
SCRIPT DIR                            List script file directory.
SCRIPT SHOW <file>                    Display script file.
SCRIPT TEST <script>                  Enter interactive script interpreter.

>
```

SK

```
>SK
Usage:
sk key[<operator>[value]] |
    get [x|a][ filter|custom|@] |
    set [x|a] |
    here |
    help |
    log |
    shortcut [filter|custom|@]
 Where key:
    segment1.segment2....
    where segment:
        word | word[index] | word.index
        where word:
            defined by factory or scripting dictionaries
        where index:
            number | 'all'
    where referenced as:
        static: referring to one value
        indexed: referring to multiple values depending on index(es)
        enumerated: referring to a finite set of values
 Where operator:
    =: write value
    @: read/write access levels
    #: read key possible values where enumerated
    $: read key restriction class
    %: read key instance count where indexed
    +: read eventsensor index instance set
    -: reset to default value
 Where shortcut:
    g: get a
    c: get a custom
    s: set a
    ?: get a status
 Examples:
    sk get: read all keys and be prompted for transfer method
    sk get a: read all keys at terminal
    sk get x: read all keys via xmodem transfer
    sk set: write keys and be prompted for transfer method
    sk set a: write keys at terminal, delimit with 'end' on line by itself
    sk set x: write keys by transferring a file of them via xmodem to the unit
    sk get a custom: read non-default keys at terminal
    sk get a net: read all net keys at terminal
    sk g: same as 'sk get a'
    sk s: same as 'sk set a'
    sk c: same as 'sk get a custom'
    sk ?: same as 'sk get a status'
    sk here: perform key operations in interactive interface
    sk help: display this help screen
    sk <key>: read a key setting value
    sk <key>=<value>: write a key setting value
    sk <key>@: read key access levels
    sk <key>@<read level,write level>: write key access levels
    sk get a @: read all access levels at terminal
    sk <indexed-key>^: read the next key instance of an indexed key
    sk log: output log of last 'set' operation
    sk serial.i-: reset all settings under index branch 'serial' to default
    sk net-: reset all settings under non-indexed branch 'net' to default
    sk event.sensor[16]-: reset all settings for eventsensor 16 to default
>
```

SSH

```
>SSH
usage: ssh [-1246AaCfgkMNnqsTtVvXxY] [-b bind_address] [-c cipher_spec]
           [-D [bind_address:]port] [-e escape_char] [-F configfile]
           [-i identity_file] [-L [bind_address:]port:host:hostport]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-R [bind_address:]port:host:hostport] [-S ctl_path]
           [-w tunnel:tunnel] [user@]hostname [command]

>
```

SSHC

```
>SSHC
No client key exists.  Use "sshc -t rsa" to make an RSA key.
Usage: sshc [options]
Options:
  -h            Specify Host key
  -o            Specify Authorized key
  -c            Specify Client key (default)
  -k            Specify Known host key
  -n            Specify authentication banner
  -t key_type   Type of key to generate (rsa|dsa)
  -b bits       Bits to use (1024|2048) default=1024
  -s url        URL to send public client key to
(ftp://user:password@host/directory)
  -d            Delete keys/banner (default is key)
  -dd           Delete everything
  -a            Add item (authorized key, known host key, or banner)
  -l            List key(s)/banner
  -i            Use FTP active mode
  -m hostname   Specify hostname

Examples:
 1. Create the host key as 2048-bit RSA: sshc -h -t rsa -b 2048
 2. Delete the host key: sshc -dh
 3. List the host key: sshc -lh
 4. Create the client key as 1024-bit RSA: sshc -t rsa
 5. Create the client key as 1024-bit DSA and transfer as
    "Asentria_<key-type>_<serial-number>" to an FTP server:
    sshc -t dsa -s "ftp://user:password@some.ftp.server/some/directory"
    (note quotes around URL)
 6. Delete the client key: sshc -d
 7. List the client key: sshc -l, or sshc with no arguments
 8. Add authorized key(s): sshc -ao
 9. Delete all authorized keys: sshc -do
10. List authorized keys: sshc -lo
11. Add authentication banner: sshc -an
12. Delete authentication banner: sshc -dn
13. List authentication banner: sshc -ln
14. Add known host key: sshc -ak
15. Delete known host key for host 'myhost': sshc -dkm myhost
16. List known host keys: sshc -lk

Note: If SFTP push discovers a known host key has changed then you must
      reestablish its authenticity to the unit manually: first delete its
      known host key (sshc -dkm <host>) and then invoke PUSHTEST.

>
```

TCPDUMP

```
>TCPDUMP
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ixp0, link-type EN10MB (Ethernet), capture size 68 bytes

<CTRL-C to escape>
>
```

TELNET

```
>TELNET
BusyBox v1.00 (2009.09.19-20:48+0000) multi-call binary

Usage: telnet HOST [PORT]

Telnet is used to establish interactive communication with another
computer over a network using the TELNET protocol.

>
```

TRACEROUTE

```
>TRACEROUTE
Version 1.4a5
Usage: traceroute [-dFInrvx] [-g gateway] [-i iface] [-f first_ttl] [-m max_ttl]

        [ -p port] [-q nqueries] [-s src_addr] [-t tos] [-w waittime]
        host [packetlen]

>
```

VWB

```
>VWB
-------- SLOT CONTENTS ------------------
```

XF

```
>XF
Usage: XF [X|Y|Z|T|F|S|A] GET|PUT [filename] [host] [user] [directory]

>
```

Expansion Card Insertion Procedures

The T850-2 and T850-6 models can be purchased with a variety of optional Expansion Cards that are normally inserted in the expansion bays on the back panel of the unit when it is built at the factory. These Cards can also be purchased separately and inserted by field technicians after the unit has been installed in the field. When doing this, there are some specific precautions and steps that must be followed in a specific order when inserting Expansion Cards in the field:

- The field technician must take precautions to ensure he/she is electrically grounded so as not to damage the Expansion Card circuit board, or the main circuit board of the unit. Follow normal Electrostatic Discharge (ESD) procedures for handling electronics per IPC-610.
- The Expansion Card should remain in its protective ESD bag until it is time to actually insert it into the expansion bay.

Follow these steps to install an Expansion Card:

1. Unplug the power cable from the T850. Expansion Cards are NOT hot-swappable.
2. Unplug the telephone cord from the internal modem (if connected). This MUST be done before removing any expansion port cover plates.
3. Remove the two screws for any expansion bay cover plate and set the plate aside.
4. Carefully remove the Expansion Card from its protective ESD bag and slide it into the plastic rails inside the expansion bay. Visually confirm that the Card is in both rails and properly aligned.
5. Push the card until it is fully inserted in its slot.
6. Replace the two screws previously removed so the Card is held securely in the bay.
7. Place the Expansion Card label on the back panel directly above or below the Expansion Card, taking care to align the markings on the label with appropriate I/O points or ports on the Card.

» **Note:** If installing a Wireless Modem Expansion Card, attach an antenna to the SMA connector. (An antenna such as the MobilMark RMA3-900/1900 or equivalent is recommended). A rubber GSM antenna is included for a convenient interim antenna. This can be screwed on to the SMA connector for trial. The unit should not be powered up without an antenna connected to the modem.

8. Replace the telephone cord in the internal modem jack (if used).
9. Plug the power cable into the host unit.
10. After the unit reboots, proceed with connecting devices to, and configuring the Expansion Card, as necessary for the type of Card it is.

Wireless Modem

The wireless modem expansion card supports the same features as connecting directly to the T850 interface, including Telnet, FTP, SSH, and so on. It also supports PPP routing, which allows communication with devices connected to one of the local Ethernet interfaces.

The wireless EDGE modem is for use in TeleBoss products with firmware version 2.00.240 and above.
The wireless GPRS modem is for use in TeleBoss products with firmware version 2.00.330 and above.

Installation

If installing the wireless modem for the first time (not factory installed), follow these installation instructions:

- Make sure the the host T850 is powered down and remove the telephone line from the internal modem (if used).

- Insert your SIM card into the slot on the wireless modem module, with the contacts on the bottom, using the card outline printed on the circuit board as a guide.

- Remove the two screws from any of the expansion bay covers on the back panel of the T850 and set the cover aside. Carefully slide the wireless modem card into the plastic rails inside the expansion bay and push the card in all the way. Replace the two screws previously removed to hold the card securely in the bay.

    - » **Note:** When adding a SIM card to an already installed wireless modem card, remove the existing wireless modem card from the unit by removing the two outermost screws only. Do not remove the two innermost screws closest to the SMA connector.

- Attach an antenna to the SMA connector. (An antenna such as the MobilMark RMA3-900/1900 or equivalent is recommended). A rubber GSM antenna is included for a convenient interim antenna. This can be screwed on to the SMA connector for trial. The unit should not be powered up without an antenna connected to the modem.

- Power up the host unit.

Setup

In addition to installing an activated SIM card in the wireless modem card, certain settings on the host unit need to be configured for the wireless connection to work. These settings can be configured via either Setting Keys or the Setup Menus as described below. Changing any of these settings should be done with **`net.wireless.mode`** set to OFF, otherwise unexpected behavior may occur.

Setting Keys

Following are the Setting Keys used to configure the wireless modem card. All of the Setting Keys below can also be configured in the Setup menus listed in parenthesis after each.

`net.wireless.mode` `(Setup -> Modem Settings -> Wireless Modem Settings)`
Enables or disables the wireless modem. Possible values are OFF (disable modem), PERMANENT (maintain "always-on" connection with EDGE modem), and CIRCUIT-SWITCHED. The default setting is OFF.

`net.wireless.apn` `(Setup -> Modem Settings -> Wireless Modem Settings)`
The Access Point Name (APN) as defined by your wireless provider. Default setting is " ".

`net.wireless.pin` `(Setup -> Modem Settings -> Wireless Modem Settings)`
The PIN associated with the SIM card, if any.

`net.wireless.idletimeout` `(Setup -> Modem Settings -> Wireless Modem Settings)`
The period of inactivity, in minutes, after which the modem connection is recycled. The allowed range is 3-255 minutes. The default setting is 5 minutes. The purpose of this setting is to allow the modem to get reset after a period of time to ensure the modem connection is working properly.

`net.wireless.pppusername` `(Setup -> Modem Settings -> Wireless Modem Settings)`
`net.wireless.ppppassword` `(Setup -> Modem Settings -> Wireless Modem Settings)`
Used to set the login credentials for the PPP session.

`net.ppprouting.enable` `(Setup -> Network Settings -> PPP Settings -> IP Routing)`
This setting controls whether the unit routes IP traffic from PPP to an Ethernet interface specified by the destination IP address's subnet. On products which have DIP switches, this setting is mechanically locked with a DIP switch for added security. On products with this feature but without DIP switches, there is no way to lock this.

`net.eth.nat` `(Setup -> Network Settings -> Ethernet Settings -> Ethernet` *`n`* `Settings)`
This setting controls whether the unit does Network Address Translation (NAT) on routed frames egressing the unit on the specified interface. That is, when PPP routing is operating and forwarding frames received on the PPP interface (which can be the same thing as the wireless modem interface), the unit rewrites the source IP address of forwarded frames leaving the unit to the IP address of the ethernet interface on which they leave. If this setting is disabled then forwarding may still happen since it is governed only by the PPP routing settings, but the source IP address of the forwarded frames is not rewritten.

`net.wireless.defaultrouteenable` `(Setup -> Modem Settings -> Wireless Modem Settings)`
When ON, the wireless interface is set as the default route when connected (which is either never, or all the time, with our current options). When OFF, the wireless interface will not become the default route when connected. The default is OFF. For a change to this setting to take effect and if the wireless link is already up, the wireless link must be restarted. While it is possible to detect a change to this setting and automatically restart the wireless link, it is possible that an ongoing session (such as a web session, which would not be seen as an ongoing connection) could get interrupted. To avoid this restart the wireless connection, using the **WIRELESS RESTART** command. This brings down the wireless link, and it automatically comes back up with the new setting in effect.

Setup Menu

All of the **`net.wireless`** settings above can be accessed in the setup menu at: `Modem Setting -> Wireless Modem`

```
TeleBoss 850 - Wireless Modem Settings
A) Mode                                 [OFF]
B) APN                                  []
C) PIN                                  []
D) Idle Timeout (minutes)               [5]
E) Band (GPRS only)                     [DUAL-850/1900]
F) PPP/Wireless User Name               []
G) PPP/Wireless Password                [********]
H) Default Route Enable                 [OFF]
```

Operation

With **`net.wireless.mode`** set to PERMANENT (depending on the type of modem installed), the unit attempts to maintain a connection to the wireless network at all times. If the connection goes down for any reason, including inactivity, the unit immediately attempts to reconnect. When there is no activity on the link for longer than the inactivity timeout (see below), the connection is terminated and immediately restarted. If **`net.wireless.mode`** is set to OFF, wireless modem operations are terminated immediately (there may be up to a minute's delay if certain operations are pending).

The **WIRELESS RESTART** command causes the wireless modem to terminate the connection and restart it based on the current settings; this is useful if a setting other than "mode" is changed.

The default setting for the wireless connection is to NOT be the default route for outbound IP frames. A static route must be entered for any frame to be sent out on the wireless connection. If Default Route Enable is changed to ON for the wireless connection, then all IP frames that do not match an existing static route will be sent out on the wireless connection. For situations where the wireless modem is the only means of off-net access, Default Route Enable should be set to ON.

The front-panel MODEM LED shows the status of the wireless modem. If **net.wireless.mode** is set to OFF then the LED should remain unlit. When **net.wireless.mode** is set to PERMANENT the LED flashes once per second while the modem is attempting to establish a network connection. Once the connection is established, the LED blinks every 3 seconds.

Status Commands

On all products, the current status of the wireless connection can be displayed using the "**?W**" or "**STATUSW**" commands. (Note that "**?WIRE**" or "**?WIRELESS**" or "**STATUSW**" or "**STATUS WIRELESS**" are also valid commands.) The unit will respond with: "**Wireless modem status: <state>** Possible states are:

:not installed	wireless card not detected
:not enabled	net.wireless.mode=OFF
:connecting	attempting to establish connection *
:connected	connection established, no active TCP session
:active	connection established, one or more active TCP sessions
:idle	which it may be for only a moment between sessions

* if it says "Connecting" most of the time, there is a problem and it would be advisable to contact Asentria Tech Support to check the wireless modem log.

?W INFO will display Network Registration and Subscriber & Equipment information similar to the following:

```
?w info
Wireless Modem Information:

 Network Registration:
    Registration Status    : Registered to home network
    Location Area Code     : 0xCB52 (52050)
    Cell ID                : 0xCC89 (52361)
    Signal Strength        : 5 of 5 bars (0:00:06 ago)

 Subscriber and Equipment:
    IMSI                   : 310410169697053
    Phone Number           : 12069137572
    Local IP Address       : 166.130.3.202
    Manufacturer ID        : SIEMENS
    Model ID               : MC75
    IMEI                   : 010644000067887
    Revision ID            : REVISION 03.010
    Network Name           : Cingular
    (E)GPRS Status         : EGPRS attached
    Current Band           : 850/1900 MHz
    Mobile Channel         : 0135
    Mobile Country Code    : 310
    Mobile Network Code    : 410
    PLMN Color             : 3
    Base Station Color     : 7
    Max Power RACH         : 0
    Min Rx Level           : -111
    Base Coefficient       : 52
    SIM Status             : SIM inserted
    ICCID                  : 89014103211696970536
```

Troubleshooting Commands

For troubleshooting, user either the **"?W LOG"** or "**STATUSW LOG**" command. (Note that **"?WIRE LOG"** or "**?WIRELESS LOG**" or "**STATUSW LOG**" or "**STATUS WIRELESS LOG**" are also valid commands. The word "log" must be preceded by a space.) Contact Asentria Tech Support if troubleshooting is required as the log data probably will not be useful to the user.

ADSL Modem

TeleBoss units that are ADSL-modem-equipped can connect to the Internet via ADSL. This means that the unit can reach Internet hosts and have an Internet IP address but the address is completely firewalled so you will not be able to, for example, ping the unit's DSL interface IP address.

Note: Full ADSL modem functionality is only available on TeleBoss products with the “SitePath” build (version 2.03.000 or greater). If there is any question about whether your unit has the SitePath build, contact Asentria Technical Support.

Installation

If installing the ADSL modem for the first time (not factory installed), follow these installation instructions:

- Make sure the the host T850 is powered down and remove the telephone line from the internal modem (if used).
- Remove the two screws from any of the expansion bay covers on the back panel of the T850 and set the cover aside. Carefully slide the ADSL modem card into the plastic rails inside the expansion bay and push the card in all the way. Replace the two screws previously removed to hold the card securely in the bay.
- Power up the host unit.

Description of ADSL

ADSL (Asymmetric Digital Subscriber Line) is a technology where data is modulated onto higher frequencies of copper telephone lines not used for voice in such a way that upstream and downstream data rates differ. Certain Asentria TeleBoss units can have an ADSL modem expansion card installed to provide an interface to a line. The machine on the other end of the line is a DSLAM (Digital Subscriber Line Access Multiplexer). DSLAMs exist typically inside telephone company central offices (COs) but also exist in standalone hutches (remote DSLAMs).

The abbreviations "DSL" and "ADSL" are used interchangably in this documentation; where "DSL" is written, "ADSL" also applies unless the difference is explicitly specified.

Certain terms and acronyms are used throughout this guide that may require further explanation. These are hyper-linked to the Glossary at the end of the guide.

Configuration

The ADSL modem can be configured via two methods in the TeleBoss unit: command line menus or Setting Keys. For simplicity, only the Setting Keys method is discussed in this guide. However, as you are working through the configurations you are welcome to also use the related Command Line menus (Setup ->Network Settings -> DSL Settings) or web-interface menus in your TeleBoss unit to view or configure specific settings.

There are four ways to configure ADSL depending on the specifications from your ADSL and ISP providers. In some cases the ADSL provider and ISP provider are the same. For simplicity and unless otherwise specified, "ADSL provider" means the entity that provides all settings required for the unit to use the Internet over the ADSL.

The key datum to get from your ADSL provider is what type of addressing is to be used: **PPPoA** (PPP over ATM), **PPPoE** (PPP over Ethernet), **Static**, or **DHCP**. Make note of this, then proceed with configuring the ADSL modem as described below.

Set the value of the **net.dsl.type** Setting Key to either **PPPoA**, **PPPoE**, **Static**, or **DHCP** as instructed by your ADSL provider. This is the most important DSL setting since its value determines what other DSL settings are applicable to the DSL configuration. Each of these connection protocols requires specific settings, so refer to the paragraph below for the protocol you will be using. But first, there are some settings that must be configured regardless of how **net.dsl.type** is set.

Required Settings Regardless of Connection Protocol

net.dsl.vpi
This specifies the VPI (Virtual Path Identifier) used on the DSL interface. This is provided for you by your DSL provider and is required for DSL operation. Values are: 0 to 4095

net.dsl.vci
This specifies the VCI (Virtual Channel Identifier) for the DSL interface. This is provided for you by your DSL provider and is required for DSL operation. Values are: 0 to 65535.

net.dsl.encap
This controls whether the encapsulation is LLC (Logical Link Control) or VCM (Virtual Channel Multiplexed). This is provided for you by your DSL provider and is required for DSL operation. Values are LLC or VCM.

Settings for PPPoA or PPPoE

net.dsl.username
This specifies the PPP username for the DSL interface. This is provided for you by your DSL provider. Values are text strings up to 64 characters.

net.dsl.password
This specifies the PPP password for the DSL interface. This is provided for you by your DSL provider. Values are text strings up to 64 characters.

Settings for Static

net.dsl.mode
This controls whether the DSL is set up for Bridged mode or Routed mode. This is provided for you by your DSL provider. Values are BRIDGED or ROUTED.

net.dsl.ip
This is the public IP address of the unit in the case where the DSL link is active. This is essentially inaccessible from the outside world because it is completely firewalled on the unit. This is provided for you by your DSL provider. Value is a dotted quad IP address.

net.dsl.mask
This controls the mask used on the DSL interface. This is provided for you by your DSL provider. It is applicable only when net.dsl.type is STATIC. Value is a dotted quad subnet mask.

net.dsl.router
The router for the DSL interface. This is provided for you by your DSL provider. This is applicable only when net.dsl.type is STATIC. Value is a dotted quad IP address.

net.dns
This specifies Domain Name System addresses to use. This is provided for you by your DSL provider. Value is a dotted quad IP address.

Settings for DHCP

If **net.dsl.type** is DHCP then no additional settings need to be configured.

Activation

Once the DSL interface is configured it must be activated. This happens automatically or manually according to how the Start Mode setting is configured:

net.dsl.startmode Set this to MANUAL to require user intervention to raise the DSL interface, or to let a VPN (if it is configured to use DSL) raise the DSL interface when the VPN needs to use DSL. Set this to AUTO to tell the unit to automatically raise the DSL interface upon boot. Values are MANUAL or AUTO. Default setting is MANUAL.

Manual Activation

net.dsl.command Set this to 1 to manually activate the DSL interface, and set this to 0 to manually deactivate the DSL interface.

In manual activation the DSL interface will not activate unless some purpose requires it: either you tell it to activate or your ADSL-based VPN, when it is being raised, tells it to activate. If you tell the interface to activate then do this by

setting `net.dsl.command=1`. The unit returns COMPLETE, meaning it has started the activation process; it does not mean that the inteface is ready to use yet. Activation is a multistep process and may take a minute or two to complete.

If the VPN tells the interface to activate, then activation happens when the VPN raises.

Read `net.dsl.command` (or net.dsl.status) to check the status of the DSL interface.

`net.dsl.command=0` when the DSL interface is not activated
`net.dsl.command=1` when DSL activation is in process
`net.dsl.command=2` when the DSL interface is trained but not yet fully activated
`net.dsl.command=3` when the DSL interface is fully activated (ready to use for network traffic)

If the interface doesn't activate, then first check if anything about the configuration on the unit is invalid. Then check this configuration against what was specified by the ADSL provider.

Automatic Activation

In automatic activation the unit raises the DSL interface upon boot and keeps it up until it is explicitly deactivated by the user by setting `net.dsl.command=0`.

Once the interface is activated you can use it as an outbound-only interface. It is completely firewalled to the Internet. The only traffic allowed in is traffic associated with existing connections, meaning all connections must originate from unit. Pinging (ICMP), TCP, and UDP traffic is the only traffic allowed and this traffic must originate from the unit.

Data on the ADSL connection can be viewed with the `net.dsl.info.*` key branch:

`net.dsl.info.isp.ip`
Read this key to see what IP address the DSL interface is using with the ISP.

`net.dsl.info.isp.linktime`
Read this key to see how long the unit has been connected to the ISP (i.e., how long the unit has had Internet access) since the connection was started.

`net.dsl.info.isp.status`
Read this key to see whether the unit is connected to the ISP; it returns "Connected" or "Not Connected". Another key that gives the same information in a different format is `net.dsl.status`.

`net.dsl.info.isp.discreason`
Read this key to see why, if available, DSL connectivity was lost.

`net.dsl.info.link`
Read this key to see whether the unit has DSL connectivity (as opposed to ISP connectivity shown with `net.dsl.info.isp.status`).

`net.dsl.info.speed`
Read this key to see the speed of the link (provided there is DSL connectivity, as shown with `net.dsl.info.link`).

`net.dsl.info.ver.sw`
Read this key to see the ADSL modem software version.

`net.dsl.info.ver.fw`
Read this key to see the ADSL modem firmware version.

`net.dsl.info.ver.atm`
Read this key to see the ADSL modem ATM driver version.

net.dsl.info.ver.dslhal
Read this key to see the ADSL modem DSL HAL version.

net.dsl.info.ver.sarhal
Read this key to see the ADSL modem SAR HAL version.

net.dsl.info.ver.pump
Read this key to see the ADSL modem data pump version.

net.dsl.info.updated
Read this key to see the last date/time at which the values in the **net.dsl.info.*** key hierarchy were last updated. These values are updated when directed by the user (by setting **net.dsl.command** to 20) or every few seconds by the unit until the ADSL modem is connected to the ISP (at which time it doesn't update until directed by the user or ISP connectivity is lost).

DSL Status

net.dsl.status is a read-only key that displays a value that reflects the current state of the DSL interface. Values are an integer >=0.

- 0 means it is not activated (the unit is not talking to the modem, no address is usable with the ISP, the DSL is not trained)
- 1 means the interface is in an intermediate level of availability: there is no address usable with the ISP and the DSL is not trained, but the unit ***can*** talk (but not necessarily ***is*** talking) to the modem.
- 2 means the interface is in an intermediate level of availability, moreso than value "1": there is no address usable with the ISP but the DSL is trained and the unit has good communication with its DSL modem.
- 3 means the interface is fully activated: DSL is trained and there is an address usable with the ISP.

These values are analagous to modem LEDs seen on some DSL routers: power, "link", "DSL", "Internet". 0 can be though of as "power", 1 can be thought of as "link", 2 can be thought of as "DSL", and 3 can be thought of as "Internet".

Connectivity

When the interface is activated it can be used for Internet connectivity. The simplest way to use it is as ADSL gateway via the DSL routing function (see DSL Routing section).

Deactivation

Deactivation means the unit is no longer connected to the ISP provider via ADSL. Deactivate by setting **net.dsl.command**=0. When the DSL interface is deactivated the line may still be trained.

ADSL specifications

- Full rate ANSI T1.413 Issue2, ITU-T G.992.1 and ITU-T G.992.2 standards compliant
- ITU G.992.3, ITU G.992.5 and READSL2 ADSL2/2+ standards compliant
- Annex M and Annex L specification
- Downstream and upstream data rates up to 24Mbps and 1Mbps
- Reach length up to 22Kft.
- Dying Gasp functionality
- OAM F4/F5 loop back
- VC and LLC multiplexing

- Multiple protocols over AAL5 (RFC 2684 / RFC 1483)
- PPPoA (RFC 2364)
- PPPoE (RFC 2516)
- UBR, CBR, rt-VBR and nrt-VBR traffic shaping QoS

DSL Routing

DSL routing is used to make the unit route, and do network address translation (NAT) on, NAT-capable traffic (TCP, UDP, and ICMP) from the unit's Ethernet ports to the unit's DSL peer, and hence on to the Internet. For example, a PC that uses one of the unit's Ethernet addresses as its default router can browse the web via the unit's DSL connection. The DSL interface is firewalled such that only traffic related to already-existing-outgoing connections is allowed in.

Configuration

The following Setting Keys need to be configured:

`net.dsl.startmode`
Set this to AUTO to tell the unit to automatically raise the DSL interface upon boot. Set this to MANUAL to require user intervention to raise the DSL interface, or to let a VPN (if it is configured to use DSL) raise the DSL interface when the VPN needs to use DSL. Values are MANUAL or AUTO. Default setting is MANUAL.

`net.default.router`
This setting allows you to select the default router (gateway) for the unit. Each network interface has a router setting which you can configure; this is the machine on that interface to which frames will be sent if they do not route to the local network of that interface. However the unit uses only one of those configured routers at this time. As you configure router settings the unit will choose a default router for you. This is available for you to see (and override) via this **`net.default.router`** setting. The values you may choose for this setting (i.e., router addresses) must be in the set of routers which you have specified, or the special value, "DSL", which means that the DSL interface peer is the default router. For DSL Routing, set **`net.default.router`**`=DSL`.

The unit uses a routing table to determine how to send any outbound IP frame. Each entry in the routing table tells the unit how to send a frame whose destination address matches a rule in the routing table. Routing table entries are examined from most-restrictive to least-restrictive, so the default routing table entry is the last entry in the table since it is the least restrictive. It is the catch-all route: it tells the unit how to send a frame when it doesn't know how else to send it. The only routes on the unit at this time are network interface routes and the default route. Network interface routes tell the unit how to send a frame bound for a machine on one of the unit's local networks (subnets). These routes are automatically configured when you configure the address of a network interface. If an outbound frame is destined for a machine off all local networks then it is sent according to what the default route specifies. The default route specifies the default router to use for these frames.

If you have configured only one router for all of your network interfaces then you don't have to worry about this setting: the unit configures it for you and there is nothing you can override it with. The default router is engaged as soon as it is configured.

`net.dsl.routing.enable`
Set this to ON to make the unit forward frames received on either Ethernet interface (and not addressed to the unit) out the DSL interface. Frames are NAT-ed as they leave the DSL interface. Frames arriving on the DSL interface not associated with existing connections are blocked (the unit is firewalled). Note that the unit's default router must be set to DSL (**`net.default.router`**`=DSL`) for DSL routing to work. Set this to OFF to make the unit not do this. Values are: ON or OFF. Default is OFF.

`net.dsl.override`
Set this to a non-zero value to enable ADSL web configuration access on the TCP port specified by the value. Set this to 0 to disable web configuration access. Values are: 0 to 65535. Default is 0.

`net.dsl.cmd`
This has the same behavior as **`net.dsl.command`**.

`net.dsl.status`
Upon read this returns 0, 1, 2 or 3. Refer to the **net.dsl.status** description above for further details.

DSL Routing Example

1) Configure the unit so it sits on an Ethernet network.

2) Enter the following keys to configure the unit for routing:

```
net.dsl.startmode=manual
net.default.router=dsl
net.dsl.routing.enable=on
```

3) Say the DSL provider sent you these settings:

```
PPPoA (VCM)
VPI: 0
VCI: 38
Username: dsluser
Password: dslpassword
```

4) Enter the following Setting Keys to configure the unit accordingly:

```
net.dsl.type=pppoa
net.dsl.mode=vcm
net.dsl.vpi=0
net.dsl.vci=38
net.dsl.username=dsluser
net.dsl.password=dslpassword
```

5) Enter the following function key to raise the DSL interface:

```
net.dsl.cmd=1
```

6) Upon setting this key to 1 the unit begins the process of raising the DSL interface. You can query the status of the DSL interface by reading the **net.dsl.status** function key. To lower the DSL interface, set:

```
net.dsl.cmd=0
```

7) After a minute or two this key (or the **net.dsl.status** key) will return 3. If something went wrong then it will stay at 1 or 2 in which case the configuration should be rechecked.

8) To make the interface raise upon boot, enter:

```
net.dsl.startmode=auto
```

9) Test the connection by pinging an Internet host from the unit. Once it is verified good, proceed to configure machines which will use the unit as a DSL router. On these machines set their default router to the unit's Ethernet IP address (address that is on the same subnet as these machines). Optionally you can configure this same address as a DNS server for these machines. Test the routing connection by pinging an Internet host from these machines.

DSL Glossary

ATM
Asynchronous **T**ransfer **M**ode is a network technology based on transferring data in cells or packets of a fixed size. The cell used with ATM is relatively small compared to units used with older technologies. The small, constant cell size allows ATM equipment to transmit video, audio, and computer data over the same network, and assure that no single type of data hogs the line.

DHCP
Dynamic **H**ost **C**onfiguration **P**rotocol, a protocol for assigning dynamic IP addresses to devices on a network. With dynamic addressing, a device can have a different IP address every time it connects to the network.

DSLAM

A **D**igital **S**ubscriber **L**ine **A**ccess **M**ultiplexer is a mechanism at a phone company's central location that links many customer DSL connections to a single high-speed ATM line. When the phone company receives a DSL signal, an ADSL modem with a splitter detects voice calls and data. Voice calls are sent to the PSTN (Public Switched Telephone Network), and data are sent to the DSLAM, where it passes through the ATM to the Internet, then back through the DSLAM and ADSL modem before returning to the customer's PC or networked-device.

LLC and VCM

Logical **L**ink **C**ontrol and **V**irtual **C**hannel **M**ultiplexing are methods of encapsulating data on an ATM communication link. Encapsulation is the process of storing cells from the foreign protocol inside PPP frames.

PPP

Point-to-**P**oint **P**rotocol is a method of connecting a PC or networked-device to the Internet.

Setting Keys

A Setting Key is a "<setting> = <value>" statement. <setting> is a series of keywords that describe a particular function of the unit, or setting. These keywords are separated by periods, for example **`net.dsl.startmode`**. The current value of a Setting Key can be obtained by typing **sk <setting>** at the command line and pressing the Enter key. A new value for a Setting Key can be set by typing **sk <setting> = <value>** and pressing the Enter key.   The value must be valid for that particular Setting Key, and the unit will respond with COMPLETE when it is accepted. If the value is invalid, the unit will respond with Invalid Value.   Contact Asentria Tech Support for more information on Setting Keys if necesary.

Signal-to-noise ratio

Signal-to-noise ratio is an electrical engineering concept defined as the ratio of a signal power to the noise power corrupting the signal. In less technical terms, signal-to-noise ratio compares the level of a desired signal to the level of background noise. The higher the ratio, the less obtrusive the background noise is.

Trained

This refers to the general ability of a modem to adjust itself to optimize the communication channel. When a modem modulates data on a line, the communication infrastructure degrades the data. Some of this degradation is due to noise and some of it is due to the modem's own echo. Part of training the modem (also sometimes referred to as "training the line") involves having the modem select optimal signal-to-noise ratio as well as teaching the modem what its own "voice" (its echo) sounds like on the line. A modem receives not only data from the other modem but also its own echoes, like when you yell to someone across a canyon and listen for their response; training helps the modem separate its own echos from the signal from the other modem.

VCI

A **V**irtual **C**hannel **I**dentifier is a unique identifier which indicates a particular virtual circuit on a network. **I**t is a 16-bit field in the header of an ATM cell. The VCI, together with the VPI (Virtual Path **I**dentifier) is used to identify the next destination of a cells as it passes through a series of ATM switches on its way to its destination.

VPI

Virtual **P**ath **I**dentifier refers to an 8-bit (user to network packets) or 12-bit (network-network packets) field within the header of an ATM cell. The VPI, together with the VCI (Virtual Channel **I**dentifier) is used to identify the next destination of a cell as it passes through a series of ATM switches on its way to its destination. VPI is useful to reduce the switching table for some Virtual Circuits which have common path.

VPN

Virtual **P**rivate **N**etwork is a network that is tunneled (the virtual part), typically across a public network, and secured (the private part).

Battery Backup Module

The TeleBoss 850-2 and 850-6 are available with an optional battery backup that provides backup power for the unit in the event of power loss.

Setup

Ensure the battery enable/disable switch is in the 'enable' position. There is no other setup associated with using the battery module, nor are there any settings related to it.

Operation

As long as the battery enable/disable switch is in the 'enable' position, the battery will be available in case of power loss. The amount of time that the host unit can run off battery power depends on various things including the state of battery charge at the time, and the number and type of optional devices installed in the host unit.

If the unit is running on battery power, and the battery enable/disable switch is changed to the 'disable' position, the host unit will immediately shut down.

The host unit cannot be started up from the battery. This is because battery relay (which connects the battery power to the system) is open when no power is applied; it gets closed once the unit starts up and the battery manager application runs. Only at that point does battery power become available.

The status of the battery module can be determined from the command processor via the battery status command.

Basic Status

```
>status battery
Battery Status

Enable switch position: ON
Running on battery: YES (0:05:13
```

Note that the command can also be invoked in a more abbreviated format such as **"? BATTERY"**, "**STATUSB**" or even "**?B**".

When the charging current goes below 100mA, the charging voltage is switched from high (7.35 volts) to low (6.85 volts).

When running on battery power, if the battery voltage falls below 5.435 volts, the unit shuts down. Several warning messages are sent to all open command processors as the battery voltage gets low.

Appendices

User Rights Table

Each command has an associated minimum user right required to execute it. (Unlike the minimum user rights for settings, these cannot be changed for any command; they are hard-coded.) Here is a list of TeleBoss commands and their associated minimum user right numbers and aliases.

Command	**Minimum required right number and alias**
!	2 VIEW
?	2 VIEW
access	3 ADMIN1
addlf	3 ADMIN1
bye	0 NONE
bypass	3 ADMIN1
clear	3 ADMIN1
clog	6 MASTER
coldstart	6 MASTER
compress	3 ADMIN1
count	3 ADMIN1
crc	3 ADMIN1
default	6 MASTER
del	6 MASTER
delete	3 ADMIN1
destroy	6 MASTER
doalarm	2 VIEW
domail	2 VIEW
dopage	2 VIEW
dosms	2 VIEW
dotrap	2 VIEW
dir	2 VIEW
duplex	3 ADMIN1
exec	6 MASTER
execc	6 MASTER
exit	0 NONE
factory	6 MASTER
free	2 VIEW
ftest	6 MASTER
help	0 NONE
inits	6 MASTER
iprc	2 VIEW
ip	6 MASTER
kill	6 MASTER
logoff	2 VIEW
mbyte	6 MASTER
mfill	6 MASTER
modemtalk	6 MASTER
mword	6 MASTER
netstat	6 MASTER
next	3 ADMIN1
ntpq	2 VIEW
password	6 MASTER
ping	3 ADMIN1
post	6 MASTER
posterror	6 MASTER
prt	3 ADMIN1

pushnow	3 ADMIN1
ppp	3 ADMIN1
pushtest	3 ADMIN1
resend	3 ADMIN1
restart	0 NONE
restore	3 ADMIN1
rlmode	3 ADMIN1
rl	3 ADMIN1
sa	3 ADMIN1
sx	3 ADMIN1
script	5 ADMIN3
sensors	2 VIEW
setup	3 ADMIN1
sk	2 VIEW
spawn	6 MASTER
spawnc	6 MASTER
ssh	6 MASTER
sshc	6 MASTER
sslc	6 MASTER
stamp	3 ADMIN1
status	2 VIEW
switch	2 VIEW
tag	3 ADMIN1
tcpdump	6 MASTER
tcplog	6 MASTER
telnet	6 MASTER
testtime	2 VIEW
traceroute	6 MASTER
trim	6 MASTER
type	3 ADMIN1
wait	3 ADMIN1
wireless	3 ADMIN1
wrap	3 ADMIN1
ver	2 VIEW
vw	6 MASTER
xf	2 VIEW
zap	6 MASTER
zero	3 ADMIN1

Setup Menu Permissions

Settings	**View**	**Admin1**	**Admin2**	**Admin3**	**Master**
Most settings	View	X	X	X	X
Authentication				View	X
Passwords					X
Event log	View	View	View	X	X
Audit log	View	View	View	X	X
PPP dial username		View	View	View	X
PPP dial password					X
Caller ID				View	X

Control Characters

Some of the following control characters may be used in various functions within the T850, including CRC mode for AsentriaAlarms and the Escape Key.

Char	Dec	Hex	Control Key	Control Action
NUL	0	00	^@	Null
SOH	1	01	^A	Start of heading
STX	2	02	^B	Start of text
ETX	3	03	^C	End of text
EOT	4	04	^D	End of transmission
ENQ	5	05	^E	Enquiry
ACK	6	06	^F	Acknowledge
BEL	7	07	^G	Bell
BS	8	08	^H	Backspace
HT	9	09	^I	Horizontal tab
LF	10	0A	^J	Line feed
VT	11	0B	^K	Vertical tab
FF	12	0C	^L	Form feed
CR	13	0D	^M	Carriage return
SO	14	0E	^N	Shift Out
SI	15	0F	^O	Shift In
DLE	16	10	^P	Data link escape
DC1	17	11	^Q	XON
DC2	18	12	^R	Device control 2
DC3	19	13	^S	XOFF
DC4	20	14	^T	Device control 4
NAK	21	15	^U	Negative acknowledge
SYN	22	16	^V	Synchronous idle
ETB	23	17	^W	End transmission block
CAN	24	17	^X	Cancel
EM	25	19	^Y	End of medium
SUB	26	1A	^Z	Substitute
ESC	27	1B	^[	Escape
FS	28	1C	^\	File separator
GS	29	1D	^]	Group Separator
RS	30	1E	^^	Record Separator
US	31	1F	^_	Unit Separator

Internal Modem Guidelines

The internal modem supplied with this product complies with Part 68 of the FCC Rules and Regulations. The labeling on the modem provides the FCC Registration number and the Ringer Equivalence Number (REN) for the modem. This information is also listed below. You must provide, upon request, this information to your telephone company.

The REN is useful to determine the quantity of devices you may connect to a telephone line and still have all of these devices ring when the number is called. In most, but not all areas, the sum of the RENs of all devices connected to one line should not exceed five (5.0). To be certain of the number of devices you may connect to a line, as determined by the REN, you should contact the local telephone company to determine the maximum REN for your calling area.

If the modem causes harm to the telephone network, the telephone company may temporarily discontinue your service. If possible, they will notify you in advance. If advance notification is not possible, you will be notified as soon as possible.

Your telephone company may make changes in its facilities, equipment, operations or procedures that could affect proper functioning of your equipment. If they do, you will be notified in advance to give you an opportunity to maintain uninterrupted telephone service.

If you experience trouble with the modem, contact Asentria Technical Support for information on obtaining service or repairs. The telephone company may ask you to disconnect the device from the network until the problem has been corrected or until you are sure that the device is not malfunctioning.

This device may not be used on coin service lines provided by the telephone company (this does not apply to private coin telephone applications which use standard lines). Connection to party lines is subject to state tariffs.

Modem	**FCC ID**	**REN**
2400 Baud Modem	EUD-5U9-BRI4480	0.8B
33.6K Baud Radicomm Modem	406CHN-31735-PT-E REN 1.1B	1.1B
33.6K Baud OmniModem	6KMUSA-34184-MME REN 0.9B	0.9B
33.6K Baud MultiModem	AU7-USA-46014-MD-E	0.1B

Canadian Department of Communications

NOTICE: The Canadian Department of Communications Label identifies certified equipment. This certification means that the equipment meets certain telecommunications network protective, operational and safety requirements. The Department does not guarantee the equipment will operate to the user's satisfaction.

Before installing this equipment, users should ensure that it is permissible to be connected to the facilities of the local telecommunications company. The equipment must also be installed using an acceptable method of connection. In some cases, the company's inside wiring associated with a single line individual service may be extended by means of a certified connector assembly (telephone extension cord). The customer should be aware that compliance with the above conditions may not prevent degradation of service in some situations.

Repairs to certified equipment should be made by an authorized Canadian maintenance facility designated by the supplier. Any repairs or alterations made by the user to this equipment, or equipment malfunctions, may give the telecommunications company cause to request the user to disconnect the equipment.

Users should ensure for their own protections that the electrical ground connections of the power utility, telephone lines and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

Caution: Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

The Load Number (LN) assigned to each terminal device denotes the percentage of total load to be connected to a telephone loop, which is used by the device, to prevent overloading.

The termination of a loop may consist of any combination of devices subject only to the requirement that the total of the Load Numbers of all the devices does not exceed 100. The load number of this unit is five.

This digital apparatus does not exceed the Class A limits for Radio noise emissions from digital apparatus set out in the interference-causing equipment standard entitled "Digital
Apparatus", ICES-003 of the Department of Communications.

AVIS: - L'étiquette du ministère des Communications du Canada identify le materiel homologué. Cette étiquette certifie que le matériel est conforme a certaines normes de protection, d'exploitation et de sécurité des réseaux de télécommunications. Le Ministère n'assure toutefois pas que le matériel fonctionnera a la satisfaction de l'utilisateur.

Avant d'installer ce matériel, l'utilisateur doit s'assurer qu'il est permis de le raccorder aux installations de l'entreprise locale de télécommunication. le matériel doit également etre installé en suivant une méthod acceptée de raccordement. Dans certains cas, les fils intérieurs de l'entreprise utilisés pour un service indivuduel a linge unique peuvent etre prolongés au moyen d'un dispositif homologué de raccordement (cordon prolongateur téléphonique interne). L'abonné ne doit pas oublier qu'il est possible que la conformité aux conditions énoncées ci-dessus n'empechent pas la dégradation du service dans certaines situations. Actuellement, les entreprises de télécommunication ne permettent pas que l'on raccorde leur matériel a des jacks d'abonné, sauf dans les cas précis prévus pas les tarrifs particuliers de ces entreprises.

Les réparations de matériel homologué doivent etre effectuées pas un centre d'entretien Canadien autorisé designé par le fournisseur, La compagnie de télécommunications puet demander a l'utilisateur de débrancher un appareil a la suite de réparations ou de modifications effectuées par l'utilisateur ou a cause de mauvais fonctionnement.

Pour sa propre protection, l'utilisateur doit s'assurer que tous les fils de mise a la terre de la source d'energie electrigue, des lignes téléphoniques et des canalisations d'eau métalliques, s'il y en a, sont raccordés ensemble. Cette précaution est particuliérement importante dans les régions rurales.

Avertissement. - L'utilisateur ne doit pas tenter de faire ces raccordements lui-meme; il doit avior recours a un service d'inspection des installations électriques, ou a electricien, selon le cas.

L'indice de charge (IC) assigné a chaque dispositif terminal indique, pour éviter toute surcharge, le pourcentage de la charge totale qui peut etre raccodée a un circuit téléphonique bouclé utilisé par ce dispositif. La terminaison du circuit bouclé peut etre constituée de n'import quelle combinaision de dispositif, pourvu que la somme des indices de charge de l'ensemble des dispositifs ne dépasse pas 100. L'indice de charge de cet produit est 5.

Cet appereil numérique respecte les limites de bruits radioélectriques applicables aux appareils numériques de Classe A prescrites dans la norme sur le matériel brouilleur :"Appareils Numériques", NMB-003 édictée par le ministre des Communications.

Warranty Information

Asentria Corporation hereby warrants that it will, as the buyers sole remedy, repair or replace, at its option, any part of the T850 which proves to be defective by reason of improper materials or workmanship, without charge for parts or labor, for a period of 12 (twelve) months. This warranty period commences on the date of first retail purchase, and applies only to the original retail purchaser.

To obtain service under this warranty, you must obtain, by telephone, postal letter, or email, a return authorization number from Asentria Technical Support. This authorization number may be obtained by contacting Asentria Technical Support at the address and/or phone number below. The defective unit is to be returned to Asentria with shipping prepaid, and the return authorization number must be clearly marked on the outside of the package containing the defective unit.

The dealer's bill of sale or other satisfactory proof of the date of purchase may be required to be presented in order to obtain service under this warranty.

This warranty applies if your T850 fails to function properly under normal use and within the manufacturer's specifications. This warranty does not apply if, in the opinion of Asentria Corporation, the unit has been damaged by misuse; neglect; or improper packing, shipping, modification, or servicing by other than Asentria or an authorized Asentria Service Center.

In no event shall Asentria Corporation be liable for any loss, inconvenience or damage, whether direct, incidental, consequential or otherwise, with respect to the T850. Asentria Corporation's liability shall be limited to the purchase price of the T850. No warranty of fitness for purpose, or of fitness of the T850 for any particular application is provided. It is the responsibility of the user to determine fitness of the T850 for any particular application or purpose.

This warranty gives you specific legal rights. These rights may vary from state to state, as some states do not allow limitations on liability.

You may request information on how to obtain service under this warranty by contacting Asentria Technical Support at the address and phone number below:

Asentria Technical Support
1200 North 96th St.
Seattle, WA 98103
206.344.8800
support@asentria.com

www.asentria.com