2-System and Networking Objects

System and Networking Objects

This section describes the ACE objects for the top-level system objects, the configuration of serial and network interfaces, and other networking protocols.

System Configuration

The RediGate System object is the parent object for all the other objects in the configuration. It contains some base properties for the RediGate operation.

Attributes	Function
Object Type	System
Parent(s)	N/A

Properties	Values
Unit Address	Enter a valid and unique unit address between 1 and 255. *The Unit Address is used in some host systems:* *- Identifies this unit in an Elecsys HCP (must be unique)* *- Identifies this unit in an Elecsys OPC Server (must be unique)* - May be part of topic string to MQTT broker/OPC Server, if configured in the MQ RBE object (must be unique if using Topic option with "UnitAddress") *Note that the "Unit Address" property is different from any individual Field Unit being polled and reported to the host. The Unit Address refers to the RediGate itself, and must be explicitly configured to be unique across all devices reporting to the HCP or MQTT/OPC Server.*
Unit Name	Depending on the configuration, the User Name may be up to a length of either 13 or 128 characters. *The Unit Name is used to identify this unit in diagnostics menus and is also used with some host systems:* *- Identifies this unit in an Elecsys HCP (must be unique)* *- Identifies this unit in an Elecsys OPC Server (must be unique)* - May be part of topic string to MQTT broker/OPC Server, if configured in the MQ RBE object (must be unique if using Topic option with UnitName) Note that this "Unit Name" property is different from any individual Field Unit name configured in other ACE objects and reported to the host. The Unit Name refers to the RediGate itself, and must be explicitly configured to be unique across all devices reporting to the HCP or MQTT/OPC Server.
User Name	Enter a valid password between 1 and 13 characters. *User name is a NULL terminated character string, used for setting up a user account for MMI and file system access.*
Password	Enter a valid password between 1 and 8 characters. *Password is a NULL terminated character string, used for setting up a user account for MMI and file system access. A NULL string value will disable password protection for the MMI over a network connection, but a 'root' user will still be able to access the MMI.*
Date Format	Select the date format to specify which time zone to use for the internal system clock. The values in parentheses are the Linux localtime filenames used for adjusting the clock. *Options are:* *Universal Coordinated Time=GMT (UTC)* *U.S. Eastern Time (EST5EDT)* *U.S. Eastern, no daylight savings (EST)* *U.S. Central Time (CST6CDT)* *U.S. Mountain Time (MST7MDT)* *U.S. Mountain, no daylight savings (MST)* *U.S. Pacific Time (PST8PDT)* *Great Britain Time (GB)* *Western Europe Time (WET)* *Central Europe Time (CET)* *Eastern Europe Time (EET)*

Linux System Object

The RediGate system consists of RediGate application software that runs within a Linux operating system. The "RediGate" functionality is generally concerned with protocol gateway, SCADA operations and data communication (Master/Slave channels, Terminal Server/Client, etc.). Other functions (primarily networking) are handled directly in the Linux operating system.

The System Configuration object (described in the section System Configuration) and many of its child objects are used to configure software components specifically related to the core application software. Other ACE objects (such as network settings, NTP, etc.) are used to configure components which are part of the Linux operating system, but which operate independently of the gateway application.
In many cases, a customer may wish to use ACE to fully manage all of these Linux services, including the disabling of non-configured items. For instance, a customer may wish to explicitly disable the DHCP Server, which can be done by disabling or removing the DHCP Server ACE object from the configuration. Other customers may desire certain network or Linux system components to be managed by their IT department, separate from the RediGate application and ACE configuration environment (in keeping with security or other corporate network administration policies).

The Linux System icon is included in the configuration to allow the flexibility of independent management of certain Linux-level OS features. The "Linux System" ACE object tells the RediGate how to handle these services when processing a new ACE configuration file. It includes a list of Linux system components which are outside the specific RediGate application capabilities, and allows a customer to determine whether the service will be managed by the ACE configuration or separately.

Attributes	Function
Object Type	LinuxSystem
Parent(s)	System
Instance	Must be 0

Properties	Values
Linux Services	Click the **Edit Table** button to edit the list of system-managed services, defining how the RediGate should handle these services. **Service Name** – Select the Linux service to manage from the list of available services. Services in the Linux System object include: ***Hostname*** *– Configured in the System Configuration object, this sets the Linux system name when logging into the command prompt.* ***User Password*** *– Configured in the System Configuration object, this option allows the user password for the user MMI (not the 'root' password) to be configured in ACE or managed separately. The 'root' and other Linux system accounts must always be managed by a system administrator, separate from the ACE configuration.* ***Timezone*** ***Ethernet (eth0)*** ***Ethernet (eth1)*** ***PPP (ppp0)*** ***Route Table (sroutes)*** ***DHCP Server*** ***NTP*** ***SNMP*** **ManagedBy** – Select how the service should be managed whenever a new ACE configuration is downloaded to the unit. Options include: ***Customer-managed service*** *– If the object representing the Linux service does not appear in ACE (including the Hostname, User Password and Timezone properties which appear in the System Configuration object), the RediGate will allow this service to be managed separately from the ACE configuration. If the object is disabled or deleted from ACE, it will not be disabled in Linux. A user with 'root' access will need to manage these services themselves.* ***ACE should manage service*** *– If the object representing the Linux service does not appear in the ACE configuration or is disabled, its function will be disabled in Linux. Hostname, User Password and Timezone will be set in Linux according to the properties which appear in the System Configuration object.*

Networks

The Networks placeholder is the parent for objects that define the physical communications connections for the system. The RediGate is capable of supporting up to a certain number of serial and network communication ports with a wide array of operational parameters. The maximum number of ports available depends on the limitations of the individual hardware platform (see the appropriate *Hardware Manual* for details).

> **NOTE**: Configuration of hardware that is not present on the RediGate may cause errors in operation of the RediGate software.

Attributes	Function
Object Type	Networks
Parent(s)	System
Instance	Must be 0

Ethernet Port

The Ethernet Port configuration defines the operational properties of a physical Ethernet port on the device.

Attributes	Function
Object Type	EtherPort
Parents	System Networks
Instance	Enter a unique instance number between 0 and 16. The instance number is ***required*** *to correspondwith the Linux interface name for the Ethernet. Instance #0 and 1 configure the built-in 'eth0' and 'eth1' ports, and Instance #2 configures the 'eth2' port of the optional AIM104-ETHER.*

Properties	Values
Network Card Type	(Included only for compatibility with older ACE objects)
Network Card Address	(Included only for compatibility with older ACE objects)
Network Card IRQ	(Included only for compatibility with older ACE objects)
Network Card DMA	(Included only for compatibility with older ACE objects)
Domain Name	Enter a unique name for this interface, used in certain ACE objects to identify this network adapter. This is case-sensitive.
Network Card IP	Enter the IP address for the Ethernet adapter, in dotted notation. *To set the Ethernet to use DHCP client(to obtain an IP address, subnet, and default gateway from a DHCP server), set the Network Card IP to 0.0.0.0.*

Subnet Mask	Enter the subnet mask, in dotted notation. *Make sure that all IP interfaces are configured for non-overlapping subnets. If using DHCP client, this field is ignored and may be set to 0.0.0.0.*
Default Gateway	Enter the default gateway, which is the IP address of a router for the RediGate to connect to addresses beyond its local subnet. *If a default gateway is configured in a Routes object in the configuration, or if there is no default gateway to be configured, set this property to 0.0.0.0. If using DHCP client, this field is ignored and may be set to 0.0.0.0.*

Multi-Home

The Multi-Home configuration allows additional IP addresses to be defined on the same Ethernet interface. This object should be omitted unless more than one IP address must be defined.

Attributes	Function
Object Type	Multi-Home
Parent(s)	System Networks EtherPort
Instance	Must be 0

Properties	Values
IP Homes	Click the **Edit Table** button to edit the list of Multi-Home addresses. **Network Card IP** – Enter the additional IP address to be used, in dotted notation (do not include the primary IP address defined in the Ethernet object). **Subnet Mask** – Enter the Subnet Mask to be associated with this IP address, in dotted notation. **Default Gateway** – Enter the Default Gateway to be used with this IP address, in dotted notation.

DHCP Server

The DHCP Server is a child object to an Ethernet interface, and defines the ability to act as a DHCP server to other devices on that network, responding to DHCP requests, assigning address, subnet, default gateway, and DNS

Attributes	Function
Object Type	DHCP Server
Parent(s)	System Networks EtherPort
Instance	Must be 0

Properties	Values
LAN Interface Name	Enter the Linux interface name of the Ethernet port on which to run the DHCP server.

LAN Subnet Address	Enter the subnet address of the subnet that should be served to clients as part of the DHCP information, in dotted notation. Subnet address should follow normal IP rules (for instance, on a 192.3.1.x network with 255.255.255.0 subnet mask, the subnet address would be 192.3.1.0).
LAN Subnet Mask	Enter the subnet mask of the subnet that should be served to clients, in dotted notation.
Served Address Range Start IP	Enter the starting IP address that should be served to clients, in dotted notation.
Served Address Range End IP	Enter the ending IP address that should be served to clients, in dotted notation. The range of addresses between the Start IP and End IP determines how many DHCP clients be supported simultaneously on the interface.
Served Default Gateway	Enter the address of the Default Gateway to be served to DHCP clients, in dotted notation.
Served Domain Name	Enter the domain name to be served to DHCP clients (must be from 1 to 64 characters).
Served DNS Server Primary	Enter the address that will be served to DHCP clients as their primary DNS server, in dotted notation.
Served DNS Server Secondary	Enter the address that will be served to DHCP clients as their secondary DNS server, in dotted notation.
Served Broadcast Address	Enter the address that will be served to DHCP clients as the broadcast IP. The broadcast IP should follow normal IP rules (for instance, on a 192.3.1.x network with 255.255.255.0 subnet mask, the broadcast IP address would be 192.3.1.255).
Lease Time-Default	Enter default lease time, in seconds
Lease Time-Max	Enter the maximum lease time, in seconds.
Authoritative	Select whether or not to make this DHCP Server "authoritative." *Setting this to "No" means that if a client requests an address that the server knows nothing about and the address is incorrect for that network segment, the server will not send a DHCPNAK (which tells the client it should stop using the address.) Setting this to "Yes" will send a DHCPNAK in this case, to force the client to stop using the incorrect address on the network and immediately request a new address.*

Async Port

The Async Port configuration defines the asynchronous serial communication properties of a physical serial port. Do not configure an Async Port object for any serial port used as an IP network, such as PPP or SLIP.

> **Note:**
> Async ports can be defined as "Virtual Ports," that represent internal software links between tasks rather than actual, physical communication ports. For additional information, see the section Virtual Ports.

Attributes	Function
Object Type	AsyncPort
Parent(s)	System Networks

Instance	Enter a unique instance number between 0 and 127. When configuring physical serial ports, the instance number must match the COM port number in the Linux system. *For RediGate 1xx series, you must use* instance 2 *for COM2.* For built-in Zeus processor ports, use instance numbers 0 through 3 for COM0-COM3 (Linux /dev/ttyS0 through /dev/ttyS3). For the AIM104-COM8 expansion card, use instance numbers 4 through 11 for COM4-COM11 (Linux /dev/ttyS4 through /dev/ttyS11). *To configure virtual serial ports, see the section Virtual Ports.*

Properties	Values
Baud Rate	Select baud rate for the Async Port speed. *For the Serial MMI port, typically use 115,200. Otherwise, set the serial properties according to the communication requirements of the external devices.*
Parity	Select the parity for the serial port (None, Odd, or Even).
Word Length	Select the data bits for the serial port (7 or 8 bits).
Stop Bits	Select the stop bits for the serial port (1 or 2 bits).
Rx Buffer Size	Enter the receive buffer size in bytes. *The receive buffer holds incoming data while waiting for processing by the application.*
Tx Buffer Size	Enter the transmit buffer size in bytes. *The transmit buffer holds outgoing data while waiting for the serial port hardware to deliver the data.* *For the Serial MMI port, use at least 2048, to allow the MMI to transmit large blocks of diagnostic data.*
Warm Up Time	Enter value for warm up time. *This is the amount of time to wait before sending data after the RTS handshaking lead has been asserted.*
Warm Down Time	Enter value for warm down time. *This is the amount of time to wait after the entire message packet has been shifted out to keep the RTS handshaking lead asserted.* *There are three modes of operation based on the Warm-up and Warm-down settings:* ***No Handshaking** – Set both the Warm-up and Warm-down to a value of -1. RTS will not be activated, CD is not required.* ***Hardware Handshaking** – Set both the Warm-up and Warm-down to a value of -1. RTS will not be activated, CD is not required.* ***Timed Handshaking** – Set either the Warm-up or Warm-down or both to positive numbers. RTS will be asserted for the configured Warm-up time, then data will be sent regardless of the condition of CTS. After data has been sent, RTS will be asserted for the configured Warm-down time, and then lowered.* When using the Async Port with an RS-485 device, modem, or external HART device, typically requires the Warm-up and Warm-down to be set to 0 or a fixed positive integer for hardware flow control.

Virtual Ports

This section describes the "Virtual Port" feature of the RediGate. Virtual Ports are configured as Async Ports which do not define physical communication hardware, but rather internal communication links.

The purpose of Virtual Ports is to connect two different internal processes that ordinarily communicate over a physical serial port. Rather than using two actual serial ports and connecting them together using a null modem cable, the Virtual Ports connect the processes internally via a software link. Data from one process is immediately transferred to the other and vice-versa.

Several rules must be understood to use Virtual Ports:

- Virtual Ports use the same object definition as Async Ports (see the section Async Port).
- Virtual Ports may be selected in objects (such as Circuits) in the same way that Async Ports are selected, and use Instance numbers 52 through 67.
- Virtual Ports must always be created and used in pairs, and pass data from one to the other in the same way as an external null modem cable between physical comm ports. COM52 is connected to COM53, COM54 to COM 55, etc.
- If you are using the Elecsys cellular modem and have "Enable Serial MUX" set to Yes, the Mux automatically opens virtual ports 72, 74, and 76 for AT command and GPS access to the modem. In this case, these sets of virtual ports may only be used with the child ports under the CellModem object. ACE objects for ports 73, 75, or 77 (the corresponding virtual pairs of ports associated with 72, 74, and 76, respectively) should be defined as child objects under the CellModem, not under Networks.

Attributes	Function
Object Type	AsyncPort
Parent(s)	System Networks
Instance	Enter a unique instance number between 0 and 127 *Virtual Ports must be* added in pairs, using instance numbers: 52 & 53, 54 & 55, up to 66 & 67

Properties	Values
Baud Rate, etc	When configuring Virtual Port definitions, all fields in the Async Port object are unused, and are simply included for compatibility with other physical Async Port objects.

Cell Modem

The CellModem configuration defines the configuration for a PPP (Point-to-Point Protocol) connection to an Elecsys EModem.

Attributes	Function
Object Type	CellModem
Parent(s)	System Networks
Instance	Enter a unique instance number between 0 and 18. Instance *#0 is the configuration for the numbered interface, such as 'ppp0.*

Properties	Values
PPP Port	Select the physical communication port to be used for PPP. This should be an internal port to which the cell modem is physical connected. ***Do not** configure this port as an Async or other type of port in addition to the PPP port configuration. If there are Async and PPP objects defined for the same physical COM port, neither will work properly.*
Baud Rate	Select baud rate for the cell modem port.
Parity	Select the parity for the cell modem port (None, Odd, Even). Parity options supported are None, Odd and Even.
Word Length	Select the data bits for the cell modem port (7 or 8 bits).
Stop Bits	Select the stop bits for the cell modem port (1 or 2 bits).

Warm Up Time	Enter value for warm up time. *This is the amount of time to wait before sending data after the RTS handshaking lead has been asserted. The default entry of 0 should be used, denoting that RTS/CTS hardware handshaking will be used.*
Warm Down Time	Enter value for warm down time. *This is the amount of time to wait after the entire message packet has been shifted out to keep the RTS handshaking lead asserted.* *The default entry of 0 should be used, denoting that RTS/CTS hardware handshaking will be used.*
Domain Name	Enter the domain name. *Name used by certain tasks internally to identify different IP adapters. This is case-sensitive.*
PPP IP	Enter the PPP IP address. *This is the address at which other network devices will see this device when trying to make a connection via PPP. If connecting to a cell modem network that automatically assigns an IP address, this parameter should be configured as 0.0.0.0 for DHCP.*
Subnet Mask	Enter the subnet mask. *Should always be 255.255.255.255 for PPP.* *If a static IP is used and a Default Gateway is required to make outbound connections beyond the local subnet, the Routes object must also be configured (see the section Route).*
Connection TimeToDie	Number of seconds to operate a PPP session before killing the connection. *This time is absolute, based on the time at which the session was initiated. The PPP connection will be closed regardless if data is still being transferred when the TimeToDie timer expires. This may be used to force a dial connection to hang up to limit cell phone connection charges.* *Disable the TimeToDie by setting it to -1 if the connection should never be closed automatically.*
Modem Type	Select the type of modem being configured. This will depend on the hardware physically available on this device. Options are: • SARA-R4 (LTE/CAT-M1) • HE910 (GSM/HSPA+) • DE910-DUAL (CDMA/EVDO, Verizon)
AT Init Strings	Enter one or more optional text entries for AT commands to be sent to the modem upon initialization. Text strings are limited to 80 characters. The AT Init Strings and all built-in modem initialization commands and responses are logged in a file /tmp/modemlog.txt. *Consult modem manual for initialization parameters or other AT commands available.* ***NOTE**: For the EVDO modem, if using mobile IP (MIP) on a Verizon network, it is recommended to add the following initialization string to force the modem to use MIP rather than permitting fallback to Simple IP (SIP): AT$QCMIP=2*
Connect String	Enter the modem connect string. This is the AT command telling the modem to enter an IP data session and depends on the modem model. For CAT-M1 or GSM/HSPA+ modem (SARA-R4 or HE910), use: ATD*99***1# For EVDO modem (DE910-DUAL), use: ATDT#777
Enable Serial MUX	Select whether to use a serial multiplexer to the modem. This should typically be set to 'Enabled'. *The serial multiplexer allows the data PPP session and other diagnostics to occur simultaneously to the modem. See the sections Modem Ports 73/75/77 and AT Commands for other options that can be used*
Use as Default Gateway	Select whether to use this cell modem network as the Default Gateway. *Typically this should be set to 'Yes'.*
Network Inactivity Watchdog	Enter the number of minutes of inactivity to be allowed, before the modem and PPP session will be restarted.

APN	Enter the APN (Access Point Name), which is the network gateway through which the cell modem will connect. This is typically dependent on the cellular carrier that the modem is activated on, and may be a public or private APN depending on the cellular account settings. *Used for CAT-M1 or GSM/HSPA networks only; leave blank for EVDO networks.*
Use Peer DNS	Select whether to use DNS from the cellular network provider.
Authentication Type	Select the type of PPP Authentication required by the cellular network. This setting and the Auth User Name and Password will depend on the cellular account activation. *Authentication types are:* ***Disabled*** ***PAP Authentication*** ***CHAP Authentication***
Auth User Name	Enter the user name required by the cellular network for PAP or CHAP authentication. *User Name is case sensitive and limited to 32 characters.*
Auth Password	Enter the password required by the cellular network for PAP or CHAP authentication. *Password is case sensitive and limited to 32 characters.*

Modem Ports 73/75/77

When the Cell Modem configuration is used with Serial MUX 'enabled', the modem can be queried for operational information simultaneously with the PPP/IP data traffic. This feature can be used for AT commands (reading the modem's signal strength, etc.) and obtaining GPS location.

These modem options require one or more child objects to be configured for the Cell Modem, which are Virtual COM Ports dedicated specifically to the modem. It is recommended to enable the Serial MUX and to define at least Port 73 for AT command access, and the other ports if needed. The ports are identified as follows:

Port	Description
73	AT commands, used by the RediGate MMI to query the modem
75	(for DE910 CDMA/EVDO modem) – AT commands only. Port 75 may be selected in another serial configuration, such as Terminal Server (for HE910 GSM/HSPA modem) – AT commands or GPS NMEA data. Port 75 may be selected in the FieldUnitNMEA object or another serial configuration.
77	(for DE910 CDMA/EVDO modem) – AT commands or GPS NMEA data. Port 77 may be selected in the FieldUnitNMEA object or another serial configuration. (for HE910 GSM/HSPA modem) – *unavailable*

For additional information on the FieldUnitNMEA object, see the section NMEA (GPS) Field Unit. The ports 73, 75, and 77 are defined under the Cell Modem object, and their paired virtual serial ports are internally generated. You should not create AsyncPort objects with instance numbers 72 through 77 under the Networks placeholder.

Attributes	Function
Object Type	Port73b_AT-CMDs, Port75b_AT-CMDs_GPS-HE910, Port77b_GPS-DE910
Parent(s)	System Networks Cell Modem
Instance	Instance number for each port must be 0. *The ACE template is built so that each of these objects creates the appropriate AsyncPort filename: port073, port075, port077*

Properties	Values (Port 73)
	n/a (only use for AT command access)

Properties	Values (Port 75)
Port Settings	Select the AT command or GPS option for this port: - ***AT commands (disable GPS – HE910)*** *– Only use the port for AT commands on the HE910 or DE910 modem. If used with HE910, disable power to the GPS receiver in the modem.* - ***AT commands (power GPS – HE910)*** – Only use the port for AT commands on the HE910 or DE910 modem. If used with HE910, enable power to the GPS receiver. For instance, this might be used to query GPS location using AT commands. *The following selections enable the GPS receiver to automatically output location data in NMEA format once/second (Port 75 only supports GPS on the HE910 modem). NOTE: a GPS antenna connection is only available on the RediGate 400, not the RediGate 100 series. For a description of the NMEA data messages, see theTelit MT GNSS Software User Guide.* - ***GPS (HE910) - All GPS Sentences/Clock****–Output all NMEA commands listed below, can be used to set the system's real-time clock.* - ***GPS - GGA Only (Lat,Long,Sats,Alt,DOP)****–Output only 'GGA' message (14 comma-separated values).* - ***GPS - GLL Only (Lat,Long)****–Output only 'GLL' message (6 comma-delimited values).* - ***GPS - GSA Only (Sats,DOP)****–Output only 'GSA' message (17 comma-separated values).* - ***GPS - GSV Only (Sats, 1-4 sentences)****–Output only 'GSV' messages (19 comma-separated values per message, up to 4 messages depending on number of satellites in view).* - ***GPS - RMC only/Clock (Lat,Long,Speed)****–Output only 'RMC' message, can be used to set the system's real-time clock (12 comma-separated values).* - ***GPS - VTG Only (True Tracking,Speed)****–Output only 'VTG' message (9 comma-separated values).* - ***GPS - RMC and VTG/Clock****–Output only 'RMC' and 'VTG' messages, can be used to set the system's real-time clock.* - ***GPS - GGA, RMC, and VTG/Clock****–Output only 'GGA', 'RMC', and 'VTG' messages, can be used to set the system's real-time clock.*

Properties	Values (Port 77)
	Same as Port Settings for Port 75, except that it is only applicable to the DE910 modem and should not be used with the HE910.

USAGE NOTE:

There are two ways to get receive GPS data from the modem into RTDB registers and/or use the GPS date/time to synchronize the RediGate system clock:

1) AT commands This method uses a simpler configuration, but data is obtained less frequently (multiple 10's of seconds, depending on the number of AT commands defined).
Select ***AT Commands (Power GPS)*** as the setting for Port 75 or 77, and in the AT Commands object define the $GPSACP command (see the section AT Commands).

2) Real-time NMEA data This method requires a more complicated configuration, but GPS data can be obtained frequently (within a few seconds). More frequent data acquisition from the modem will also potentially impact the bandwidth available to PPP network traffic.
Select one of the ***GPS*** port settings for Port 75 or 77. In addition, you will need to define the FieldUnitNMEA (with AsyncCircuit pointing to port 75 or 77) including a Poll Table to define which command(s) to parse into RTDB registers, include the registers in an RTDB, add one or more scan entries in the Master Channel to set the frequency of GPS data storage, and additionally configure the NMEA_SPY object to capture the unsolicited NMEA data into internal memory buffers.

AT Commands

When the Cell Modem configuration is used with Serial MUX 'enabled', the modem can be queried for operational information simultaneously with the PPP/IP data traffic. The AT Commands object allows one or more AT commands to be defined that will regularly query the cellular modem. This may be used, for instance, to query cellular signal strength, registration status, etc., and store the information into RTDB registers that can be published with MQTT or shared via a SCADA protocol.

The RediGate regularly sends an AT command to read cellular signal strength in order to control the cellular LED. If any user-configured commands are included in the AT Commands object, those commands will be sent alternately with the built-in signal strength query. AT commands are sent at a regular interval of 5 seconds. For instance, if two user commands are defined to read signal strength and registration status into RTDB registers, the AT command sequence will be:

AT+CSQ(built-in)
(5 seconds)
AT+CSQ(user AT command)
(5 seconds)
AT+CSQ(built-in)
(5 seconds)
AT+CREG?(user AT command)
(5 seconds)

When the response to each user-configured command is received, it is parsed according to certain rules, as described below under the 'Conversion' type field. Often, commands will return a comma-separated list of values. The AT Commands object allows these values to be parsed based on comma.

Attributes	Function
Object Type	AT_Commands
Parent(s)	System Networks Cell Modem
Instance	Always 0

Properties	Values (Port 77)
PropertiesValues Timeout Msec	Enter the timeout (in milliseconds) to wait for modem response to AT command.
AT Cmds	This table defines any user-configured AT commands to be queried regularly
RTDB Map	Enter one or more rows in the AT Cmds table to use this feature
AT Command	Enter the AT command string to send to the modem, or a single uppercase character 'C'. The AT string must be a command that is recognized by the modem model being used. If the command returns several different values to be parsed, the 'C' indicates a continuation row. This allows the response from a previous command to be skipped or parsed according to different rules, as described in the remaining properties, below.

Conversion	Select the type of conversion to use when parsing the command response from the modem. • ***SINT16*** – Store value(s) as 16-bit signed integer • ***SINT32*** – Store value(s) as 32-bit signed integer • ***SINT32*** – Store value(s) as 64-bit signed integer • ***REAL32*** – Store value(s) as 32-bit floating point • ***STRING-32*** – Store parsed parameter(s) as a 32-character string. The Count refers to the number of comma-separated strings. • ***STRING-256*** – Store the entire remainder of the AT command response into a STRING-256 register. The Count field is ignored. • ***SKIP*** – Discard one or more comma-separated parameters from the AT command response, based on Count. Use the following GPS conversion options with the "AT$GPSACP" command, which returns GPS data from the modem in the format (the Count column is ignored): `$GPSACP: 214127.000,3853.5898N,09447.4488W,0.9,315.4,3,0.0,0.0,0.0,310715,07` • ***GPS REAL32*** – Store each comma-delimited parameter of the $GPSACP command into thirteen REAL32 registers verbatim, as: 1. UTC time as hhmmss.sss (e.g. 214127.000=9:41:27 PM) 2. Latitude as DDMM.mmmm (e.g. 3853.5898) 3. Latitude direction, N=78, S=83 4. Longitude as DDDMM.mmmm (e.g. 09447.4488) 5. Longitude direction, W=87, E=69 6. HDOP/Horizontal dilution of precision (e.g. 0.9) 7. Altitude, meters above mean sea level (e.g. 315.4) 8. Fix, 0=No fix, 2=2D fix, 3=3D fix 9. Course over ground, as degrees (ddd.mm) 10. Speed over ground (Km/hr) 11. Speed over ground (knots) 12. Date of Fix, as ddmmyy (e.g. 310715=July 31, 2015) 13. Total number of satellites in use (0 to 12) • ***GPS Set Clock*** – Use the time and date returned in the $GPSACP command to set the real-time clock of the RediGate. • ***GPS DDMM.mm to De.gree*** – Store each comma-delimited parameter of the $GPSACP command into thirteen REAL32 registers (ignore Count). The latitude/longitude values are converted from their normal degree.minute(DDMM.mm) format into degrees. Values are the same as above, except Latitude and Longitude: 2. Latitude as ±dddd.dddd (positive=north, negative=south) 4. Longitude as ±dddd.dddd (positive=east, negative=west) • ***GPS Set Clock, to De.gree*** – This option combines the previous two options: convert degree/minute/second to degrees and set the real-time clock.
Channel	Enter the Master Channel number of the destination RTDB.
RTU	Enter the Field Unit address of the destination RTDB.
RTDB Dest	Enter the starting numeric register address of the destination RTDB into which data from this command will be stored. The RTDB addresses must be defined and must be of the correct data type.
Count	Enter the number of data entities of the same 'Conversion' type to parse sequentially. If the response to an AT command includes multiple values of different types, these must be handled on separate rows in the table, with the Count appropriate for each row.
Comment	Optional column, allowing a descriptive comment to be entered for each row in the table. The Comment field is unused in the configuration.

Firewall

The Firewall object provides a means of configuring the 'iptables' settings in the Linux operating system. This includes features such as allowing or blocking access to IP ports or interfaces, port forwarding, Network Address Translation (NAT), and Masquerading a network through another interface. This is an advanced option and may require some additional knowledge of 'iptables. Please consult with a network administrator for advice on the details of configuring this security option, or look for online documentation of 'iptables' such as:

http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:*Ch14*:_Linux_Firewalls_Using_iptables.

The 'iptables' utility manages tables of rules, including 'filter', 'nat', and 'mangle' tables. Each table has one or more 'chains' – for example, the 'filter' table can have INPUT and OUTPUT chains. Each chain will have one or more rules defining how packets are handled for the chain. The 'nat' table uses PREROUTING and POSTROUTING chains. The Firewall configuration properties are used to build a 'firewall.sh' script that runs on startup, which contains a series of 'iptables' commands to set the firewall rules.

Attributes	Function
Object Type	Firewall
Parent(s)	System Networks
Instance	Must be 0.

The **Comment** column used in various tables allows a descriptive comment to be entered for each row in the table. The Comment field is unused in the configuration.

Properties	Values (Port 77)
INPUT Policy **OUPUT Policy** **FORWARD Policy**	Select an INPUT packet policy from one of the following options: Accept All Input/Output/Forwarding Packets Drop All Input/Output/Forwarding Packets *The first actions in the firewall.sh script flush the existing contents of 'iptables' chains, using the commands:* iptables -F INPUT iptables -F OUTPUT iptables -F FORWARD iptables -t nat -F *Then the INPUT Policy, OUTPUT Policy, and FORWARD Policy rules configure the default rules for packets not explicitly defined in the remainder of the configuration. These define commands such as:* iptables -P INPUT DROP iptables -P OUTPUT ACCEPT iptables -P FORWARD DROP All the remainder of the properties include optional tables that may include 0 or more rows with 'iptables' rules to be added to the firewall.sh script
Accept All INPUT by Interface	Enter Linux interface name(s) for which to accept all INPUT packets. This setting overrides a global Drop or Reject rule in the INPUT Policy, and defines commands such as: iptables -A INPUT -i eth0 -j ACCEPT The following rules are included by default: iptables -A INPUT -i lo -j ACCEPT iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
Port Management	The Port Management property allows individual ports to be accepted, dropped, or rejected (with ICMP error), regardless of the above settings. Ports can be specified using the INPUT or OUTPUT chain, protocol (TCP, UDP, or ICMP), Linux interface name, and port number. Some examples of commands are: iptables -A INPUT -p tcp -i eth0 --dport 22 -j ACCEPT iptables -A OUTPUT -p udp -o eth0 --dport 500 -j ACCEPT

Masquerade

The Masquerade property allows devices on one interface to appear as if they existed on a different interface. This is often used, for instance, where devices on a local Ethernet interface need to make outbound IP connections using a public cellular/PPP interface. The local interface is "masqueraded" to the public network side of the interface.

Enter one or more rows in the Masquerade table to use this feature:

Output Interface – Select the Linux network interface name, which is the network on which devices should be made to appear.

Source Network – Enter the IP address range of addresses on one of the other network interfaces which should be allowed to masquerade on the other interface. IP address range should be entered in a format of "*IP_network/mask_bits*", such as: "192.168.1.0/24".

Following are examples of a Masquerade command. In these examples, devices on the 192.168.1.x network are masquerated to the 'eth2' interface, and addresses 172.1.1.5-6 appear on the 'ppp0' interface:

```
iptables -t nat -A POSTROUTING -o eth2 --source 192.168.1.0/24
-j MASQUERADE
iptables -t nat -A POSTROUTING -o ppp0 --source 172.1.1.5/30 -j
MASQUERADE
```

When using masquerading, the following rule is added by default to enable packet forwarding between interfaces:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Forwarding by Interface

The Forwarding by Interface option allows all packets to be freely forwarded between two Linux interfaces, which are selected from a drop-down list. There should always be two rows defined, which will forward packets in both directions. Some examples of 'iptables' commands generated by this option are:

```
iptables -A FORWARD -o eth0 -i ppp0
iptables -A FORWARD -o ppp0 -i eth0
```

DNAT Pre-routing

The DNAT Pre-routing option allows IP packets to be modified as they arrive at an input interface. By checking the packet's "destination port", the packet can be modified by being assigned a new TCP/IP destination address and port number.

Enter one or more rows in the DNAT Pre-routing table:

Interface Name – Select the Linux interface name on which the IP packets will be arriving.

Protocol – Select the protocol of packets to be routed (TCP, UDP, or ICMP).

Dest Port – Enter the numeric IP port number of the incoming packets to be listening for.

New IP AndOr Port – Enter the new IP address and optional port number. This should be entered as "*IP_address:port*", such as "10.10.10.2:161" (this field is limited to 20 characters). Some examples of 'iptables' commands generated by this option are:

```
iptables -t nat -A PREROUTING -i ppp0 -p tcp --dport 8080 -j
DNAT --to-destination 10.10.10.2:80
```

SNAT Post-routing	The SNAT Post-Routing option allows IP packets to be modified before they leave an output interface. By checking the packet's source address and destination port, the packet can then be modified by assigning a new TCP/IP source address and destination port number.
	Enter one or more rows in the SNAT Post-routing table:
	Interface Name – Select the Linux interface name on which the IP packets will be arriving.
	Protocol – Select the protocol of packets to be routed (TCP, UDP, or ICMP).
	Source IP – Enter the IP address of the outgoing packets to be modified.
	Dest Port – Enter the numeric destination IP port number of the incoming packets to be modified. Use only a colon instead of a number to exclude the port setting from the 'iptables' command.
	New IP AndOr Port – Enter the new IP address and port number. This should be entered as "*IP_address:port*", such as "10.10.10.2:161" (this field is limited to 20 characters). Some examples of 'iptables' commands generated by this option are:
	`iptables -t nat -A POSTROUTING -o ppp0 -p udp -s 10.10.10.2`
	`--dport 161 -j SNAT --to-source 192.168.55.22:1661`

Drop All INPUT by Interface	This property allows for any other INPUT packets that were not caught in previous 'iptables' rules on a given interface to be dropped. Select the Interface Name to drop packets. An example of this rule is: `iptables -A INPUT -i ppp0 -j DROP`

Custom IPTABLES	Finally, the Custom IPTABLES option allows you to configure any other 'iptables' commands that the previous Firewall object properties didn't support. The 'iptables' utility has many options and variations that might be needed for certain networking situations. These custom rules are added to the firewall.sh script verbatim, with one qualifier:
	The free format table entry only allows a maximum of 80 characters per line. If the command requires more than 80 characters, use a tilde (~) character at the end of a line to indicate that the next line contains a continuation of the command. The tilde character will be converted to a backslash (\) character in the script to perform the continuation. Here are two recommended examples of custom entries allowing incoming and outgoing 'ping' traffic:
	`iptables -A INPUT -p icmp -m state –state ~`
	`NEW,ESTABLISHED,RELATED -j ACCEPT`
	`iptables -A OUTPUT -p icmp -m state –state ~`
	`NEW,ESTABLISHED,RELATED -j ACCEPT`

Routes

The Routes configuration defines IP route information that is used for specifying a Default Gateway and other route entries. Serial IP networks (PPP, SLIP) require this because their ACE objects do not include a Default Gateway option in their parameters. Route entries to specific addresses or subnets are occasionally used for more advanced networking options.

Attributes	Function
Object Type	Routes
Parent(s)	System Networks EtherPort

Instance	Must be 0.

Properties	Values
Route Table	Click the **Edit Table** button to edit the list of Multi-Home addresses.
Destination Address	Enter an IP address in the range of addresses defined in this route entry, typically the first one in the range. When defining a Default Gateway, it must appear in the first row and have the Destination Address and Net Mask set to 0.0.0.0. Also make sure that no other Default Gateway is used for other interfaces in the configuration, including those obtained through DHCP.
Net Mask	The Subnet Mask defines the range of addresses to be defined by this route entry. If defining the Default Gateway (first row of table only), this must be set to 0.0.0.0.
Gateway	Enter the IP address to use as the Default Gateway for addresses defined in this route entry. If the first row in the Routes table is defined as a Gateway of 0.0.0.0, it is treated as the Default Gateway for the system (overriding a Default Gateway setting in Ethernet objects). Set the Gateway to an address other than 0.0.0.0 to define a specific route definition. OR, you can define a route based on the interface rather than a specific IP address. To do this, set the Metric to one of several specially designated values (90, 91, 100, 101, etc.), as described below. In this case, the Gateway property may be set to 0.0.0.0 to omit the 'gw' field in the Linux route command. Note that when defining the Gateway property (other than 0.0.0.0), the address of the gateway must be reachable via the networking defined in other ACE objects for the specified interface.
Interface	Enter the text identifier of the network interface to use for the addresses appearing in this route. Note: This is case-sensitive. For instance, if the route entry specifies an address range on the Ethernet network, and the Ethernet object is configured with "Ether1" for its Domain Name, then "Ether1" *must be entered as the Interface here.*
Metric	The Metric indicates the relative priority when two routes might be used to reach the same network address. The Metric with the lower number will be given priority. OR, use the following specially designated values in the Metric field to set up a static route based on interface name rather than IP address: • Use Metric of 90 to use the 'ppp0' interface (91=ppp1, 92=ppp2, etc.) • Use Metric of 100 to use the 'eth0' interface (101=eth1, 102=eth2, etc.) • With these designations, the Linux interface name is used in the 'route' entry instead of IP addresses.
Comment	Optional column, allowing a descriptive comment to be entered for each row in the table. The Comment field is unused in the configuration.

TLS Tunnels

The TLS Tunnels object is used to configure TLS/SSL encryption, using the Linux 'stunnel' security agent. TLS tunnels may be used to wrap an otherwise unsecure communication channel on a single TCP/IP socket inside an authenticated, encrypted protocol to protect the network devices and data being transmitted. You may need to consult with a network administrator for advice on the details of configuring TLS/SSL encryption, or consult Linux documentation for more information on 'stunnel.'

(Note: earlier ACE configurations used an SSL_Tunnels object, which included a free form text table for many of the 'stunnel' properties. The TLS Tunnels object is equivalent, but provides individual settings. Only one or the other object may be used in a configuration at a time.)

Attributes	Function
Object Type	TLS Tunnels
Parent(s)	System Networks
Instance	Must be 0.

The following parameters are used to create the stunnel configuration file, located at /etc/stunnel/stunnel.conf.

Properties	Values
TLS Version	Select the version of TLS or SSL protocol to use. *TLS protocol versions are more secure than SSL. Select "all" to allow the client and server to negotiate the protocol.*
Compression	Selet the type of data compression to use. *Select 'none', 'zlib', or 'rle'.*
Verify Certificate	Select whether (and how) to use certificate verification for authentication to an TLS/SSL server. A security certificate is optional for a client but required on a server. *The number after the option indicates the "verify=" stunnel value:* • *NO certificate verification* • *ALWAYS require peer cert (2)* • *Request and ignore peer cert (0)* • *Validate only if cert is present (1)* • *Verify peer with locally installed cert (3)* • *Ignore CA chain & only verify peer cert (4)*
Certificate File	If the Verify Certificate option has been selected to employ certificate authentication, identify the location on the Linux file system containing the certificate chain PEM file. If used, this property must begin with "cert = ". **NOTE:** If no certificate is to be used, this field must be disabled, either by adding a semicolon at the beginning ("**;** cert = ") or by clearing the property value entirely. Otherwise, the TLS/SSL connection will fail. *The certificate file must be obtained from an appropriate certificate authority containing credentials for this device, which are also known by the TLS/SSL server. The certificate file must be put on the device in the specified location, and must be in PEM format.*
Key File	If the Verify Certificate option has been selected to employ certificate authentication, identify the location on the Linux file system containing the private key assocated with the certificate. If used, this property must begin with "key = ". **NOTE:** If no certificate is to be used, this field must be disabled, either by adding a semicolon at the beginning ("**;** key = ") or by clearing the property value entirely. Otherwise, the TLS/SSL connection will fail. *The key file is typically created along with the certificate and must be put on the device in the specified location, and must be in PEM format.*
CA File	If the TLS/SSL server's certificate must be validated with a Certificate Authority before connecting to it, a file identifying the CA must be stored on the Linux file system. If used, this property must begin with "CAfile = " (case-sensitive). *The CA file must be in PEM format.*
CRL Path	If using a Certificate Revocation List file(s) to confirm the validity of the server's certificate, this property is used to identify the directory on the Linux file system where the CRL file(s) will be stored. *Only two options are available:* • *none* • */etc/stunnel/crls* *If using CRL files, they must be stored in the above directory in PEM format.*
Connect Timeout	Select the amount of time to wait for a TLS/SSL connection to be established. *Default selection is 10 seconds.*
Idle Timeout	Select the amount of time to keep an idle connection open when there is no data transmitted. *Default selection is 1 hour.*
Busy Timeout	Select the amount of time to wait for expected data in case of a busy network. *Default selection is 5 minutes.*
FIPS mode	Select whether to use FIPS 140-2 encryption mode. *Default is no. (FIPS mode is not currently supported.)*

Cipher List	Enter a list of encryption ciphers to allow for the TLS/SSL connection. This property must begin with "ciphers = " and must contain some criteria for the list of ciphers to include or exclude. Use a colon (:) to separate cipher names or criteria. (This property is not required and may be disabled by adding a semicolon before "ciphers" or by clearing the property entirely.) *Example: ciphers = !SSLv3:DH+AES:ECDH:-AES128* *In Linux, the ciphers available in the system may be listed using the command:* `openssl ciphers -v` *or (for example):* `openssl ciphers -v '!SSLv3:DH+AES:ECDH:-AES128'` *The openssl command lists ciphers of various strengths, including those used by SSL or TLS protocol versions. In order to ensure more robust encryption, the list may be filtered to allow only more secure ciphers.* *In the above example, "!SSLv3" excludes all ciphers used with the older SSLv3. "DH+AES" includes ciphers that use DH or AES, but excludes those using RSA. "ECDH" includes protocols that use ECDH. "-AES128" filters the list of whatever ciphers may have been included in the previous list by excluding those which use AES with 128-bit encryption, but allows those with 256-bit or better.* *Consult 'openssl' documentation for further information.*
Renegotiation	Select whether to support connection renegotiation.
Delay DNS	Select whether to delay DNS lookup until connection.
Debug Level	Select the debugging level for TLS/SSL diagnostics. *The default level is 5 (notice). Use level 7 for a greater quantity of diagnostic messages in the Log File to troubleshoot connection problems.*
Log File	This property is hard-coded and indicates where the TLS/SSL debug messages may be found. *Only option is /var/log/messages*
Socket option 1	Sets TCP socket options for the connection. This is an optional field, but if used for socket options it must begin with "socket = ". See stunnel documentation for further information. *Default value is "socket = l:TCP_NODELAY=1"*
Socket option 2	Sets TCP socket options for the connection. This is an optional field, but if used for socket options it must begin with "socket = ". See stunnel documentation for further information. *Default value is "socket = r:TCP_NODELAY=1"*
PID	Name of PID file used by Linux for the TLS/SSL process. *This option is hard-coded to /var/run/stunnel.pid*
Param 1 **Param 2** **Param 3**	Additional (optional) stunnel parameters. *If used, these fields must be formatted as proper 'stunnel' configuration options and will be placed verbatim in the stunnel.conf Linux configuration file.*
Client Mode	Choose whether to use client mode for the TLS/SSL connection. In Client Mode, this will listen for a local (non-secure) connection to be made to its listener port, and then initiate a connection to a remote server. If set to Server Mode, this will operate as a TLS/SSL server, waiting for a connection to be made to it from another secure client.
STUNNEL Parameters	In the STUNNEL Parameters field, enter a series of properties that are used to define one or more TLS/SSL tunnel between a non-secure and a secure port connection.
Tunnel Name	Enter a unique logical name of the stunnel service (limited to 16 characters) for each tunnel being defined.
Accept Connection	Enter a string that defines the port which will receive the connection, and an optional IP address. Some examples of port or "IP:port" are given below: • *443* • *127.0.0.2:1883* • *192.168.1.2:3040*
Connect To	Enter a string that defines the address and IP port to which a connection will be made after receiving a socket on the "Accept Connection" port. The address being connected to must be accessible using the system's DNS and routing rules. Some examples are: • *10.1.2.1:443* • *xyz.com:20000* • *127.0.0.3:3040*

Network Monitor

The Network Monitor icon is a placeholder in the ACE configuration, under which individual NetMon objects are defined to monitor system or network conditions.

Attributes	Function
Object Type	NetworkMonitor
Parent(s)	System Networks
Instance	Must be 0

NetMon (Network Monitor instance)

The NetMon icon defines a Network Monitor process, allowing the RediGate to detect certain conditions in the system or networking, such as: ping success/failure, RTDB register value, network port or interface status, etc. If the measured 'condition' value matches a certain criteria, an action is performed in response, such as: send pings, switch redundant path, write to an RTDB register, restart networking, or run a script.

Each NetMon instance performs its condition checking and actions independently from all other instances. The same Monitor or Action Register may be used by more than one NetMon instance to store similar information, but realize that each NetMon instance will overwrite the value stored by other instances.

Attributes	Function
Object Type	NetMon
Parent(s)	System Networks NetworkMonitor
Instance	Enter a unique instance number between 0 and 99.

Properties	Values
MONITOR Interval	Enter period (in seconds) for how often to check the system condition.

Condition	Select which network condition to monitor. For most conditions, the actual measured value is checked against VALUE property, using the comparison type specified in Criteria. A resulting action will be triggered if the Criteria is satisfied. • **No Criteria** – Always trigger on MONITOR Interval (timed action). • **PING FAIL** – Send one ping at a time (to one or more network addresses) and check the failure count. Trigger only occurs if pings to ALL addresses fail a number of sequential times as compared with VALUE. If "Interface or Register" property is set to a Linux interface (such as "eth0" or "ppp0"), that interface will be enforced for pings. Otherwise, leave Interface blank to send ping according to network routing rules. • **PING FAIL ANY** – Send one ping (to one or more network addresses) and check the failure count. Trigger occurs if pings to ANY address fails a number of sequential times as compared with VALUE. If "Interface or Register" property is set to a Linux interface (such as "eth0" or "ppp0"), that interface will be enforced for pings. Otherwise, leave Interface blank to send ping according to network routing rules. • **PING GOOD ANY** – Send one ping (to one or more network addresses) and check the success count. Trigger occurs if pings to ANY address succeeds a number of sequential times as compared with VALUE. If "Interface or Register" property is set to a Linux interface (such as "eth0" or "ppp0"), that interface will be enforced for pings. Otherwise, leave Interface blank to send ping according to network routing rules. • **READ REGISTER value** – Read the value specified in Channel/RTU and compare with VALUE. The "Interface or Register" property must be set to the RTDB numeric register (e.g. 40001) or the register's Tag name. • **RX PACKET COUNT on Interface** – Compare VALUE with the Linux network specified in the "Interface or Register" property (such as "eth0", "ppp0", etc.) for the total "RX packets" count in 'ifconfig'. • **RX PACKET ERROR on Interface** – Compare VALUE with the Linux network specified in the "Interface or Register" property (such as "eth0", "ppp0", etc.) for the total RX packets "error" count in 'ifconfig'. • **# of STATIC ROUTES on Interface or all** – Compare VALUE with the number of 'route' entries. If the "Interface or Register property specifies a Linux interface (such as "eth0", "ppp0", etc.), only count those. If Interface is left blank, then count all route entries on all interfaces. • **# of ESTABLISHED port connections** – Compare VALUE with the number of entries in 'netstat' which have "ESTABLISHED" TCP connections. Set the "Interface or Register" property to be a specific numeric port number to check for ESTABLISHED connections, or leave it blank to check for all ports. • **# of FAILED PASSWORD login attempts** – Compare VALUE with the number of "Failed password" entries in /var/log/auth.log. Login count resets on reboot or if a large auth.log file is reset by the log file management. • **# of ACCEPTED PASSWORD logins** – Compare VALUE with the number of "Accepted password" entries in /var/log/auth.log. Login count resets on reboot or if a large auth.log file is reset by the log file management.
Criteria	Select the criteria to use for detecting a trigger condition that will result in an Action. The measured value obtained from the Condition, above, is compared with the VALUE property of this NetMon instance. Criteria may be: • Measured value is "**Greater than or equal to**" the VALUE property (or "**Greater than**", "**Less than or equal to**", "**Less than**", "**Equal to**", or "**Not equal to**" VALUE) • For the following options (Changed, Increased, Decreased), VALUE must be a positive integer. The measured value is compared with the value obtained the last time the action was triggered. On startup, the "last value" is set to current value the first time this NetMon instance runs. • **Changed Value** – The action will occur if the measured value changes (increase or decrease) more than the amount specified in the VALUE property. (NOTE that a number that wraps around positively or negatively will count as a change and cause a trigger.) • **Increased Value** – The action will occur if the measured value increases more than the amount specified in the VALUE property. A value that stays the same or decreases will NOT cause a trigger. (NOTE that a decreasing integer that wraps from 0 to a large maximum value will be counted as an increase. However, for an increasing integer that wraps around to a small number, the "last value" will be taken as 0 for comparison.) • **NOT Increased Value** – The action will occur if the measured value does not increase more than the amount specified in the VALUE property. This can detect a value which should normally increase (such as network packet count or a PLC heartbeat) but stops incrementing for some reason. A value that stays the same or decreases will cause a trigger. (NOTE that a decreasing integer that wraps from 0 to a large value will be counted as an increase. A large increasing integer that wraps to a small number will cause a trigger.) • **Decreased Value** – The action will occur if the measured value decreases more than the amount specified in the VALUE property. A value that stays the same or increases will NOT cause a trigger. (NOTE that an increasing integer that wraps from a large maximum value to 0 will be counted as a decrease. A decreasing integer that wraps from 0 to a large value will not be counted as a decrease.)
VALUE	Signed integer value (-2,000,000,000 to 2,000,000,000) used for comparison with measured system Condition value according to Criteria.
Channel	Master Channel number (0-15) used for READ REGISTER condition. Unused for other options.
RTU	Field Unit number used for READ REGISTER condition. Unused for other options.

Interface or Register	• For Condition options using a network interface (RX PACKET or ERROR count, STATIC ROUTES), this property should be set to the Linux interface name (such as "eth0" or "ppp0"). For STATIC ROUTES, leave it blank to count all route entries on all interfaces. • For the READ REGISTER condition, this property should be set to the numeric register address in the RTDB (e.g. 40001) or the register's Tag name (up to 127 characters). • For the ESTABLISHED Ports condition, this property can be set to a specific numeric TCP port, or leave it blank to check all ports. • Otherwise, this field is ignored.
Ping Addresses	The Ping Addresses table should contain a list of one or more network addresses (numeric IPv4 address or named server) to use for the PING GOOD or PING FAIL condition. It is ignored for all other monitoring conditions. When using named server addresses, make sure DNS is used to resolve network names.
Redundant Path	The Network Monitor may be used to control routing for installations including a Primary/Secondary network path that require static routes or default gateway to be changed dynamically. For instance, a RediGate may have primary path on satellite and secondary path over cellular. The system might be set up with one NetMon instance sending a ping over satellite (only while on the primary path), which if it fails will switch routing to the cellular network. Another NetMon instance could run (only while on the secondary path) to send pings over the satellite to detect when the primary path becomes available again. The Redundant Path property is used as an additional qualification to allow the Condition checking for this NetMon instance only when the network is on either the primary or secondary path. • **N/A** – Set this to "N/A" if not using this NetMon instance for redundant path control. • **Action on Path 0 ONLY** – Only check the Condition when the RediGate is on Path 0. The RediGate is assumed to be on Path 0 at startup and after running the Action "Switch to PATH 0". • **Action on Path 1 ONLY** – Only check the Condition when the RediGate is on Path 1. The RediGate is assumed to be on Path 1 after running the Action "Switch to PATH 1".
ACTION Taken	If the NetMon instance verifies that a system or network Condition value meets the specified Criteria, an Action will be taken. Select an action from the following options: • **None** – This option only results in the Monitor Register (below) being updated with the current system value used in the Condition checking, but otherwise no action is taken if the Criteria is satisfied. • **SEND PINGS** – If Criteria is satisfied, send one or more pings to a network address. The number of pings to send is configured in the Ping Count property. The destination (IPv4 address or named server) must be configured in the Action Text property (which may be prefixed with ping options `-I`, `-s`, and/or `-W`). • **Switch to PATH 0** – Run the Linux command (or script) configured in the Action Text property, which will be considered as a change from Path 1 to Path 0. For instance, the command or script might add a static route, change default gateway, etc. • **Switch to PATH 1** – Run the Linux command (or script) configured in the Action Text property, which will be considered as a change from Path 0 to Path 1. • **Run SCRIPT** – Run the Linux command (or script) configured in the Action Text property. This will not be considered to indicate a change between the redundant Path 0/Path 1 states. • **REGISTER WRITE** – Write a value into an RTDB register location. The value to write is configured in the Action Text property (the value to write should be of an appropriate data type for the destination register type). The value is written to the database address specified by the NOTIFY Channel, NOTIFY Rtu, and Action Register properties. • **Restart ETHERNET Ports** – Restart networking on all Ethernet ports with the Linux command: `S40network restart` • **Restart CELLULAR Ports** – Reset cellular networking with the Linux command: `S15cellmodem restart`. This only applies when using an Elecsys internal modem. • **Restart CELL MODEM** – Power cycle and completely restart cellular modem networking with the Linux command: `S11emux restart` (on RediGate 100) or: `S03emux restart` (on RediGate 400). This only applies when using an Elecsys internal modem. • **RECONFIGURE** – Issue 'reconfigure' command to restart RediGate operation and/or install newly loaded configuration. Reconfigure will be delayed 30 seconds after the Action is triggered. • **REBOOT Linux** – Shutdown and restart the RediGate. Reboot will be delayed 30 seconds after the Action is triggered.
Ping Count	Number of pings to send to destination address (only used for SEND PINGS action).

Action Text	Text field containing properties used for several Action types (up to 255 characters). • For **SEND PINGS** action, this property must contain the IP address or named server to ping. When using named server addresses, make sure DNS is used to resolve network names. Destination address can be prefixed with one or more ping options **<u>before</u>** the address, including: `-I iface` (uppercase '`I`') specify ping on Linux interface 'iface' (e.g., "eth0") `-s size` (lowercase 's') send 'size' data bytes in each packet (default 56) `-W sec` (uppercase 'W') timeout of 'sec' seconds to wait for ping response (default 10) For instance, if the desired Action is to ping the address 'www.google.com' 5 times, specifically over interface eth0, with a ping timeout of 15 seconds, the Ping Count would be 5, and the Action Text would be: `-I eth0 -W 15 www.google.com` • For **Run SCRIPT** or **Switch to PATH** actions, this property must contain the Linux command line (script or command, with all parameters) to execute. • For **REGISTER WRITE** action, this property must be set to the value to be written into an RTDB register (integer, floating point number, Boolean 1 or 0, or string). • For all other actions, this property is ignored.
NOTIFY Channel	Master Channel number to store the value of a system Condition whenever it is checked by the NetMon process, <u>and</u> to store the result of an Action.
NOTIFY Rtu	Field Unit address to store the value of a system Condition whenever it is checked by the NetMon process, <u>and</u> to store the result of an Action.
Monitor Register	RTDB register address (or Tag name, up to 100 characters) to store the value of a system Condition whenever it is checked by the NetMon process after each 'MONITOR Interval' (if the current path matches the Redundant Path setting). Values stored for each Condition are: • **No Criteria** – Store '1'. • **PING FAIL** – Store count of failed pings on any address (single counter, resets to 0 if any ping is successful). • **PING GOOD** – Store the <u>largest</u> successful ping count on any address. • **READ REGISTER value** – Store value obtained from source RTDB register. • **RX PACKET COUNT on Interface** – Store RX packet count (should be UINT32). • **RX PACKET ERROR on Interface** – Store RX error packet count (should be UINT32). • **# of STATIC ROUTES on Interface or all** – Store number of static route entries. • **# of ESTABLISHED port connections** – Store number of ESTABLISHED port connections. • **# of FAILED PASSWORD login attempts** – Store number of failed logins. • **# of ACCEPTED PASSWORD logins** – Store number of successful logins.
Action Register	RTDB register address (or Tag name, up to 100 characters) to store the result of an Action whenever it occurs (no change is made unless the Action occurs). Values stored for each Action are: • **None** – Nothing is stored. • **SEND PINGS** – Store number of successful pings resulting from Action (Note, this is different from the pings sent by the 'ping' Condition checking options). • **Switch to PATH 0** – Store '0' ('0' is also stored in /tmp/director/NetMonPath). • **Switch to PATH 1** – Store '1' ('1' is also stored in /tmp/director/NetMonPath). • **Run SCRIPT** – Store the number of times the 'Action' has triggered the script to run. • **REGISTER WRITE** – Store the value in the Action Text property. Note that the Action Register should be defined in the RTDB as the correct type (Strings can't be stored into Integer registers, etc.). • **Restart ETHERNET Ports** – Store the number of times the 'Action' has triggered an Ethernet restart. • **Restart CELLULAR Ports** – Store the number of times the 'Action' has triggered a cellular reset. • **Restart CELL MODEM** – Store the number of times the 'Action' has triggered a hard reboot of the modem. • **RECONFIGURE** – Store '1'. • **REBOOT Linux** – Store '1'.
Debug Level	Set the default Debug Level of this NetMon instance at RediGate startup. Options are the same as in RediGate debugging menu: 0=Fatal Errors only, through 4=Data Dump/verbose output. The additional option 5=Engineering (extra verbose) output generates even more diagnostic information in RediGate diagnostics and at the command line and is not generally intended for customer use. The Debug Level for NetMon processes cannot currently be set within the normal RediGate diagnostics menu. However, a command line process can be used to change the Debug Level during operation. Contact Elecsys for details.

DNS Client

The DNS Client object is used to manually configure DNS entries into the Linux resolv.conf file.

Attributes	Function
Object Type	DNS Client
Parent(s)	System Networks
Instance	Must be 0.

Properties	Values
DNS Server #1-6	Enter up to 6 DNS server addresses to use for resolving named servers, in dotted notation. *DNS addresses should be entered consecutively starting with #1. Any entries after a 0.0.0.0 entry will be ignored.*
Search	(optional) Enter a search string to use in the Linux 'resolv.conf" for the DNS host name lookup

Quagga (RIP routing)

Quagga is a Linux version of network routing software, which includes support for protocols such as RIP (Routing Information Protocol). Along with the RIP-Quagga child object, these ACE objects are used in cases where network routing functions are required to be responsive to an exernal router using the RIP protocol.

Contact Elecsys for advice on the configuration and use of Quagga.

Attributes	Function
Object Type	Quagga, RIP_Quagga
Parent(s)	System Networks
Instance	Must be 0.

VLAN

The VLAN object effectively subdivides an Ethernet port into multiple virtual LAN ports and adds 802.1Q VLAN tagging bytes to the TCP/IP network packet data. This feature must be used in conjunction with an external router or switch supporting VLAN tagging.

For example, a RediGate 100 only has one Ethernet port, but an application requires that it connect through multiple physical ports of a VLAN-aware network switch, where each port's communication needs to be segregated at the link layer from the communication on the other ports.

NOTE: Make sure to define one EtherPort object instance for every physical and virtual LAN device used in the VLAN Table.

Attributes	Function
Object Type	VLAN
Parent(s)	System Networks
Instance	Must be 0.

Properties	Values
VLAN Table	In the VLAN Table field, add a table row for every VLAN to be defined. **Physical Device** – Select the physical LAN device to be divided into VLANs, such as eth0 (corresponding to EtherPort object with instance 0). In Linux, the original network interface will be renamed (e.g., eth0 will be renamed to eth0_base) unless the VLAN_ID is 0. If the interface is renamed to "eth?_base", the IP address settings configured in ACE for that physical device are not used. However, the instance of the physical port still must be defined in order to give Linux a network interface that can be divided into VLANs. **New Device Name** – Select the Virtual LAN device to associate with the Physical Device selected (above). The IP settings for this VLAN device will be taken from the EtherPort object with the corresponding instance number. **VLAN ID** – Enter the numeric VLAN ID to use for 802.1Q tagging (1 to 4094). Use 0 to keep the original interface with untagged packets (i.e. don't rename to eth?_base). It is recommended to avoid VLAN 1.

PPP Port

The PPP port configuration defines the physical PPP (Point-to-Point Protocol) connections. PPP is a serial IP connection that is used for some dial-out or dial-in applications. (For an Elecsys E-Modem, use the Cell Modem object instead of this generic PPP object.)

Attributes	Function
Object Type	PPPport
Parent(s)	System Networks
Instance	Must be 0. This defines the interface as 'ppp0'. The instance number is the next consecutive number, **starting from zero**. Instance #0 is the configuration for the 'ppp0' interface. *There is no correlation between PPP instance number and the physical COM port to which it will be attached.*

Properties	Values
PPP Port	Select the physical communication port to be used for PPP. ***Do not*** *configure this port as an Async or other type of port in addition to the PPP port configuration. If there are Async and PPP objects defined for the same physical COM port, neither will work properly.*
Baud Rate	Select baud rate for the PPP port.
Parity	Select the parity for the PPP port (None, Odd, Even).
Word Length	Select the data bits for the PPP port (7 or 8 bits).
Stop Bits	Select the stop bits for the PPP port (1 or 2 bits).

Warm Up Time	Enter value for warm up time. *This is the amount of time to wait before sending data after the RTS handshaking lead has been asserted.* *An entry of -1 denotes that no handshaking be used. An entry of 0 denotes that RTS/CTS hardware handshaking will be used (no data will be sent until CTS is asserted, and active CD must be present to receive data)). A positive value will transmit data after the configured number of milliseconds, independent of CTS.*
Warm Down Time	Enter value for warm down time. *This is the amount of time to wait after the entire message packet has been shifted out to keep the RTS handshaking lead asserted.* *An entry of -1 denotes that no handshaking be used.*
Domain Name	Enter the domain name. *Name used by certain tasks internally to identify different IP adapters. This is case-sensitive.*
PPP IP	Enter the PPP IP address. *This is the address at which other network devices will see this device when trying to make a connection via PPP. If this device is connecting to a PPP device that can automatically assign an IP address, this parameter may be configured as 0.0.0.0.*
Subnet Mask	Enter the subnet mask. *Should always be 255.255.255.255 for PPP.* *If a static IP is used and a Default Gateway is required to make outbound connections beyond its subnet, the Routes object must also be configured (see the section Route).*
Connection TimeToDie	Number of seconds to operate a PPP session before killing the connection. *This time is absolute, based on the time at which the session was initiated. The PPP connection will be closed regardless if data is still being transferred when the TimeToDie timer expires. This may be used to force a dial connection to hang up to limit telephone connection charges. Disable the TimeToDie by setting it to -1 if the connection is a permanent hard-wired connection, so that it will never be closed.*

PPP PSTN Dialer (PSTN)

The PSTN Dialer configuration defines how the unit will dial out to the public switch telephone network (PSTN) using a dial-up modem. The PSTN object used for PPP is optional, depending on the needs of the system.

Include the PSTN object if:

1. Dial-out on PPP is required (this device initiates the connection); or,
2. A host computer initiates the connection, but the application requires a "Time to Live" that will automatically hang up after a period of inactivity.

Exclude the PSTN object if:

1. The PPP connection is hard-wired rather than using modems; or,
2. Connection is Dial-in only, and no Time to Live setting is required.

Attributes	Function
Object Type	PSTN
Parent(s)	System Networks PPPport
Instance	Must be 0

Properties	Values

Initialize String	Enter text for an AT command to be sent to the modem upon initialization. *Consult modem manual for initialization parameters. Do not include the phone number to dial here. Multiple AT command strings can be sent if separated by "\r". If additional initialization characters are needed, terminate this field with backslash ("\") and continue the string in the Init String 2 field.*
Dial String	Enter the AT string with the phone number to dial (0 to 31 characters). *Spaces and dashes will have no effect. Use a comma to insert a pause of 1 second. Be sure to include 9 for an outside line if necessary, and the full number including 1 for long distance, and area code.*
Init String 2	Continuation of Init String, if last character is \. Otherwise this is ignored.
Prompt 1	Enter text that will be returned by the server for an input prompt. *Often this will be the word "Login", prompting the user to enter a username.* *The string is case sensitive, so it is recommended to leave off the initial "L" since some servers will return "login" and others "Login:" etc.*
Response 1	Enter text to be sent to the server in response to the Prompt 1. *This is case sensitive and should typically be the user name allowed by the server, if Prompt 1 is a login prompt.*
Prompt 2	Enter text that will be returned by the server for a second input prompt. *Often this will be the word "Password", prompting the user to enter a username. The string is case sensitive, so it is recommended to leave off the initial "P".*
Response 2	Enter text to be sent to the server in response to the Prompt 2. *This is case sensitive.*
Prompt 3	Enter text that will be returned by the server for a third input prompt. *The string is case sensitive. Any of the Prompt and Response parameters can be left blank if not required by the dial-in server.*
Response 3	Enter text to be sent to the server in response to the Prompt 3. *This is case sensitive.*
Master Network TimeToLive	Enter the Time to Live (in seconds) for this connection (1 to 86400). *The Time to Live is the amount to keep the session alive without data traffic before closing the connection.* *The TimeToLive allows the connection to be closed after a period of silence. However, the PPP TimeToDie property will force the PPP connection closed automatically regardless of data traffic.*
Connect Retry Count	Enter the number of retry attempts to dial-in to the server.

PPP Authentication (PppAuth)

The PPP Authentication configuration allows the PPP connection to be authenticated by a server that requires PAP or CHAP authentication. If the PPP Server does not require authentication, this object should be omitted from the configuration.

Note: the RediGate does not support MSCHAP authentication. The PppAuth object only allows the RediGate to identify itself to be authenticated at the other end of the connection. To require the RediGate to authenticate external devices, use the PPP Secrets object.

Attributes	Function
Object Type	PppAuth
Parent(s)	System Networks PPPport
Instance	Must be 0

Properties	Values

Authentication Type	Select the type of PPP Authentication required by the PPP network. *Authentication types are:* ***PAP Authentication*** ***CHAP Authentication***
User Name	Enter the user name required by the PPP server PAP or CHAP authentication. *User Name is case sensitive.*
Password	Enter the password required by the PPP server PAP or CHAP authentication. *Password is case sensitive.*
Authentication Tries	Enter the number of times to attempt authentication.
Authentication Timeout	Enter the timeout (in seconds) to wait for confirmation of each authentication attempt.

PPP Secrets

The PPP Secrets object is an optional ACE configuration object, that allows entries to be added to the Linux 'secrets' file. The 'secrets' file is used by the Linux pppd process for authenticating external devices connecting into this RediGate (the PPP Authentication (PppAuth) object authenticates the RediGate in another system). If a system configuration requires a customized entry to be added into the secrets file, it may be added in this object.

Attributes	Function
Object Type	Secrets
Parent(s)	System Networks PPPport PppAuth
Instance	Must be 0

Properties	Values
Secrets_00 through **Secrets_10**	Enter the Linux 'secrets' entry as a text field. through *Each Secrets entry must include four text fields separated by a space.* *The four fields are:* • ***Client name*** • ***Server name*** • ***Authentication secret*** • ***Optional IP address*** *(this may be entered as a range of addresses with asterisks, such as ...)* Search for "PPP Secrets" documentation on the Internet for additional information on the format of the IP address field.

Host Dial Backup

In HCP applications, it is sometimes necessary to define a primary and secondary connection path from the HCP to the RediGate. The Host Dial Backup object tells the HCP which network interfaces to use for primary and secondary networks, and some characteristics of network failover.

The ACE object is not used at all within the RediGate, but is only used by the HCP for starting up its connections. The Host Dial Backup object was originally designed to allow the host system to initiate a secondary dial-up connection to a device when the primary link over a satellite network failed, but the redundant connection is not limited to a dial-up network.

Attributes	Function
Object Type	HostDialBackup
Parent(s)	System Networks
Instance	Must be 0

Properties	Values
Primary Connection Network	Select the network interface through which the HCP should make the primary connection. ***Ethernet 0*** uses the primary IP network address configured in the Ethernet object (instance 0). ***Slip 0*** *and* ***Slip 1*** *options are currently unused.* ***PPP 0*** *uses the IP network address configured in PPP object 0.* ***PPP 1*** *uses the IP network address configured in PPP object 1.*
Secondary Connection Network	Select the network interface to which the HCP should make a secondary connection whenever the primary connection is unavailable. *The same options are selected as for the Primary network.* *Select "No Secondary Connection" if there is only a single IP address/network to which the HCP can connect.*
Time to Fail to Secondary	Enter the time (in seconds) before the HCP should attempt to make connection to the Secondary network address, after losing connection on the Primary network. *This is ignored if no Secondary connection is defined.*
Time to Stay on Secondary	Enter the time (in seconds) before the HCP should attempt to make connection to the Secondary network address, after losing connection on the Primary network. *This is ignored if no Secondary connection is defined.*
Secondary Idle Time	Enter the time (in seconds) after disconnecting from the Secondary network address before reconnecting to the Secondary, if the Primary network is still unavailable. *This option may be used to reduce long distance charges by dialing the Secondary network infrequently during a long outage of the Primary network. For instance, the HCP might connect via dial-up PSTN line once or twice an hour to get critical data updates and then disconnect.*
Startup Auto/Man	Select the default failover behavior for HCP connections. ***Automatic*** *– On startup, the HCP will automatically switch between Primary and Secondary connection paths.* ***Manual*** *– On startup, the HCP will wait for an operator to manually switch from the Primary to the Secondary connection. This is the default setting for the connection upon first starting the HCP. The Auto/Manual setting for each RediGate can be overridden in the HCP user console at any time.*

Secondary Route Test

The Secondary Route Test uses Telnet port 23 and was used on old (pre-Linux) Directors for checking integrity of the secondary route. For a more secure method of secondary route testing, use the Secondary Slave Test in the following section instead.

Secondary Slave Test

When the RediGate is used with an HCP with Primary and Secondary routes configured, the Secondary Route Test performs a basic check of the TCP/IP network communication capability. The Secondary Slave Test object allows the HCP to also test reading and/or writing of a Modbus slave device on the RediGate to verify that protocol data can be exchanged over the secondary route.

Attributes	Function
Object Type	SecondarySlaveTest
Parent(s)	System Networks HostDialBackup
Instance	Must be 0

Properties	Values
Primary Connection Network	Select the network interface through which the HCP should make the primary connection. ***Ethernet 0*** *uses the primary IP network address configured in the Ethernet object (instance 0).* ***Slip 0*** *and* ***Slip 1*** *options are currently unused.* ***PPP 0*** *uses the IP network address configured in PPP object 0.* ***PPP 1*** *uses the IP network address configured in PPP object 1.*
Connect Port	Enter the IP port of the Modbus slave on this unit to use for Modbus communication. This feature requires that a network Modbus slave be configured on the RediGate (encapsulated Modbus, not Open Modbus/TCP).
Test Tries	Enter the number of tries to read or write Modbus data to the device when secondary route testing is performed.
Test Day	Select the day of the week on which to initiate Secondary Slave testing. *Select the day, or "Never" to disable the test.*
Slave Virtual Unit	Enter the Modbus slave device address.
Write Address	nter the starting register address to use for writing data. *Starting address should be a 40xxx register.*
Write Num Registers	Enter the number of registers to write, or 0 to disable the write test.
Read Address	Enter the starting register address to use for reading data.
Read Num Registers	Enter the number of registers to read, or 0 to disable the read test.
Response Timeout	Enter the number of seconds to wait for slave read or write response.

SNMP

The SNMP object is a configuration of a few properties used by the Simple Network Management Protocol. SNMP is an Internet protocol used to manage nodes on an IP network.

One component of SNMP is the MIB (Management Information Base), containing a set of parameters which can be queried from an SNMP management station. Linux uses a standard MIB-II (RFC1213) system group, Interfaces Group and IP Group using a standard SNMP Manager. The access is provided by a read-only community name, with no support for SNMP traps. The MIBs are located under /usr/director/bin/mibs.

Attributes	Function
Object Type	SNMP
Parent(s)	System Networks

Instance	Must be 0

Properties	Values
	The following parameters are stored into the SNMP configuration file, located at /etc/snmpd.conf.
rocommunity	Read-only community name. The RediGate currently only supports read-only community, not read-write community. *Enter a text string between 1 and 63 characters.*
sysdescr	User-defined system description. *Enter a text string between 1 and 127 characters.*
syslocation	User-defined system location. *Enter a text string between 1 and 127 characters.*
syscontact	System contact of the individual who manages this system. *Enter a text string between 1 and 127 characters.* *The following parameters are used as command-line arguments for the script which calls the snmpd service. Elecsys uses a standard Linux SNMP agent, and documentation on these properties can be obtained from public sources if extra options might be needed.*
Agent_ExtraOpts	Extra command line options for the SNMP agent service may be entered here. Normally, this should be left blank. *Text string must be 127 characters or less.*
Agent_ListenOn	This sets the port to listen for an SNMP management system connection. *The default option, 'UDP:161' establishes a server on the standard UDP port 161 to use for SNMP. Multiple ports or protocols (such as TCP) can be added, separated by commas. For example the string 'UDP:161,5000,TCP:2000@localhost' would listen for SNMP on ports 161 and 5000 using UDP protocol, and using TCP protocol on localhost only at port 2000.*