LM Direct Reference Guide

1 Introduction

Welcome to the LM Direct™ Reference Manual. This manual is intended to give you detailed information on the usage and operation of the LM Direct™ location messaging interface. This includes message specifications, operational issues, unit configuration and messaging examples.

1.1 About The Reader

In order to limit the size and scope of this manual we have made some assumptions about the reader.

- Familiar with basic programming concepts
- Understands IP networking concepts and the UDP protocol
- Familiar with GPS concepts and terminology
- Able to query and set values into an LMU using LMU Manager or AT Commands
- Familiar with the features and capabilities of the LMU

1.2 About CalAmp

CalAmp is a leading provider of wireless communications products that enable anytime/anywhere access to critical information, data and entertainment content. With comprehensive capabilities ranging from product design and development through volume production, CalAmp delivers cost-effective high quality solutions to a broad array of customers and end markets. CalAmp is the leading supplier of Direct Broadcast Satellite (DBS) outdoor customer premise equipment to the U.S. satellite television market. The Company also provides wireless data communication solutions for the telemetry and asset tracking markets, private wireless networks, public safety communications and critical infrastructure and process control applications. For additional information, please visit the Company's website at www.CalAmp.com (http://www.calamp.com) .

1.3 Overview

The LM Direct™ interface was designed as a means for an LMU to provide location and status information to an application without requiring the LM Exchange middleware. It also allows the application to read and write the LMU's configuration parameters including the PEG event list and trigger parameters. These applications typically will take one of two forms: Either an in-vehicle application running on a laptop (or PDA) or a management/tracking application on a back-end system.

LM Direct™ provides the following features and capabilities to an application:

- Location Data — Time-of-Fix, Latitude, Longitude, Heading, Speed, Altitude, HDOP, Number of Satellites and Fix Status
- Communications Data — Signal Strength (RSSI), Comm State and Carrier/Operator Identifiers
- LMU Data — Mobile ID, Event Code, Event Index, Event Timestamp, Report Sequence Number, Input States, and Accumulator values.
- PEG-based event reporting
- User messaging between the application and a serial device attached to the LMU
- Full programming of the LMU's PEG functionality
- Full programming of other LMU configuration parameters
- Acknowledgement and retransmission functionality
- Optional logging of event reports and user messages in the LMU when delivery to the application isn't possible

1.4 Communications

The LM Direct™ interface in the LMU listens on UDP port 20510 for any requests generated by the application; that is, all outbound messages (i.e., those sent to an LMU) are sent to port 20510. Responses to outbound messages (e.g., ACK/NAK messages) are sent back to the originating IP address and port. Inbound request messages (e.g., Event Reports, User Messages) are sent by the LMU to the IP address and port currently selected from the Inbound Routing List. Typically the application will listen for these messages on port 20500.

2 Message Structures

Messages passed between an application and the LM Direct™ interface have the same basic packet structure. This structure consists of a 20 byte IP Header, an 8 byte UDP header, the LM Direct™ header and the LM Direct™ message.

Note that all bytes in multi-byte fields are transmitted in Net Endian format (Big Endian) where the most significant bits are transmitted first. For example, for a 32-bit field, bits 31-24 are transmitted first, 16-23 second, 8-15 third and 0-7 last

2.1 IP Header

The IP Header is the first section of code received in the Basic Message packet structure. This format will not be included for each type of message throughout this doc, but will be assumed to preceed each of the messages described.
This is a sample IP Header:

```
Raw Data:
  45 00 00 51 00 0A 00 00 80 11 BE 8B 33 33 33 33 44 69 F2 37
  ...
```

```
Decoded:
  4            Version
  5            Header Length
  00           Type of Service (TOS)
  0051         Total Length (in bytes)
  000A         Identification
  0000         Flags & Fragment Offset
  80           Time to Live (TTL)
  11           Protocol
  BE8B         Header Checksum
  33333333     Source IP
  4469F237     Destination IP: 68.105.242.55
```

2.2 UDP Header

The UDP Header follows the IP Header in the Basic Message packet structure. This format will not be included for each type of message throughout this doc, but will be assumed to preceed each of the messages described.
This is a sample UDP Header

```
Raw Data:
  45 00 00 51 00 0A 00 00 80 11 BE 8B 33 33 33 33 44 69 F2 37
  50 1E 50 14 00 3D 52 3B ...

Decoded
  501E         Source Port: 20510 (The Port the LMU uses to send)
  5014         Destination Port: 20500 (The server port)
  003D         UDP Packet Length: 61 Bytes
  523B         UDP Checksum
```

2.3 Basic Message Structure

The generic message structure of an LM Direct™ message is:

	7 0	
Options Header	Options Byte (MSBit always set)	
	Mobile ID Length	These fields present only if corresponding bit is set on options bite.
	Mobile ID Byte 0	
	…	
	Mobile ID Byte n	
	Mobile ID Type Length	
	Mobile ID Type Byte	
	Authentication Length (=4)	
	Authentication Byte 0	
	…	
	Authentication Byte 3	
	Routing Length (=8)	
	Routing Byte 0	
	…	
	Routing, up to Byte 7	
	Forwarding Length (= 8)	
	Forwarding Address (4 Bytes)	
	Forwarding Port (2 Bytes)	
	Forwading Protocol (1 Byte)	
	Forwarding Operation (1 Byte)	
	Resp. Redirection Length (= 6)	
	Resp. Redirection Address (4 bytes)	
	Resp. Redirection Port (2 bytes)	
Message Header	Service Type	
	Message Type	
	Sequence Number (msByte)	
	Sequence Number (lsByte)	
Message Contents	Message Byte (msByte)	
	…	
	Message Byte (lsByte)	

An LM Direct™ message comprises the Options Header, Message Header and Message Contents. The Options byte at the beginning of the Options Header is bit-mapped to indicate which Options Header fields will follow. If the most-significant bit (bit 7) of the first byte of a message is set, the Options Header is present in the message beginning with the Options Byte. Otherwise, the message begins with the Message Header (Service Type is the first byte); that is to say that if the first byte of an LM Direct™ message has the most-significant bit set (bit 7), than the Options Header is present. Likewise, if it is cleared, the Options Header is not included in the message.

2.3.1 Options Header

The Options Header fields used in LM Direct™ are defined as follows:

Options

Header Content Options:
Bit 0: Mobile ID (0 = disabled, 1= enabled)
Bit 1: Mobile ID Type (0 = disabled, 1 = enabled)
Bit 2: Authentication Word (0 = disabled, 1 = enabled)
Bit 3: Routing (0 = disabled, 1 = enabled)
Bit 4: Forwarding (0 = disabled, 1 = enabled)
Bit 5: Response Redirection (0=disabled, 1=enabled)
Bit 6: Not Used – Reserved for future use
Bit 7: Always set.

Mobile ID Length

The number of bytes to follow which contain the Mobile ID data

Mobile ID

The Mobile ID of the LMU that either originated the LM Direct™ message or the LMU for which the message is intended. The Mobile ID is made up of numerical digits and is encoded in this field as packed BCD. The most significant digit is placed in the upper four bits of the first byte. In the case where an odd number of digits are used, a 0x0F is used to pad the lower 4 bits of the last byte.

Mobile ID Type Length

The number of bytes to follow which contain the Mobile ID Type data. Note that this value will always be 1.

Mobile ID Type

The type of Mobile ID being used by the LMU:
0 – OFF
1 – Electronic Serial Number (ESN) of the LMU
2 – International Mobile Equipment Identifier (IMEI) or Electronic Identifier (EID) of the wireless modem
3 – International Mobile Subscriber Identifier (IMSI) of the SIM card (GSM/GPRS devices only)
4 – User Defined Mobile ID
5 – Phone Number of the mobile (if available)
6 – The current IP Address of the LMU

Authentication Length

The number of bytes to follow which contain the Authentication word; normally this value is 4

Authentication

The Authentication word is used with an outbound message to gain access to the LMU if authentication is enabled.

Routing Length

The number of bytes to follow which contain the Routing field. If present, this value is user defined and can be up to 8.

Routing

The Routing field is up to 8 bytes long and can be used by the Application to store identity information of the originating user for requests made to the LMU from an application subsystem. It is only used in outbound messages. If the Routing field is present in a request received by the LMU, the LMU will include the same field, byte for byte, in the response sent back to the Application. The Application can then route the response back to the originating user based on information received. See Appendix D for further definition and uses of the Routing Field.

Forwarding Length

The number of bytes to follow which contain the Forwarding field. If present, this value is normally 8.

Forwarding

The Forwarding field is 8 bytes long and is used by an LM Direct™ forwarding agent to manage source and destination addresses and ports. It is only used in outbound messages. If the Forwarding field is present in a request received by the LMU, the LMU will include the same field, byte for byte, in the response sent back. The first four bytes of the forwarding field contain the forwarding address. The next two bytes contain the forwarding port. The following byte contains the forwarding protocol (TCP=6, UDP=17). The last byte is for the Forwarding Operation Type. The current operations are Forward (0x00), Proxy (0x01) and Forward with Lookup (0x02). See Appendix D for further definition and uses of the Forwarding field.

Response Redirection Length

The number of bytes in the following Response Redirection field. If present, this value is normally 6.

Response Redirection

The Response Redirection field is 6 bytes long and is used by an LM Direct™ recipient to redirect an acknowledged response back to the specified address and port instead of back to the sender. The first four bytes of the response redirection field contain the redirection address. The next two bytes contain the redirection port.

Options Extension Length
The number of bytes to follow which contain the Options Extension word; normally this value is 1

Option Extension
The Options Extension field is 1-byte field is bit-mapped in the same manner as the Options byte. It contains bit flags that indicate if there are additional fields in the Options Header. If any flag bits are set in this byte, the respective fields will follow this byte with the usual length-followed-by-data format. The flags are bit mapped as follows:

Options Extension:
Bit 0: ESN (0 = disable, 1=enabled)
Bit 1: Not Used – Reserved for future use
Bit 2: Not Used – Reserved for future use
Bit 3: Not Used – Reserved for future use
Bit 4: Not Used – Reserved for future use
Bit 5: Not Used – Reserved for future use
Bit 6: Not Used – Reserved for future use
Bit 7: Not Used – Reserved for future use

ESN Length
The number of bytes to follow which contain the ESN data

ESN
The ESN is the Electronic Serial Number of the LMU that either originated the LM Direct message or for which the message is intended. Normally the ESN is only included for messages exchanged between the LMU and the Maintenance Server. The ESN is made up of numerical digits and is encoded in this field as packed BCD. The most significant digit is placed the upper four bits of the first byte. In the case where an odd number of digits are used, a 0x0F is used to pad the lower 4 bits of the last byte.

2.3.2 Message Header

The Message Header fields used in LM Direct™ are defined as follows:

Service Type
0 = Unacknowledged Request
1 = Acknowledged Request
2 = Response to an Acknowledged Request

Message Type
0 = Null message
1 = ACK/NAK message
2 = Event Report message
3 = ID Report message
4 = User Data message
5 = Application Data message
6 = Configuration Parameter message
7 = Unit Request message
8 = Locate Report message
9 = User Data with Accumulators message
10 = Mini Event Report message
11 = Mini User Message

Sequence Number
A 16-bit number used to uniquely identify a message. This number shall be initialized to 1 by the LMU on a cold boot and will be incremented in the LMU each time an inbound message is originated by the LMU. The LMU remembers its current Sequence Number during sleep. Eventually the Sequence Number will rollover from 65535 to 1, skipping zero.

The Sequence Number is also used to identify acknowledgements and retransmissions. Responses to Acknowledged Requests shall contain the Sequence Number from the associated Acknowledged Request. A Server receiving from an LMU can also use the Sequence Number as a means to identify whether the received message is an original or a retransmitted copy, thus avoiding processing any duplicate reports from an LMU.

A Server can set the Sequence Number field to zero for all outbound messages sent to the LMU if it is not important that the LMU distinguish an original message from a retransmitted copy. However, if this distinction is necessary, the server can insert a changing non-zero value into the Sequence Number field of each new message sent to the LMU. The LMU will remember the last sequence number it received and will compare it to the non-zero sequence number for the new message. If different, it will process the message normally. If the same, it will not process the message and will return a NAK response with the 'ACK' field set to 7.

2.4 Null Message (Message Type 0)

The Null message is an empty message and has no Message Content field. It is primarily used by the LMU which periodically sends it to the Server to keep network firewalls open so the Server can initiate messages to the LMU at any time. Normally the Null Message is sent as an Unacknowledged Request.

7	0
Options Header (optional)	
...	
Message Header	
...	

Null Message Format

Null Message Example

This is an example of a basic Null message format.

```
Raw Data:
  83 05 01 02 03 04 05 01 01 00 00 00 01

Decoded:
  -------Options Header--------
  83            Options Byte
  05            Mobile ID field length
  0102030405    Mobile ID
  01            Mobile ID Type Length
  01            Mobile ID Type
  -------Message Header--------
  00            Service Type, for this message 0 = Unacknowledged Request
  00            Message Type, for this message 0 = Null Message
  0001          Sequence number of the message you are responding to
```

2.5 Acknowledge Message (Message Type 1)

Acknowledge messages are used by the LMU or Server to acknowledge or reject requests made over the LM Direct™ interface.

7	0
Options Header (optional)	
...	
Message Header	
...	
Type	
ACK	
spare	
App Version (byte 0)	
App Version (byte 1)	
App Version (byte 2)	

Acknowledge Message Format

Type (1 byte)
This field indicates the type of message being acknowledged (e.g., it will match the 'Msg Type' field of the Request message)

ACK (1 byte)
0 – ACK (operation successful)
1 – NAK (operation failed, no reason)
2 – NAK (operation failed, not a supported message type)
3 – NAK (operation failed, not a supported operation)
4 – NAK (operation failed, unable to pass to serial port)
5 – NAK (operation failed, authentication failure)
6 – NAK (operation failed, Mobile ID look-up failure)
7 – NAK (operation failed, non-zero sequence number same as last received message)

Spare (1 byte)
This field is not used and should be set to zero.

App Version (3 bytes)
This 3-byte is used by the LMU to indicate its three-character application software version identifier (i.e. "60g"). This field should be set to all zeroes when the message is originated by the Server.

Acknowledge Message Example
This is an example of a basic Acknowledge message format as a Response to an Acknowledge Request.

```
Raw Data:
  83 05 01 02 03 04 05 01 01 02 01 00 01 00 00 00 00 00 00

Decoded:
  -------Message Header--------
  02            Service Type, for this message 2 = Response to Acknowledged Request
  01            Message Type, for this message 1 = ACK/NAK Message
  -----Acknowledge Message-----
  00            Type
  00            Ack
  00            Spare/Unused
  000000        App Version
```

2.6 Event Report Message (Message Type 2)

Event Report messages are initiated by the LMU and are generated by the LMU's Programmable Event Generator (PEG). They can be either Acknowledged or Unacknowledged Requests. The Server should respond to an Acknowledged Event Report Request with an Acknowledge Message. Event reports have the following structure:

15	0
Options Header (optional)	...
...	...
Message Header	...
...	...
Update Time (msByte)	Update Time
Update Time	Update Time (lsByte)
Time of Fix (msByte)	Time of Fix
Time of Fix	Time of Fix (lsByte)
Latitude (msByte)	Latitude
Latitude	Latitude(lsByte)
Longitude (msByte)	Longitude
Longitude	Longitude (lsByte)
Altitude (msByte)	Altitude
Altitude	Altitude(lsByte)
Speed (msByte)	Speed
Speed	Speed (lsByte)
Heading (msByte)	Heading (lsByte)
Satellites	Fix Status
Carrier (msByte)	Carrier (lsByte)
RSSI (msByte)	RSSI (lsByte)
Comm State	HDOP
Inputs	Unit Status
Event Index	Event Code
Accums	spare
Accum List(msByte)	Accum List
Accum List	Accum List(lsByte)
...	...

Event Report Message Format

Update Time (4 bytes)
The time tag of the message in seconds, referenced from Jan. 1, 1970

Time of Fix (4 bytes)
The last known time of fix from the GPS satellites. This value is reported in seconds from Jan. 1, 1970

Latitude (4 bytes)
The latitude reading of the GPS receiver, measured in degrees with a 1x10^-7 degree lsb, signed 2's complement

Longitude (4 bytes)
The longitude reading of the GPS receiver, measured in degrees with a 1x10^-7 degree lsb, signed 2's complement

Altitude (4 bytes)
The altitude reading of the GPS receiver measured in centimeters above the WGS-84 Datum, signed 2's complement

Speed (4 bytes)
The speed as reported by the GPS receiver, measured in centimeters per second

Heading (2 bytes)
The heading value reported in degrees from true North

Satellites (1 byte)
The number of satellites used in the GPS solution

Fix Status (1 byte)
The current fix status of the GPS receiver bitmapped as follows:
Bit 0 – Predicted

> Bit is set when the position update has a horizontal position accuracy estimate that is less that the Horizontal Position Accuracy Threshold defined in S-Register 142 (and the threshold is non-zero).

Bit 1 – Differentially Corrected

> This bit is set when WAAS DGPS is enabled (S-Register 139) and the position has been differentially corrected

Bit 2 – Last Known

> This bit is set when the current GPS fix is invalid but a previous fix's value is available.

Bit 3 – Invalid Fix

> This bit is set only after a power-up or reset before a valid fix is obtained.

Bit 4 – 2D Fix

This bit is set when 3 or fewer satellites are seen/used in the GPS fix. (i.e. with 3 satellites or less, an altitude value cannot be calculated)

Bit 5 – Historic

This bit is set when the message has been logged by the LMU.

Bit 6 – Invalid Time

This bit is set only after a power-up or reset before a valid time-sync has been obtained.

Carrier (2 bytes)
The identifier of the Carrier/Operator the wireless modem is currently using

RSSI (2 bytes)
The received signal strength of the wireless modem in dBm. This value is signed in a 2's complement format.

Comm State (1 byte)
The current state of the wireless modem bit mapped as follows
Bit 0 – Available
Bit 1 – Network Service
Bit 2 – Data Service
Bit 3 – Connected (PPP Session Up)
Bit 4 – Voice Call is Active
Bit 5 – Roaming
Bit 6 – 3G Network (ie UMTS)
Bit 7 – Not Used

HDOP (1 byte)
The GPS Horizontal Dilution of Precision - it is a unit-less value reported with a 0.1 lsb.

Inputs (1 byte)
The current state of the inputs, bit mapped as follows:
Bit 0 – Ignition
Bit 1 – Input 1
Bit 2 – Input 2
Bit 3 – Input 3
Bit 4 – Input 4
Bit 5 – Input 5

Bit 6 – Input 6
Bit 7 – Input 7

Unit Status (1 byte)
Status of key modules within the unit:
Bit 0 – LMU32: HTTP OTA Update Status (0=Ok, 1=Error), LMU8: Unused
Bit 1 – GPS Antenna Status (0=OK, 1=Error)
Bit 2 – GPS Receiver Self-Test (0=OK, 1=Error) (LMU32 only)
Bit 3 – GPS Receiver Tracking (0=Yes, 1=No)
Bit 4 – Reserved, Currently Unused
Bit 5 – Reserved, Currently Unused
Bit 6 – Reserved, Currently Unused
Bit 7 – Unused

Event Index (1 byte)
The index number of the event that generated the report; values should range from 0-74. 255 represents a Real Time PEG Action request.

Event Code (1 byte)
The event code assigned to the report as specified by the event's Action Parameter

Accums (1 byte)
The number of 4-byte values in the AccumList and the Accumulator Reporting Format Type (upper 2 bits). Refer to Appendix G, 'Accumulator Reporting Formats' for details.

Spare (1 byte)
Not used - Set to 0.

AccumList (4 bytes x 'n')
A list of 'n' 4-byte fields where 'n' is defined in the Accums field. The format for this list is dependent upon the Accumulator Reporting Format Type also defined in the Accums field. Refer to Appendix G, 'Accumulator Reporting Formats' for details.

Example Event Report Format
This is an example of a basic Event Report message format.

```
Raw Data:
 83 05 01 02 03 04 05 01 01 01 02 00 01 4F B4 64 88 4F B4 64
 88 13 BF 71 A8 BA 18 A5 06 00 00 1333 00 00 00 00 11 11 02
 33 44 44 55 55 66 77 88 99 10 11 ?? 00 ??

Decoded:
  -------Message Header--------
  01            Service Type, for this message 1 = Acknowledged Request
  02            Message Type, for this message 2 = Event Report
  -------Event Report----------
  4FB46488      Update Time (5/17/12 @ 2:38pm)
  4FB46488      Time of Fix (5/17/12 @ 2:38pm)
  13BF71A8      Latitude (33.1313576)
  BA18A506      Longitude (-117.2790010)
  00001333      Altitude
  00000000      Speed
  1111          Heading
  02            Satellites
  33            Fix Status
  4444          Carrier
  5555          RSSI
  66            Comm State
  77            HDOP
  88            Inputs
  99            Unit Status
  10            Event Index
  11            Event Code
  ??            Accums (# of 4-byte Accum List values)
  00            Spare
  ??            Accum List (Varies depending on the # of Accums reporting)
```

2.7 ID Report Message (Message Type 3)

The ID Report message is sent by the LMU under three scenarios: (1) As a result of a PEG Action (Acknowledged or Unacknowledged Request), (2) In response to a Unit Request Message (Message Type 7) (Response) or (3) In response to an SMS “SENDTO” request (Unacknowledged Request). The Server should respond to an Acknowledged ID Report Request with an Acknowledge Message. ID Reports have the following structure:

7 0
Options Header (optional)
...
Message Header
...
Script Version
Config Version
App Version (byte 0)
App Version (byte 1)
App Version (byte 2)
Vehicle Class
Unit Status
Modem Selection
Application ID
Mobile ID Type
Query ID (byte 0)
…
Query ID (byte 3)
ESN (byte 0)
…
ESN (byte 7)
IMEI (byte 0)
…
IMEI (byte 7)
IMSI (byte 0)
…
IMSI (byte 7)
MIN (byte 0)
…
MIN (byte 7)
ICC-ID (byte 0)
…
ICC-ID (byte 9)
Extension String(s)

ID Report Message Format

Script Version (1 byte)
A user-defined value stored in S-Register 121 intended to identify the version of the PEG script stored in the LMU

Config Version (1 byte)
A user-defined value stored in S-Register 143 intended to identify the version of the configuration parameters stored in the LMU

App Version (3 bytes)
The LMU's three-character software version identifier (e.g., "60g")

Vehicle Class (1 byte)
A user-defined value stored in S-Register 147 intended to identify the class of the vehicle in which the LMU is installed

Unit Status (1 byte)
Status of key modules within the unit:
Bit 0 – LMU32: HTTP OTA Update Status (0=Ok, 1=Error), LMU8: Unused
Bit 1 – GPS Antenna Status (0=OK, 1=Error)
Bit 2 – GPS Receiver Self-Test (0=OK, 1=Error) (LMU32 only)
Bit 3 – GPS Receiver Tracking (0=Yes, 1=No)
Bit 4 – Reserved, Currently Unused
Bit 5 – Reserved, Currently Unused
Bit 6 – Reserved, Currently Unused
Bit 7 – Unused

Modem Selection (1 byte)
A value stored in S-Register 120 that identifies the LMU's current modem type configuration

Application ID (1 byte)
This field identifies the Application Program software running on the device. Typically it is unique to a device hardware configuration.

Mobile ID Type (1 byte)
A value stored in S-Register 145 that identifies the LMU's current Mobile ID type configuration

Query ID (4 bytes)
A value that identifies the request that initiated the generation of the ID Report message. This is normally only used for SMS initiated ID Reports. This field is zero for all other uses.

ESN (8 bytes, Packed BCD)
The Electronic Serial Number of the LMU in packed BCD form. The most significant digit is placed in the upper four bits of the first byte. Unused trailing digits (4-bits) are filled with all ones (1111).

IMEI (8 bytes, Packed BCD)
The Electronic Serial Number of the modem in the LMU in packed BCD form. The most significant digit is placed in the upper four bits of the first byte. Unused trailing digits (4-bits) are filled with all ones (1111).

IMSI (8 bytes, Packed BCD)
The Subscriber ID of the SIM used in the GSM/GPRS modem in packed BCD form. The most significant digit is placed in the upper four bits of the first byte. Unused trailing digits (4-bits) are filled with all ones (1111).

MIN (8 bytes, Packed BCD)
The phone number of the modem in the LMU (if available) in packed BCD form. The most significant digit is placed in the upper four bits of the first byte. Unused trailing digits (4-bits) are filled with all ones (1111).

ICC-ID (10 bytes, Packed BCD)
The serial number of the SIM used in the GSM/GPRS modem in packed BCD form. The most significant digit is placed in the upper four bits of the first byte. Unused trailing digits (4-bits) are filled with all ones (1111).

Extension Strings(0 to 'n' bytes)
This field is optional and will contain information fields in the form of one or more null-terminated ASCII strings. Each string begins with a tag which indicates the contents of the string. A tag consists of alpha-numeric ASCII characters followed by a colon (':') followed by the data portion of the string. The following are currently defined Extension Strings:

MEID:<CDMA Modem serial number>
PRI:<CDMA Modem PRI version>

OTA:<Comma separated list of supported protocols>|<DeviceType1>;<Comma separated list of supported files for DeviceType1>|<Device Type2>;<Comma separated list of supported files for the Device Type2>;<Device Type'n'>;<Comma separated list of supported protocols for Device Type 'n'>

OTA Example 1:

OTA:1|0;0,1|1;0,1
Protocols Supported: HTTP (1)
First Device supported: LMU (0)
Files updateable over the air for LMU: Firmware (0) and Config (1)
Next device supported: JPod (1)
Files updateable over the air for JPod: Firmware(0) and Config/Emulation(1)

OTA Example 2:

OTA:1,2|0;0,1|2;0,1,2,3
Protocols Supported : HTTP (1), FTP (2)
First Device supported: LMU (0)
Files updateable over the air for LMU: Firmware (0) and Config (1)
Next device supported: OBD II Type1 (2)
Files updateable over the air for OBD II Type 1: Firmware(0), Database(1), SysManager (2), Boot Loader (3)

Please refer to Appendix E, section ‘OTA Download App Message (Msg Type 107) for details on the values defined for the Device Type, File Type and Protocol fields.

Extension Strings are only present in ID Reports when they are necessary to add additional information to the ID Report message.

2.8 User Message (Message Type 4)

User Messages are used to pass user data from the Server to the Host serial port on the LMU and vice versa. The Host port must be configured to operate in this mode. User Messages sent by the LMU contain location and status fields that are not present in the User Messages sent by the Server. Note that the maximum number of user data bytes is 848. User Messages have the following structure:

15	0
Options Header (optional)	...
...	...
Message Header	...
...	...
Update Time (msByte)	Update Time
Below is present only when sent by LMU	
Update Time	Update Time (lsByte)
Time of Fix (msByte)	Time of Fix
Time of Fix	Time of Fix (lsByte)
Latitude (msByte)	Latitude
Latitude	Latitude(lsByte)
Longitude (msByte)	Longitude
Longitude	Longitude (lsByte)
Altitude (msByte)	Altitude
Altitude	Altitude(lsByte)
Speed (msByte)	Speed
Speed	Speed (lsByte)
Heading (msByte)	Heading (lsByte)
Satellites	Fix Status
Carrier (msByte)	Carrier (lsByte)
RSSI (msByte)	RSSI (lsByte)
Comm State	HDOP
Inputs	Unit Status
Above is present only when sent by LMU	
User Msg Route	User Msg ID
User Msg Length (msByte)	User Msg Length (lsByte)
User Msg (msByte)	…
…	User Msg (lsByte)

User Message Format

Update Time (4 bytes)

The time tag of the message in seconds, referenced from Jan. 1, 1970

Time of Fix (4 bytes)

The last known time of fix from the GPS satellites. This value is reported in seconds from Jan. 1, 1970

Latitude (4 bytes)

The latitude reading of the GPS receiver measured in degrees with a 1x10-7 degree lsb, signed 2's complement.

Longitude (4 bytes)
The longitude reading of the GPS receiver measured in degrees with a 1x10-7 degree lsb, signed 2's complement.

Altitude (4 bytes)
The altitude reading of the GPS receiver measured in centimeters above the WGS-84 Datum, signed 2's complement.

Speed (4 bytes)
The speed as reported by the GPS receiver measured in centimeters per second

Heading (2 bytes)
The heading value reported in degrees from true North

Satellites (1 byte)
The number of satellites used in the GPS solution

Fix Status (1 byte)
The current fix status of the GPS receiver bitmapped as follows:
Bit 0 – Predicted

> Bit is set when the position update has a horizontal position accuracy estimate that is less that the Horizontal Position Accuracy Threshold defined in S-Register 142 (and the threshold is non-zero).

Bit 1 – Differentially Corrected

> This bit is set when WAAS DGPS is enabled (S-Register 139) and the position has been differentially corrected.

Bit 2 – Last Known

> This bit is set when the current GPS fix is invalid but a previous fix's value is available.

Bit 3 – Invalid Fix

> This bit is set only after a power-up or reset before a valid fix is obtained.

Bit 4 – 2D Fix

This bit is set when 3 or fewer satellites are seen/used in the GPS fix. (i.e. with 3 satellites or less, an altitude value cannot be calculated)

Bit 5 – Historic

This bit is set when the message has been logged by the LMU.

Bit 6 – Invalid Time

This bit is set only after a power-up or reset before a valid time-sync has been obtained.

Carrier (2 bytes)
The identifier of the Carrier/Operator the wireless modem is currently using

RSSI (2 bytes)
The received signal strength of the wireless modem in dBm. This value is signed in a 2's complement format.

Comm State (1 byte)
The current state of the wireless modem bit mapped as follows
Bit 0 – Available
Bit 1 – Network Service
Bit 2 – Data Service
Bit 3 – Connected (PPP Session Up)
Bit 4 – Voice Call is Active
Bit 5 – Roaming
Bit 6 – 3G Network (ie UMTS)
Bit 7 – Not Used

HDOP (1 byte)
The GPS Horizontal Dilution of Precision - it is a unit-less value reported with a 0.1 lsb.

Inputs (1 byte)
The current state of the inputs, bit mapped as follows:
Bit 0 – Ignition
Bit 1 – Input 1
Bit 2 – Input 2
Bit 3 – Input 3

Bit 4 – Input 4
Bit 5 – Input 5
Bit 6 – Input 6
Bit 7 – Input 7

Unit Status (1 byte)
Status of key modules within the unit:
Bit 0 – LMU32: HTTP OTA Update Status (0=Ok, 1=Error), LMU8: Unused
Bit 1 – GPS Antenna Status (0=OK, 1=Error)
Bit 2 – GPS Receiver Self-Test (0=OK, 1=Error) (LMU32 only)
Bit 3 – GPS Receiver Tracking (0=Yes, 1=No)
Bit 4 – Reserved, Currently Unused
Bit 5 – Reserved, Currently Unused
Bit 6 – Reserved, Currently Unused
Bit 7 – Unused

User Msg Route (1 byte)
This field was formerly the upper byte of the User Msg Type field. This new field defines which port the message originated from, or should be sent to: The possible values are:
0 – User 0 (Host) Port
2 – User 1 (Aux) Port
1,3-255 – Reserved for future use

User Msg ID (1 byte)
This field was formerly the lower byte of the User Msg Type field. This new field contains the user defined ID value assigned to the User Message. The value of this field is defined by S-Register 136 or S-Register 166 depending on which port originated or is receiving the message.

User Msg Length (2 bytes)
The length, in bytes, of the user message that follows

User Msg (n Bytes)
The contents of the user message

2.9 Application Message (Message Type 5)

Application Messages are used to pass LMU application messages to and from the LMU. They provide a subclass of messages used to manage the LMU application. See Appendix E for a description of the possible Application messages types and formats supported by the LMU. Application Messages sent by the LMU contain location and status fields that are not present in the Application Messages sent by the Server. The maximum number of data bytes is 848. Application Messages have the following structure:

15	**0**
Options Header (optional)	...
...	...
Message Header	...
...	...
Below is present only when sent by LMU.	
Update Time (msByte)	Update Time
Update Time	Update Time (lsByte)
Time of Fix (msByte)	Time of Fix
Time of Fix	Time of Fix (lsByte)
Latitude (msByte)	Latitude
Latitude	Latitude(lsByte)
Longitude (msByte)	Longitude
Longitude	Longitude (lsByte)
Altitude (msByte)	Altitude
Altitude	Altitude(lsByte)
Speed (msByte)	Speed
Speed	Speed (lsByte)
Heading (msByte)	Heading (lsByte)
Satellites	Fix Status
Carrier (msByte)	Carrier (lsByte)
RSSI (msByte)	RSSI (lsByte)
Comm State	HDOP
Inputs	Unit Status
Above is present only when sent by LMU.	
App Msg Type (msByte)	App Msg Type (lsByte)
App Msg Length (msByte)	App Msg Length (lsByte)
App Msg (msByte)	…
…	App Msg (lsByte)

Application Message Format

Update Time (4 bytes)

The time tag of the message in seconds, referenced from Jan. 1, 1970

Time of Fix (4 bytes)

The last known time of fix from the GPS satellites. This value is reported in seconds from Jan. 1, 1970

Latitude (4 bytes)
The latitude reading of the GPS receiver measured in degrees with a 1x10-7 degree lsb, signed 2's complement.

Longitude (4 bytes)
The longitude reading of the GPS receiver measured in degrees with a 1x10-7 degree lsb, signed 2's complement.

Altitude (4 bytes)
The altitude reading of the GPS receiver measured in centimeters above the WGS-84 Datum, signed 2's complement.

Speed (4 bytes)
The speed as reported by the GPS receiver measured in centimeters per second

Heading (2 bytes)
The heading value reported in degrees from true North

Satellites (1 byte)
The number of satellites used in the GPS solution

Fix Status (1 byte)
The current fix status of the GPS receiver bitmapped as follows:
Bit 0 – Predicted

> Bit is set when the position update has a horizontal position accuracy estimate that is less that the Horizontal Position Accuracy Threshold defined in S-Register 142 (and the threshold is non-zero).

Bit 1 – Differentially Corrected

> This bit is set when WAAS DGPS is enabled (S-Register 139) and the position has been differentially corrected.

Bit 2 – Last Known

This bit is set when the current GPS fix is invalid but a previous fix's value is available.

Bit 3 – Invalid Fix

This bit is set only after a power-up or reset before a valid fix is obtained.

Bit 4 – 2D Fix

This bit is set when 3 or fewer satellites are seen/used in the GPS fix. (i.e. with 3 satellites or less, an altitude value cannot be calculated)

Bit 5 – Historic

This bit is set when the message has been logged by the LMU.

Bit 6 – Invalid Time

This bit is set only after a power-up or reset before a valid time-sync has been obtained.

Carrier (2 bytes)
The identifier of the Carrier/Operator the wireless modem is currently using

RSSI (2 bytes)
The received signal strength of the wireless modem in dBm. This value is signed in a 2's complement format.

Comm State (1 byte)
The current state of the wireless modem bit mapped as follows
Bit 0 – Available
Bit 1 – Network Service
Bit 2 – Data Service
Bit 3 – Connected (PPP Session Up)
Bit 4 – Voice Call is Active
Bit 5 – Roaming
Bit 6 – 3G Network (ie UMTS)
Bit 7 – Not Used

HDOP (1 byte)
The GPS Horizontal Dilution of Precision - it is a unit-less value reported with a 0.1 lsb.

Inputs (1 byte)

The current state of the inputs, bit mapped as follows:
Bit 0 – Ignition
Bit 1 – Input 1
Bit 2 – Input 2
Bit 3 – Input 3
Bit 4 – Input 4
Bit 5 – Input 5
Bit 6 – Input 6
Bit 7 – Input 7

Unit Status (1 byte)
Status of key modules within the unit:
Bit 0 – LMU32: HTTP OTA Update Status (0=Ok, 1=Error), LMU8: Unused
Bit 1 – GPS Antenna Status (0=OK, 1=Error)
Bit 2 – GPS Receiver Self-Test (0=OK, 1=Error) (LMU32 only)
Bit 3 – GPS Receiver Tracking (0=Yes, 1=No)
Bit 4 – Reserved, Currently Unused
Bit 5 – Reserved, Currently Unused
Bit 6 – Reserved, Currently Unused
Bit 7 – Unused

App Msg Type (2 bytes)
The pre-defined 'type' value of the application message

App Msg Length (2 bytes)
The length, in bytes, of the application message that follows

App Msg (n Bytes)
The contents of the application message

Example Application Message Report Format
This is an example of a basic Application Message Report format.

```
Raw Data:
 83 05 01 02 03 04 05 01 01 01 02 00 01 4F B4 64 88 4F B4 64
 88 13 BF 71 A8 BA 18 A5 06 00 00 1333 00 00 00 00 11 11 02
 33 44 44 55 55 66 77 88 99 00 10 00 11 ?? ?? ??

Decoded:
```

```
-------Message Header--------
01           Service Type, for this message 1 = Acknowledged Request
02           Message Type, for this message 5 = Application Data Message
-------Event Report----------
4FB46488     Update Time (5/17/12 @ 2:38pm)
4FB46488     Time of Fix (5/17/12 @ 2:38pm)
13BF71A8     Latitude (33.1313576)
BA18A506     Longitude (-117.2790010)
00001333     Altitude
00000000     Speed
1111         Heading
02           Satellites
33           Fix Status
4444         Carrier
5555         RSSI
66           Comm State
77           HDOP
88           Inputs
99           Unit Status
0010         App Msg Type (Defined in Appendix E)
0011         App Msg Length
??????       App Msg (Varies depending on the type of App Msg)
```

2.10 Parameter Message (Message Type 6)

The Parameter message is used by the Server as a request to read or write one or more configuration Parameters in the LMU. It is also used as a report by the LMU to report parameter values in response to a read request or as an indication of invalid parameters in response to a write request.

2.10.1 Parameter Update Begin Operation

A parameter update begin operation is used to notify the LMU that a configuration change is about to start. Receipt of this message will cause the PEG Trigger of Update Begin with a Trigger Parameter of 0 to be fired.

2.10.2 Parameter Write Operation:

A parameter write request message consists of one or more Parameter IDs and associated 2-byte length words. Each Parameter ID and length word is followed by one or more sets of values. Each value set consists of a 1-byte index (index=0 for non-array parameters) plus one or more bytes of data. For bit-mapped operations, the data includes a mask followed by the value; non-bit-mapped is just the value. The last Parameter ID in the write request message shall always be a zero and is followed by a length of zero. The LMU will respond with a message that includes any error indications and ending with a Parameter ID of zero and a length of zero. An error indication for an invalid or non-supported Parameter ID is the bad Parameter ID followed by a length of zero. An error indication for an invalid or non-supported Parameter Index is a Parameter ID followed by a length of 1 followed by the invalid Parameter Index.

2.10.3 Parameter Update End Operation

A parameter update end operation is used to notify the LMU that a configuration change has been completed. Receipt of this message will cause the PEG Trigger of Update End with a Trigger Parameter of 0 to be fired.

2.10.4 Parameter Read Operation:

A parameter read request message consists of one or more Parameter IDs and associated 2-byte length words. Each Parameter ID and length word is followed by one or more 1-byte parameter indexes (index=0 for non-array parameters). The last Parameter ID in the read request message shall always be a zero followed by a length of zero. The LMU will respond with a message that includes data from only the successful reads. An invalid or unsupported Parameter ID will be ignored and no entry for it will be made in the response message. A valid Parameter ID will result in a response message entry of a Parameter ID followed by a 2-byte length word followed by one or more value sets. Each value set consists of a 1-byte index (index=0 for non-array parameters) plus one or more bytes of data. Each value set corresponds to a requested index for that Parameter ID unless the index is invalid or not supported, in which case it will not be include in the response.

Each parameter request sent by the Server contains an Action byte that specifies whether the operation is a read or a write. Following the Action byte, one or more parameters may be specified. Each parameter is identified by a 2-byte Parameter ID and a 1-byte Index. Some parameters have their own Parameter ID while others share an ID with the others in their list and are thus also identified by their Index (position within the list). The Index is always 0 for those parameters that are not part of a list.

The format for a parameter request with a write action is shown below. Each Parameter ID is followed by a 2-byte length which specifies how many bytes follow to specify the parameters having this ID. Each individual parameter having this Parameter ID comprises a 1-byte Index and one or more data bytes making up the value of the parameter itself. The number of data bytes is known for each Parameter ID. The list of Parameters is always terminated by a Parameter ID of 0 followed by a length of 0.

	7 — 0
	Options Header (optional)
	...
	Message Header
	...
	Action (Write Request=1)
Can have multiple instances	Parameter ID (msbyte)
	Parameter ID (lsbyte)
	Length (msbyte)
	Length (lsbyte)
	Parameter Index
	Data (msbyte)
	…
	Data (lsbyte)
	Parameter Index
	Data (msbyte)
	…
	Data (lsbyte)
	Parameter ID = 0 (msbyte)
	Parameter ID = 0 (lsbyte)
	Length = 0 (msbyte)
	Length = 0 (lsbyte)

Parameter Write Request Format

The format for a parameter request with a read action is shown below. Each Parameter ID is followed by a 2-byte Length which specifies how many bytes follow to specify the parameters having this ID. Each individual parameter requested having this Parameter ID is identified by a 1-byte Parameter Index. The list of Parameters is always terminated by a Parameter ID of 0 followed by a length of 0.

	7 — 0
	Options Header (optional)
	...
	Message Header
	...
	Action (Read Request=0)
Can have multiple instances	Parameter ID (msbyte)
	Parameter ID (lsbyte)
	Length (msbyte)
	Length (lsbyte)
	Parameter Index
	Parameter Index
	…
	Parameter Index
	Parameter ID = 0 (msbyte)
	Parameter ID = 0 (lsbyte)
	Length = 0 (msbyte)
	Length = 0 (lsbyte)

Parameter Read Request Format

The format for a parameter write report is shown below. Each Parameter ID is followed by a 2-byte length which specifies how many bytes follow to specify the parameters having this ID. Each individual parameter having this Parameter ID is comprised by a 1-byte Index and one or more data bytes making up the value of the parameter itself. The number of data bytes is known for each Parameter ID.

<table>
<tr><td></td><td>7   0</td><td></td></tr>
<tr><td></td><td>Options Header (optional)</td><td></td></tr>
<tr><td></td><td>...</td><td></td></tr>
<tr><td></td><td>Message Header</td><td></td></tr>
<tr><td></td><td>...</td><td></td></tr>
<tr><td></td><td>Action (Write Report=3)</td><td></td></tr>
<tr><td rowspan="9">Can have multiple instances</td><td>Parameter ID (msbyte)</td><td rowspan="4">Shows response for invalid Parameter ID</td></tr>
<tr><td>Parameter ID (lsbyte)</td></tr>
<tr><td>Length = 0 (msbyte)</td></tr>
<tr><td>Length = 0 (lsbyte)</td></tr>
<tr><td>Parameter ID (msbyte)</td><td rowspan="5">Shows response for invalid Parameter Index</td></tr>
<tr><td>Parameter ID (lsbyte)</td></tr>
<tr><td>Length = 0 (msbyte)</td></tr>
<tr><td>Length = 1 (lsbyte)</td></tr>
<tr><td>Parameter Index</td></tr>
<tr><td></td><td>Parameter ID = 0 (msbyte)</td><td></td></tr>
<tr><td></td><td>Parameter ID = 0 (lsbyte)</td><td></td></tr>
<tr><td></td><td>Length = 0 (msbyte)</td><td></td></tr>
<tr><td></td><td>Length = 0 (lsbyte)</td><td></td></tr>
</table>

Parameter Write Report Format

The format for a parameter read report is shown below. Each Parameter ID is followed by a 2-byte length which specifies how many bytes follow to specify the parameters having this ID. Each individual parameter having this Parameter ID comprises a 1-byte Index and one or more data bytes making up the value of the parameter itself. The number of data bytes is known for each Parameter ID.

	7 0
	Options Header (optional)
	...
	Message Header
	...
	Action (Read Report=2)
Can have multiple instances	Parameter ID (msbyte)
	Parameter ID (lsbyte)
	Length (msbyte)
	Length (lsbyte)
	Parameter Index
	Data (msbyte)
	…
	Data (lsbyte)
	Parameter ID = 0 (msbyte)
	Parameter ID = 0 (lsbyte)
	Length = 0 (msbyte)
	Length = 0 (lsbyte)

Parameter Read Report Format

The format for an Update Begin Request is shown below. This message contains only an action and the 0x0000 Parameter ID and 0x0000 Length values.

7 0
Options Header (optional)
...
Message Header
...
Action (Update Begin = 2)
Parameter ID = 0 (msbyte)
Parameter ID = 0 (lsbyte)
Length = 0 (msbyte)
Length = 0 (lsbyte)

Update Begin Request Format

The format for an Update End Request is shown below. This message contains only an action and the 0x0000 Parameter ID and 0x0000 Length values

7 0
Options Header (optional)
...
Message Header
...
Action (Update End = 3)
Parameter ID = 0 (msbyte)
Parameter ID = 0 (lsbyte)
Length = 0 (msbyte)
Length = 0 (lsbyte)

Update End Request Format

Action (1 byte)
This field is included in a Parameter Request to indicate whether the request is for a write operation or for a read operation:
0 – Read
1 – Write
2 – Update Begin
3 – Update End

Parameter ID (2 bytes)
This field holds the two-byte identifier of the parameter type being acted upon. Parameter Messages can contain one or more Param ID field along with associated length and index and value fields as necessary for Reads, Writes and Reports.

Length (2 bytes)
This field immediately follows each Param ID in the message to indicate how many bytes will follow to specify which Param Indexes and associated Param Values (if a write or report operation).

Parameter Index (1 byte)
This field indicates the position of the parameter within the list of parameters having the previous Param ID. Some parameters are the only member of the list so their Param Index is always 0. In a Parameter Read Request, there are two unique Param Index values that have special meaning: (a) A Param Index of 255 indicates to the LMU that is should return the number of members in its list for the Param ID and (b) A Param Index of 254 indicates to the LMU that it should return the values of every member of the list for the Param ID.

Data (1 or more bytes)
This field contains the parameter value and is present in Write Requests and Read Reports. The number of bytes for each Param Value field is dependent upon the Param ID which defines the type of parameter.

2.11 Unit Request Message (Message Type 7)

The Unit Request message is used by the Server to request a particular action of the LMU. The Action field defines the nature of the request and the Data8, Data16 and Data32 fields are used for some actions to provide additional information. The format for the Unit Request Message is shown below:

7 ... 0
Options Header (optional)
...
Message Header
...
Action
Data8
Data16 (msbyte)
Data16 (lsbyte)
Data32 (msbyte)
Data32 (2nd byte)
Data32 (3rd byte)
Data32 (lsbyte)

Unit Request Message Format

Action (1 byte)
This field defines the type of unit request being made

Data8 (1 byte)
Used for some Unit Requests as defined in Table below. Otherwise set to 0

Data16 (2 bytes)
Used for some Unit Requests as defined in Table below. Otherwise set to 0

Data32 (4 bytes)
Used for some Unit Requests as defined in Table below. Otherwise set to 0

Unit Request	Action Code	Data8	Data16	Data32
Reboots the LMU	1			
Returns a Version Report in an App Msg	2			
Generates a Real-time PEG Action	3	PEG Action Code	PEG Action Modifier (lower 8-bits)	
Change the authentication password for the LM Direct™ Interface	4			Password
Returns a GPS Status Report in an App Msg	5			
Clears the PEG Event List	6			
Set the Aux Sleep Time. Time=32	7			Time (in seconds)
Generates PEG Special Trigger. Index=8	8	Trigger Modifier		
Returns an ID Report	9			
Returns a Locate Report	10	No. of Accumulators		
Returns a Message Statistics Report in an App Msg	11			
Reset Message Statistics	12			
Clear Error Flags	13			
Force Inbound and Maintenance DNS Lookups	14			
Request State Report	15			
User Flag Update	16	User Flag State	User Flag Mask (8 LSBs)	Special Trigger Number (8 LSBs)
Set Zone to Current Location and Enable Zone	17	Zone #	Zone Hysterisis (meters)	Zone Size (radius in meters, limited to 429,000)
Battery Gauge	18			Battery Capacity (mAH)
Accumulator Schedule	19	Action 1=Write Config and (re)start schedule	Accumulator #	Schedule Configuration

Unit Request Data Field Usage

2.12 Locate Report Message (Message Type 8)

A Locate Report message is sent by the LMU in response to a request by the Application made with a Unit Request Message (Message Type 7) Action code 10. The number of accumulators reported in this message is specified in the Unit Request by the Application. Locate reports have the following structure:

15	0
Options Header (optional)	...
...	...
Message Header	...
...	...
Update Time (msByte)	Update Time
Update Time	Update Time (lsByte)
Time of Fix (msByte)	Time of Fix
Time of Fix	Time of Fix (lsByte)
Latitude (msByte)	Latitude
Latitude	Latitude(lsByte)
Longitude (msByte)	Longitude
Longitude	(lsByte)</center>
Altitude (msByte)	Altitude
Altitude	Altitude(lsByte)
Speed (msByte)	Speed
Speed	Speed (lsByte)
Heading (msByte)	Heading (lsByte)
Satellites	Fix Status
Carrier (msByte)	Carrier (lsByte)
RSSI (msByte)	RSSI (lsByte)
Comm State	HDOP
Inputs	Unit Status
Accums	spare
Accum List (msByte)	Accum List
Accum List	Accum List (lsByte)
...	...

Locate Report Message Format

Update Time (4 bytes)
The time tag of the message in seconds, referenced from Jan. 1, 1970

Time of Fix (4 bytes)
The last known time of fix from the GPS satellites. This value is reported in seconds from Jan. 1, 1970

Latitude (4 bytes)
The latitude reading of the GPS receiver measured in degrees with a 1x10^-7 degree lsb, signed 2's complement.

Longitude (4 bytes)
The longitude reading of the GPS receiver measured in degrees with a 1x10^-7 degree lsb, signed 2's complement.

Altitude (4 bytes)
The altitude reading of the GPS receiver measured in centimeters above the WGS-84 Datum, signed 2's complement.

Speed (4 bytes)
The speed as reported by the GPS receiver measured in centimeters per second

Heading (2 bytes)
The heading value reported in degrees from true North

Satellites (1 byte)
The number of satellites used in the GPS solution

Fix Status (1 byte)
The current fix status of the GPS receiver bitmapped as follows:
Bit 0 – Predicted

> Bit is set when the position update has a horizontal position accuracy estimate that is less that the Horizontal Position Accuracy Threshold defined in S-Register 142 (and the threshold is non-zero).

Bit 1 – Differentially Corrected

> This bit is set when WAAS DGPS is enabled (S-Register 139) and the position has been differentially corrected.

Bit 2 – Last Known

This bit is set when the current GPS fix is invalid but a previous fix's value is available.

Bit 3 – Invalid Fix

This bit is set only after a power-up or reset before a valid fix is obtained.

Bit 4 – 2D Fix

This bit is set when 3 or fewer satellites are seen/used in the GPS fix. (i.e. with 3 satellites or less, an altitude value cannot be calculated)

Bit 5 – Historic

This bit is set when the message has been logged by the LMU.

Bit 6 – Invalid Time

This bit is set only after a power-up or reset before a valid time-sync has been obtained.

Carrier (2 bytes)
The identifier of the Carrier/Operator the wireless modem is currently using

RSSI (2 bytes)
The received signal strength of the wireless modem in dBm. This value is signed in a 2's complement format.

Comm State (1 byte)
The current state of the wireless modem bit mapped as follows
Bit 0 – Available
Bit 1 – Network Service
Bit 2 – Data Service
Bit 3 – Connected (PPP Session Up)
Bit 4 – Voice Call is Active
Bit 5 – Roaming
Bit 6 – 3G Network (ie UMTS)
Bit 7 – Not Used

HDOP (1 byte)
The GPS Horizontal Dilution of Precision - it is a unit-less value reported with a 0.1 lsb.

Inputs (1 byte)
The current state of the inputs, bit mapped as follows:
Bit 0 – Ignition
Bit 1 – Input 1
Bit 2 – Input 2
Bit 3 – Input 3
Bit 4 – Input 4
Bit 5 – Input 5
Bit 6 – Input 6
Bit 7 – Input 7

Unit Status (1 byte)
Status of key modules within the unit:
Bit 0 – LMU32: HTTP OTA Update Status (0=Ok, 1=Error), LMU8: Unused
Bit 1 – GPS Antenna Status (0=OK, 1=Error)
Bit 2 – GPS Receiver Self-Test (0=OK, 1=Error) (LMU32 only)
Bit 3 – GPS Receiver Tracking (0=Yes, 1=No)
Bit 4 – Reserved, Currently Unused
Bit 5 – Reserved, Currently Unused
Bit 6 – Reserved, Currently Unused
Bit 7 – Unused

Accums (1 byte)
The number of 4-byte values in the AccumList and the Accumulator Reporting Format Type (upper 2 bits). Refer to Appendix G, 'Accumulator Reporting Formats' for details.

Spare (1 byte)
Not used. Set to 0

AccumList (4 bytes x 'n')
A list of 'n' 4-byte fields where 'n' is defined in the the Accums field. The format for this list is dependent upon the Accumulator Reporting Format Type also defined in the Accums field. Refer to Appendix G, 'Accumulator Reporting Formats' for details.

2.13 User Message with Accumulators (Message Type 9)

This type of User Message is used to pass user data to the Server from the Host serial port on the LMU including the current Accumulator values. The Host port must be configured to operate in this mode as well as set to use this message type over User Message (Message Type 4). The number of accumulators sent with each User Message is defined by the Inbound & Logged Event Report Contents. Note that the maximum number of user data bytes is 848. Also note that the number of accumulators reported might be automatically reduced for User Messages approaching the 848 byte limit. User Messages have the following structure:

	15	**0**
	Options Header (optional)	...
	...	...
	Message Header	...
	...	...
Present only when sent by LMU	Update Time (msByte)	Update Time
	Update Time	Update Time (lsByte)
	Time of Fix (msByte)	Time of Fix
	Time of Fix	Time of Fix (lsByte)
	Latitude (msByte)	Latitude
	Latitude	Latitude(lsByte)
	Longitude (msByte)	Longitude
	Longitude	Longitude (lsByte)
	Altitude (msByte)	Altitude
	Altitude	Altitude(lsByte)
	Speed (msByte)	Speed
	Speed	Speed (lsByte)
	Heading (msByte)	Heading (lsByte)
	Satellites	Fix Status
	Carrier (msByte)	Carrier (lsByte)
	RSSI (msByte)	RSSI (lsByte)
	Comm State	HDOP
	Inputs	Unit Status
	spare 1 (msByte)	spare 1 (lsByte)
	Accums	spare 2
	Accum List (msByte)	Accum List
	Accum List	Accum List (lsByte)
	...	...
	User Msg Route	User Msg ID
	User Msg Length (msByte)	User Msg Length (lsByte)
	User Msg (msByte)	…
	…	User Msg (lsByte)

User Message with Accumulators Format

Update Time (4 bytes)

The time tag of the message in seconds, referenced from Jan. 1, 1970

Time of Fix (4 bytes)
The last known time of fix from the GPS satellites. This value is reported in seconds from Jan. 1, 1970

Latitude (4 bytes)
The latitude reading of the GPS receiver measured in degrees with a 1x10-7 degree lsb, signed 2's complement.

Longitude (4 bytes)
The longitude reading of the GPS receiver measured in degrees with a 1x10-7 degree lsb, signed 2's complement.

Altitude (4 bytes)
The altitude reading of the GPS receiver measured in centimeters above the WGS-84 Datum, signed 2's complement.

Speed (4 bytes)
The speed as reported by the GPS receiver measured in centimeters per second

Heading (2 bytes)
The heading value reported in degrees from true North

Satellites (1 byte)
The number of satellites used in the GPS solution

Fix Status (1 byte)
The current fix status of the GPS receiver bitmapped as follows:
Bit 0 – Predicted

> Bit is set when the position update has a horizontal position accuracy estimate that is less that the Horizontal Position Accuracy Threshold defined in S-Register 142 (and the threshold is non-zero).

Bit 1 – Differentially Corrected

> This bit is set when WAAS DGPS is enabled (S-Register 139) and the position has been differentially

corrected.

Bit 2 – Last Known

This bit is set when the current GPS fix is invalid but a previous fix's value is available.

Bit 3 – Invalid Fix

This bit is set only after a power-up or reset before a valid fix is obtained.

Bit 4 – 2D Fix

This bit is set when 3 or fewer satellites are seen/used in the GPS fix. (i.e. with 3 satellites or less, an altitude value cannot be calculated)

Bit 5 – Historic

This bit is set when the message has been logged by the LMU.

Bit 6 – Invalid Time

This bit is set only after a power-up or reset before a valid time-sync has been obtained.

Carrier (2 bytes)
The identifier of the Carrier/Operator the wireless modem is currently using

RSSI (2 bytes)
The received signal strength of the wireless modem in dBm. This value is signed in a 2's complement format.

Comm State (1 byte)
The current state of the wireless modem bit mapped as follows
Bit 0 – Available
Bit 1 – Network Service
Bit 2 – Data Service
Bit 3 – Connected (PPP Session Up)
Bit 4 – Voice Call is Active
Bit 5 – Roaming
Bit 6 – 3G Network (ie UMTS)
Bit 7 – Not Used

HDOP (1 byte)
The GPS Horizontal Dilution of Precision - it is a unit-less value reported with a 0.1 lsb.

Inputs (1 byte)
The current state of the inputs, bit mapped as follows:
Bit 0 – Ignition
Bit 1 – Input 1
Bit 2 – Input 2
Bit 3 – Input 3
Bit 4 – Input 4
Bit 5 – Input 5
Bit 6 – Input 6
Bit 7 – Input 7

Unit Status (1 byte)
Status of key modules within the unit:
Bit 0 – LMU32: HTTP OTA Update Status (0=Ok, 1=Error), LMU8: Unused
Bit 1 – GPS Antenna Status (0=OK, 1=Error)
Bit 2 – GPS Receiver Self-Test (0=OK, 1=Error) (LMU32 only)
Bit 3 – GPS Receiver Tracking (0=Yes, 1=No)
Bit 4 – Reserved, Currently Unused
Bit 5 – Reserved, Currently Unused
Bit 6 – Reserved, Currently Unused
Bit 7 – Unused

Spare 1 (2 byte)
Not used. Set to 0

Accums (1 byte)
The number of 4-byte values in the AccumList and the Accumulator Reporting Format Type (upper 2 bits). Refer to Appendix G, 'Accumulator Reporting Formats' for details.

Spare 2 (1 byte)
Not used. Set to 0

AccumList (4 bytes x 'n')
A list of 'n' 4-byte fields where 'n' is defined in the Accums field. The format for this list is dependent upon the Accumulator Reporting Format Type also defined in the Accums field. Refer to Appendix G, 'Accumulator Reporting Formats' for details.

User Msg Route (1 byte)
This field was formerly the upper byte of the User Msg Type field. This new field defines which port the message originated from, or should be sent to: The possible values are:
0 – User0 (Host) Port
2 – User1 (Aux) Port
1,3-255 – Reserved for future use

User Msg ID (1 byte)
This field was formerly the lower byte of the User Msg Type field. This new field contains the user defined 'ID' value assigned to the User Message. The value of this field is defined by S-Register 136 or S-Register 166 depending on which port originated or is receiving the message.

User Msg Length (2 bytes)
The length, in bytes, of the user message that follows

User Msg (n Bytes)
The contents of the user message

2.14 Mini Event Report Message (Message Type 10)

Mini Event Report messages are initiated by the LMU and are generated by the LMU's Programmable Event Generator (PEG). They can be either Acknowledged or Unacknowledged Requests. The Server should respond to an Acknowledged Event Report Request with an Acknowledge Message (Message Type 1). Mini Event reports have the following structure:

15	0
Options Header (optional)	...
...	...
Message Header	...
...	...
Update Time (msByte)	Update Time
Update Time	Update Time (lsByte)
Latitude (msByte)	Latitude
Latitude	Latitude(lsByte)
Longitude (msByte)	Longitude
Longitude	Longitude (lsByte)
Heading (msByte)	Heading (lsByte)
Speed	FixStat/nSats
Comm/Unit Status	Inputs
Event Code	Accums
Accum List (msByte)	Accum List
Accum List	Accum List (lsByte)
...	...

Mini Event Report Message Format

Update Time (4 bytes)
The time tag of the message in seconds, referenced from Jan. 1, 1970

Latitude (4 bytes)
The latitude reading of the GPS receiver, measured in degrees with a 1x10-7 degree lsb, signed 2's complement

Longitude (4 bytes)
The longitude reading of the GPS receiver, measured in degrees with a 1x10-7 degree lsb, signed 2's complement

Heading (2 bytes)
The heading value reported in degrees from true North

Speed (1 bytes)
The speed as reported by the GPS receiver, measured in kilometers per hour

FixStat/nSats (1 byte)
The current fix status of the GPS receiver and the number of satellites used in the GPS solution bitmapped as follows:
Bit 0 thru Bit 3 – number of satellites (0-15)
Bit 4 – Invalid Time
Bit 5 – Invalid Fix
Bit 6 – Last Known
Bit 7 – Historic

Comm/GPS State (1 byte)
The current state of the wireless modem bit-mapped as follows
Bit 0 – Available
Bit 1 – Network Service
Bit 2 – Data Service
Bit 3 – Connected
Bit 4 – Active Voice Call
Bit 5 - Roaming
The current GPS status is bit-mapped as follows:
Bit 6 – GPS Antenna Status (0=OK, 1=Error)
Bit 7 – GPS Receiver Tracking (0=Yes, 1=No)

Inputs (1 byte)
The current state of the inputs, bit mapped as follows:
Bit 0 – Input 0 (Ignition)
Bit 1 – Input 1
Bit 2 – Input 2
Bit 3 – Input 3
Bit 4 – Input 4
Bit 5 – Input 5
Bit 6 – Input 6
Bit 7 – Input 7

Event Code (1 byte)
The event code assigned to the report as specified by the event's Action Parameter

Accums (1 byte)
The number of 4-byte values in the AccumList and the Accumulator Reporting Format Type (upper 2 bits). Refer to Appendix G, 'Accumulator Reporting Formats' for details.

AccumList (4 bytes x 'n')
A list of 'n' 4-byte fields where 'n' is defined in the Accums field. The format for this list is dependent upon the Accumulator Reporting Format Type also defined in the Accums field. Refer to Appendix G, 'Accumulator Reporting Formats' for details.

2.15 Mini User Message (Message Type 11)

This type of User Message is used to pass user data to the Server from a serial port on the LMU with minimal overhead. In place of the standard Status fields (Lat, Lon, Speed, etc.) only the Time Tag is included in this message. The LMU must be configured to use this message type over User Message (Message Type 4) or User Message with Accumulators (Message Type 9). (Ref. S-Register 177, bit-0). Note that the maximum number of user data bytes is 848. Mini User Messages have the following structure:

15	0
Options Header (optional)	...
...	...
Message Header	...
...	...
Update Time (msByte)	Update Time
Update Time	Update Time (lsByte)
User Msg Route	User Msg ID
User Msg Length (msByte)	User Msg Length (lsByte)
User Msg (msByte)	…
…	User Msg (lsByte)

Mini User Message Format

Update Time (4 bytes)
The time tag of the message in seconds, referenced from Jan. 1, 1970

User Msg Route (1 byte)
This field was formerly the upper byte of the User Msg Type field. This new field defines which port the message originated from, or should be sent to: The possible values are:
0 – User0 (Host) Port
2 – User1 (Aux) Port
1,3-255 – Reserved for future use

User Msg ID (1 byte)
This field was formerly the lower byte of the User Msg Type field. This new field contains the user defined 'ID' value assigned to the User Message. The value of this field is defined by S-Register 136 or S-Register

166 depending on which port originated or is receiving the message.

User Msg Length (2 bytes)
The length, in bytes, of the user message that follows

User Msg (n Bytes)
The contents of the user message

2.16 Mini App Message (Message Type 12)

This type of App Message is used to pass LMU application messages from the LMU to the server with minimal overhead. It functions the same as the Type 5 App Message but in place of the standard Status fields (Lat, Lon, Speed, etc.) only the Time Tag is included in this message when sent from the LMU. The LMU must be configured (S-Reg 177, bit-7 set) to use this message type in place of the standard Type 5 Application Msg for the LMU to Server direction. The maximum number of data bytes is 848. Mini Application Messages have the following structure:

15	**0**
Options Header (optional)	...
...	...
Message Header	...
...	...
Update Time (msByte)	Update Time
Update Time	Update Time (lsByte)
App Msg Type (msByte)	App Msg Type (lsByte)
App Msg Length (msByte)	App Msg Length (lsByte)
App Msg (msByte)	…
…	App Msg (lsByte)

Mini App Message Format

Update Time (4 bytes)
The time tag of the message in seconds, referenced from Jan. 1, 1970

App Msg Type (2 bytes)
The pre-defined 'type' value of the application message

App Msg Length (2 bytes)
The length, in bytes, of the application message that follows

App Msg (n Bytes)
The contents of the application message

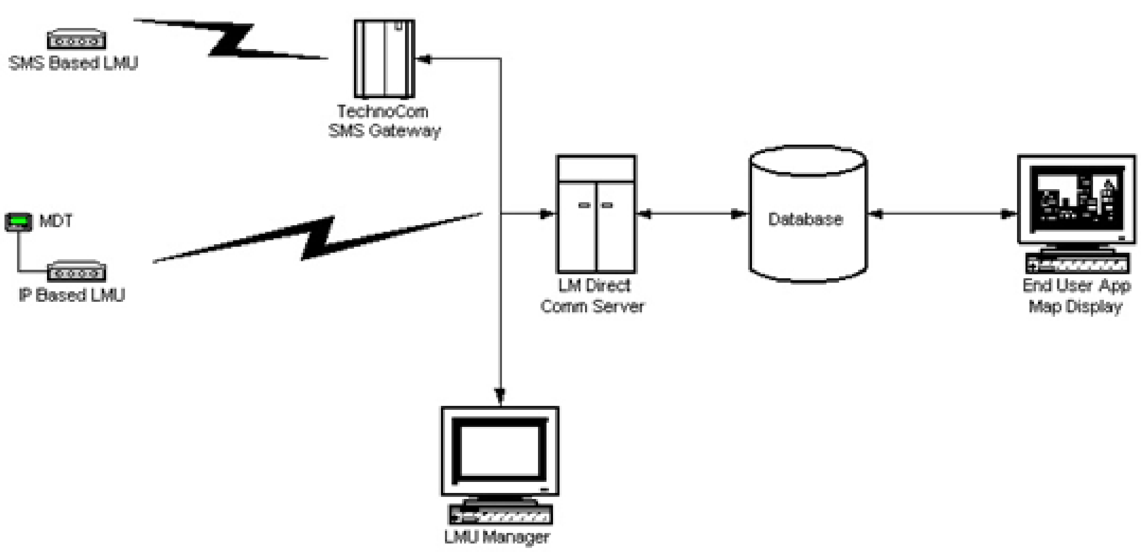
3 LM Direct™ Message Flows

3.1 Overview

This section describes the flow of various message transactions between the LMU and the backend system. For the purpose of these examples, it is assumed that the complete system comprises the following components:

- An LMU running on an IP based network (Mobile ID 999)
- A Mobile Data Terminal connected to LMU 999
- An LMU running on a Non-IP based network such as SMS (Mobile ID 777, Phone Number:888 555 1010)
- The CalAmp SMS Gateway which translates SMS messages to/from LM Direct™ messages
- A LM Direct™ Comm server
- A database (examples will be shown using SQL commands)
- A end user application/map display

A sample diagram of this system is shown below:



When needed, the examples below will assume the following:

- The current IP address of the IP Based LMU is 166.147.5.5
- The phone number of the SMS based LMU is 888 555 1010

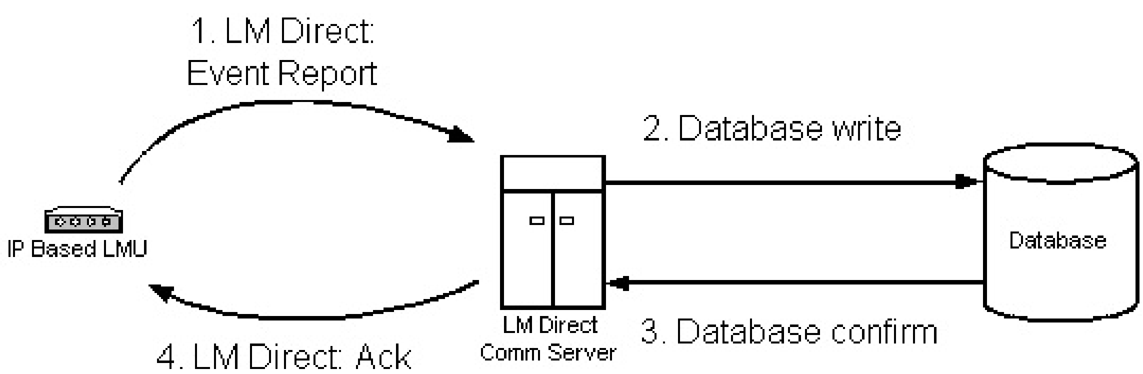
- The phone number of the SMS gateway is 999-555-1010
- The IP Address of the SMS gateway is 63.200.10.10
- The IP address of the LM Direct™ Comm Server is 63.200.10.11
- The IP address of LMU Manager is 63.200.10.12

3.2 Event Report Message Flow

3.2.1 Message Flow (Event Report)

This message flow description is centered on the IP based LMU sending Event Reports to the LM Direct™ Comm Server. It centers on the message transactions between the IP Based LMU, the LM Direct™ Comm Server and the Database.

The basic message flow is as follows:



1. The LMU creates an Event Report based on its PEG script and sends it to the LM Direct™ Comm Server
2. The LM Direct™ Comm Server receives, interprets and writes the contents of the Event Report to the database
3. The database confirms the write operation
4. The LM Direct™ Comm Server sends an Acknowledge Message (Message Type 1) to the LMU. This message is sent only if the Service Type of the event report is 0x01 (e.g., Acknowledged Request).

3.2.2 Message Contents (Event Report)

Shown below are two LM Direct™ messages that could be seen in the above sequence. Fields that are highlighted indicate values that must match between messages.

LM Direct™ Event Report	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x01
Message Type	0x02
Sequence Number	0x000A
Update Time	0x3FB54B33
Time of Fix	0x3FB54B33
Latitude	0x13B2953C
Longitude	0xBA18EBA3
Altitude	0x00000000
Speed	0x00000002
Heading	0x00D1
Satellites	0x04
Fix Status	0x00
Carrier	0x0004
RSSI	0xFFBF
Comm State	0x0F
HDOP	0x23
Inputs	0x07
Unit Status	0x00
Event Index	0x01
Event Code	0x01
Accums	0x00
Spare	0x00

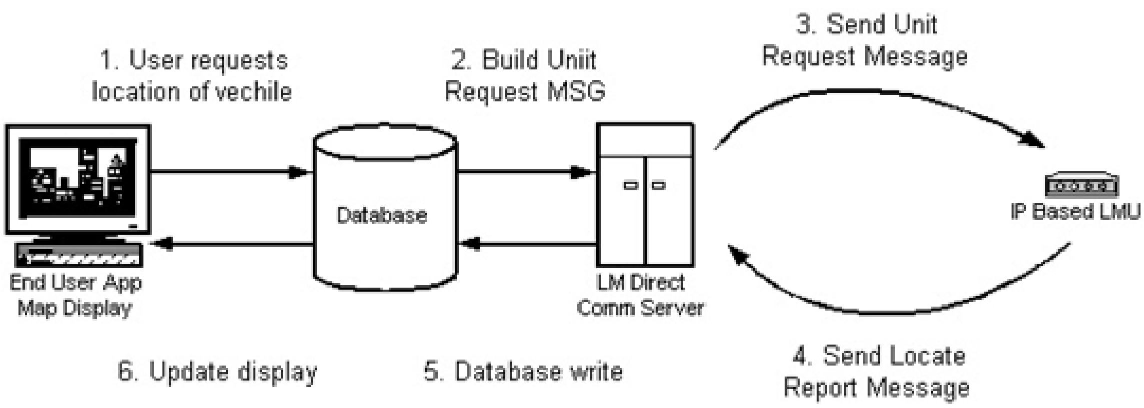
LM Direct™ ACK	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x02
Message Type	0x01
Sequence Number	0x000A
Type	0x02
ACK	0x00
Spare	0x00
App Version	0x000000

3.3 Unit Request Message Flow

3.3.1 Message Flow (Unit Request)

This message flow description is centered on the end-user application requesting the current location of the vehicle. In response, the database and LM Direct™ Comm Server should send a Unit Request Message (Message Type 7) with an action Type of 10 (Location Request) to the LMU in question. The LMU will respond with a Locate Report Message (Message Type 8).

The basic flow is as follows:



1. Using the map display, the user requests the current location of a vehicle
2. The database and LM Direct™ Comm Server build a Unit Request Message (Message Type 7) with a Locate Request Action
3. The LM Direct™ Comm Server sends the Unit Request Message (Message Type 7) to the LMU based on the last known IP address of the LMU. The database should be set up to store and provide this IP address when needed
4. The LMU returns a Locate Report Message (Message Type 8)
5. The LM Direct™ Comm Server receives, interprets and writes the contents of the Locate Report to the database
6. The Map Display is updated with the current local of the vehicle

3.3.2 Message Contents (Unit Request)

Shown below are two LM Direct™ messages that could be seen in the above sequence.

LM Direct™ Unit Request	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x01
Message Type	0x07
Sequence Number	0x0000
Action	0x0A
Data8	0x00
Data16	0x0000
Data32	0x00000000

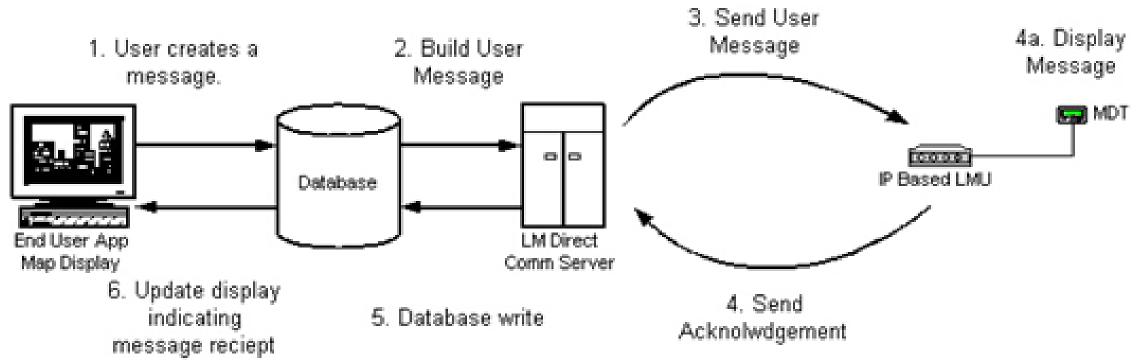
LM Direct™ Locate Report	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x02
Message Type	0x08
Sequence Number	0x0000
Update Time	0x3FB54B33
Time of Fix	0x3FB54B33
Latitude	0x13B2953C
Longitude	0xBA18EBA3
Altitude	0x00000000
Speed	0x00000002
Heading	0x00D1
Satellites	0x04
Fix Status	0x00
Carrier	0x0004
RSSI	0xFFBF
Comm State	0x0F
HDOP	0x23
Inputs	0x07
Unit Status	0x00
Accums	0x00
Spare	0x00

3.4 User Message Flow - Outbound

3.4.1 Message Flow (Outbound User Message)

This message flow description is centered on the end-user application attempting to send a message to the MDT attached to the IP based LMU's serial port. In response, the database and LM Direct™ Comm Server should build a User Message and pass it to the LMU in question. The LMU will respond with an Acknowledge Message (Message Type 1) indicating it attempted to deliver the message via the serial link. This assumes that **ATS136** is set to 18 (i.e. 0x12)

The basic flow is as follows:



1. The User creates a message to be sent to the driver of the vehicle via his/her Mobile Data Terminal (e.g., "Please call in")
2. The database and LM Direct™ Comm Server build a User Message message with they payload created by the User.
3. The LM Direct™ Comm Server sends the User Message (Message Type 4) to the LMU based on the last known IP address of the LMU. The database should be set up to store and provide this IP address when needed
4. The LMU returns an Acknowledgement indicating it received the message. At the same time the LMU will push the user message out its serial port to the MDT. Note: If the MDT is off or not connected this message will be lost.
5. The LM Direct™ Comm Server receives, and interprets the acknowledgement
6. The Map Display is updated to indicate the message was received

3.4.2 Message Contents (Outbound User Message)

Shown below are two LM Direct™ messages that could be seen in the above sequence:

LM Direct™ User Message	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x01
Message Type	0x04
Sequence Number	0x0000
User Message Route User Message ID	0x00 0x12
User Message Length	0x000F
User Message	0x506C6561 736C206361 6C6C20696E 0D

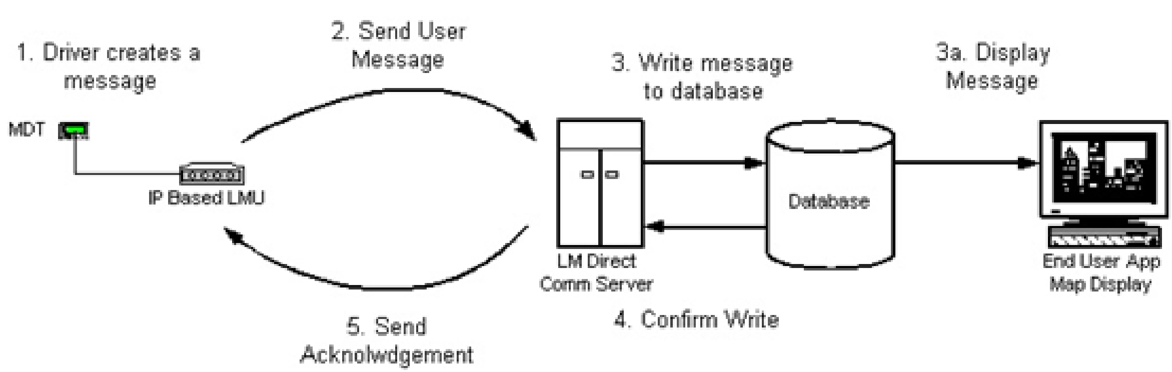
LM Direct™ ACK	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x02
Message Type	0x01
Sequence Number	0x0000
Type	0x04
ACK	0x00
Spare	0x00
App Version	0x363063

3.5 User Message Flow - Inbound

3.5.1 Message Flow (Inbound User Message)

This message flow description is centered on the driver sending a message via his/her MDT. In this case the IP base LMU will create a User Message (Message Type 4) and attempt to send it to the LM Direct™ Comm Server. Once the message has been successfully pushed into the database, the Comm Server should send an ACK to the LMU indicating it received the message. This assumes that **ATS136** is set to 18 (i.e. 0x12)

The basic message flow is shown below:



1. The Driver creates a message on his/her Mobile Data Terminal
2. The LMU packages and delivers this data as a User Message (Message Type 4) to the LM

Direct™ Comm Server
3. The LM Direct™ Comm Server receives, interprets and writes this data to the database. The driver's message is pushed to the End User Application/Map Display
4. The database indicates a successful write operation
5. The LM Direct™ Comm Server sends an Acknowledge Message (Message Type 1) to the LMU

3.5.2 Message Contents (Inbound User Message)

Shown below are two LM Direct™ messages that could be seen in the above sequence. Highlighted fields must match between messages:

LM Direct™ User Message	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x01
Message Type	0x04
Sequence Number	0x000B
Update Time	0x3FB54B40
Time of Fix	0x3FB54B40
Latitude	0x13B2953C
Longitude	0xBA18EBA3
Altitude	0x00000000
Speed	0x00000002
Heading	0x00D1
Satellites	0x04
Fix Status	0x00
Carrier	0x0004
RSSI	0xFFBF
Comm State	0x0F
HDOP	0x23
Inputs	0x07
Unit Status	0x00
User Message Route User Message ID	0x00 0x12
User Message Length	0x000F
User Message	0x476F696E67206F6E20 6C756E63680D

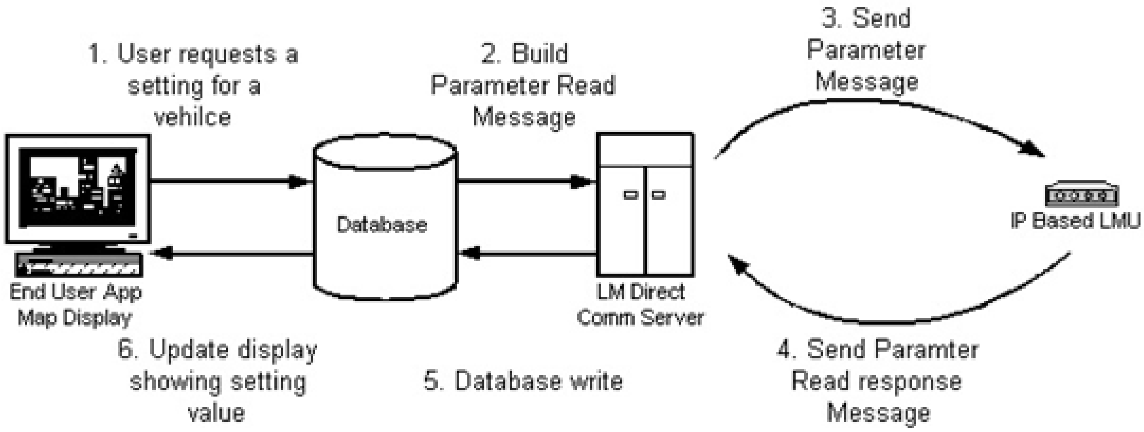
LM Direct™ ACK	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x02
Message Type	0x01
Sequence Number	0x000B
Type	0x04
ACK	0x00
Spare	0x00
App Version	0x000000

3.6 Parameter Message - Read

3.6.1 Message Flow (Read Param)

This message flow description is centered on the end-user application attempting to read one of the Parameters within an LMU (e.g., the Moving Speed Threshold).

The basic flow is as follows:



1. The user requests to read a specific value (or group of values) from the LMU
2. The Database and LM Direct™ Comm Server build a Parameter Read Request Message
3. The LM Direct™ Comm Server sends the Parameter Message (Message Type 6) to the LMU based on the last known IP address of the LMU. The database should be set up to store and provide this IP address when needed
4. The LMU sends a Parameter Read Response Message to the Comm Server
5. The Comm server receives, interprets and writes the contents of this message to the database
6. The Map Display is updated with the requested settings

3.6.2 Message Contents (Read Param)

Shown below are two LM Direct™ messages that could be seen in the above sequence. Highlighted fields must match between messages.

LM Direct™ Parameter Read Request	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x01
Message Type	0x06
Sequence Number	0x0000
Action	0x00
Parameter ID	0x040B
Length	0x0001
Parameter Index	0x00
Parameter ID	0x0000
Length	0x0000

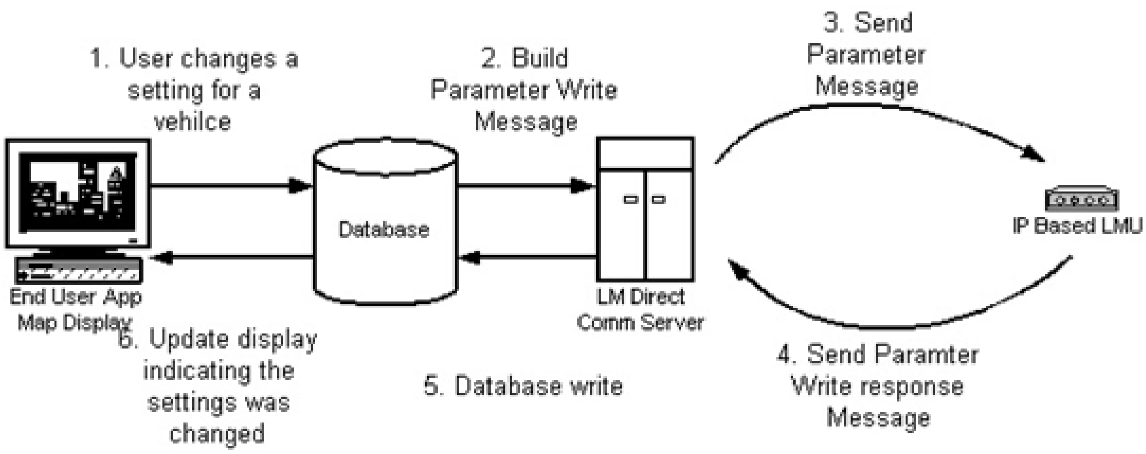
LM Direct™ Parameter Read Report	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x02
Message Type	0x06
Sequence Number	0x0000
Action	0x02
Parameter ID	0x040B
Length	0x03
Parameter Index	0x00
Data	0x0086
Parameter ID	0x0000
Length	0x0000

3.7 Parameter Message - Write

3.7.1 Message Flow (Write Param)

This message flow description is centered on the end-user application attempting to change one of the Parameters within an LMU (e.g., the Moving Speed Threshold).

The basic flow is as follows:



1. The user requests to change a specific value (or group of values) for the LMU
2. The Database and LM Direct™ Comm Server build a Parameter Write Request Message
3. The LM Direct™ Comm Server sends the Parameter Message (Message Type 6) to the LMU based on the last known IP address of the LMU. The database should be set up to store and provide this IP address when needed.
4. The LMU sends a Parameter Write Response Message to the Comm Server
5. The Comm server receives, interprets and writes the contents of this message to the database
6. The Map Display is updated indicating the setting(s) has been changed

3.7.2 Message Contents (Write Param)

Shown below are two LM Direct™ messages that could be seen in the above sequence. Highlighted fields must match between messages.

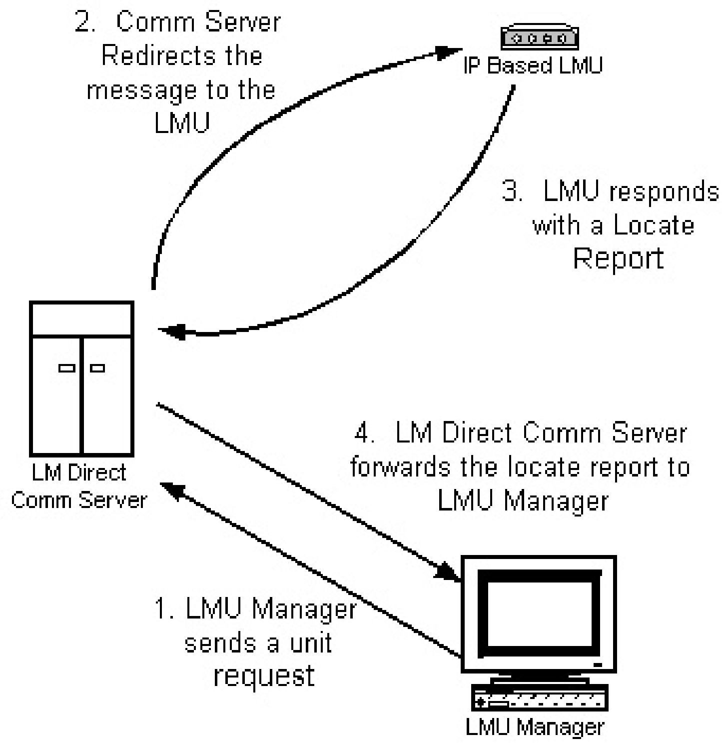
LM Direct™ Parameter Write Request	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x01
Message Type	0x06
Sequence Number	0x0000
Action	0x01
Parameter ID	0x040B
Length	0x0003
Parameter Index	0x00
Data	0x00B3
Parameter ID	0x0000
Length	0x0000

LM Direct™ Parameter Write Report	
Field	**Contents**
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x02
Message Type	0x06
Sequence Number	0x0000
Action	0x03
Parameter ID	0x0000
Length	0x0000

3.8 Message Forwarding

3.8.1 Message Flow (Forwarding)

This message flow description is centered on LMU Manager using the LM Direct™ Comm Server to forward a Unit Request Message (Message Type 7) to an LMU. In this sequence, the Comm server should be able to send the response back to LMU Manager.



1. LMU Manager creates a Unit Request Message (Message Type 7) intended for a specific LMU. LMU Manager sends this Unit Request with the forwarding feature enabled to the Inbound Port of the LM Direct™ Comm Server
2. The LM Direct™ Comm Server receives this message and swaps IP address data to send it to the LMU
3. The LMU Responds to the Comm Server with a Locate Report Message (Message Type 8)
4. Based on the forwarding information the LM Direct™ Comm Server sends the Locate Report to LMU Manager

3.8.2 Message Contents (Forwarding)

Shown below are four LM Direct™ messages that could be seen in the above sequence. Highlighted fields indicate matching values in each message:

- Comm Server IP: 63.200.10.11 = 0x3FC80A0B
- LMU Manager IP: 63.200.10.12 = 0x3FX80A0C
- LMU IP: 166.147.5.5 = 0xA6930505
- Port 20500 = 0x5014
- Port 20510 = 0x501E

LMU Manager to LM Direct™ Unit Request	
Field	**Contents**
Source Address	0x3FC80A0C
Source Port	0x5014
Destination Address	0x3FC80A0B
Destination Port	5014
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Forward Length	0x08
Forward Address	0xA6930505
Forward Port	0x501E
Forward Protocol Type	0x11
Forward Operation	0x00
Service Type	0x01
Message Type	0x07
Sequence Number	0x0000
Action	0x0A
Data8	0x00
Data16	0x0000
Data32	0x00000000

LM Direct™ to LMU Unit Request	
Field	**Contents**
Source Address	0x3FC80A0B
Source Port	5014
Destination Address	0xA6930505
Destination Port	0x501E
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Forward Length	0x08
Forward Address	0x3FC80A0C
Forward Port	0x5014
Forward Protocol Type	0x11
Forward Operation	0x00
Service Type	0x01
Message Type	0x07
Sequence Number	0x0000
Action	0x0A
Data8	0x00
Data16	0x0000
Data32	0x00000000

LMU to LM Direct™ Locate Report	
Field	**Contents**
Source Address	0xA6930505
Source Port	0x501E
Destination Address	0x3FC80A0B
Destination Port	5014
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Forward Length	0x08
Forward Address	0x3FC80A0C
Forward Port	0x5014
Forward Protocol Type	0x11
Fwd Operation	0x00
Service Type	0x02
Message Type	0x08
Sequence Number	0x0000
Update Time	0x3FB54B33
Time of Fix	0x3FB54B33
Latitude	0x13B2953C
Longitude	0xBA18EBA3
Altitude	0x00000000
Speed	0x00000002
Heading	0x00D1
Satellites	0x04
Fix Status	0x00
Carrier	0x0004
RSSI	0xFFBF
Comm State	0x0F
HDOP	0x23
Inputs	0x07
Unit Status	0x00
Accums	0x00
Spare	0x00

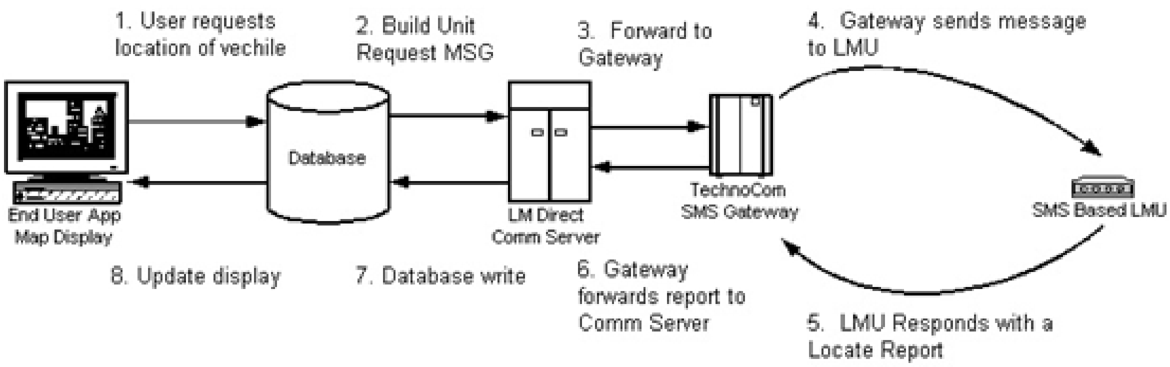
LM Direct™ to LMU Manager Locate Report	
Field	**Contents**
Source Address	0x3FC80A0B
Source Port	5014
Destination Address	0x3FC80A0C
Destination Port	0x5014
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Forward Length	0x08
Forward Address	0xA6930505
Forward Port	0x501E
Forward Protocol Type	0x11
Fwd Operation	0x00
Service Type	0x02
Message Type	0x08
Sequence Number	0x0000
Update Time	0x3FB54B33
Time of Fix	0x3FB54B33
Latitude	0x13B2953C
Longitude	0xBA18EBA3
Altitude	0x00000000
Speed	0x00000002
Heading	0x00D1
Satellites	0x04
Fix Status	0x00
Carrier	0x0004
RSSI	0xFFBF
Comm State	0x0F
HDOP	0x23
Inputs	0x07
Unit Status	0x00
Accums	0x00
Spare	0x00

3.9 Message Proxy - Outbound

3.9.1 Message Flow (Outbound Proxy)

An Outbound Proxy situation is a combination of a normal Unit Request Message (Message Type 7) and Message Forwarding. The Application is attempting to send a Unit Request Message (Message Type 7) to an LMU via the Comm Server; however the LMU is on a non-IP based network (e.g., SMS). The LMU's address must now be mapped to the non-IP based address (i.e., phone number). The proper mapping for an address will be defined in the Specifications for the CalAmp Gateway system. For the purposes of this example, the address is the LMU's phone number and is treated as a 10 digit number. This 10 digit number will be mapped into the Address and Port bytes of the Forwarding Header. It will be padded with 0s to fill all 6 bytes.

The basic message flow is as follows:



1. Using the map display the user requests the current location of a vehicle
2. The database and LM Direct™ Comm Server build a Unit Request Message (Message Type 7) with a Locate Request Action enabling the Proxy feature
3. The LM Direct™ Comm Server sends the Unit Request Message (Message Type 7) to the Gateway adding the LMU's non-IP based address to the Proxy field
4. The Gateway forwards the message to the LMU
5. The LMU responds with a Locate Report Message (Message Type 8)
6. The Gateway Receives the Locate Report Message (Message Type 8) and forwards it to the Comm Server
7. The Comm server receives, interprets and writes the contents of this message to the database
8. The Map Display is updated indicating the new position of the vehicle

3.9.2 Message Contents (Outbound Proxy)

Shown below are four LM Direct™ messages that could be seen in the above sequence. Highlighted fields indicate matching values in each message:

- Comm Server IP: 63.200.10.11 = 0x3FC80A0B
- Gateway IP: 63.200.10.10 = 0x3FX80A0A
- Gateway Phone Number: 9995551010 = 0x000253C80122
- LMU Phone Number: 8885551010 = 0x0002119EBFA2

- Port 20500 = 0x5014
- Port 20510 = 0x501E

LM Direct™ To GatewayUnit Request	
Field	**Contents**
Source Address	0x3FC80A0C
Source Port	0x5014
Destination Address	0x3FC80A0A
Destination Port	0x501E
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Forward Length	0x08
Proxy	0x0002119EBFA2
Forward Protocol Type	0x11
Forward Operation	0x00
Service Type	0x01
Message Type	0x07
Sequence Number	0x0000
Action	0x0A
Data8	0x00
Data16	0x0000
Data32	0x00000000

Gateway to LMU Unit Request	
Field	**Contents**
Source	0x000253C80122
Destination	0x0002119EBFA2
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Forward Length	0x08
Proxy	0x3FC80A0C 0x5014
Forward Protocol Type	0x11
Forward Operation	0x00
Service Type	0x01
Message Type	0x07
Sequence Number	0x0000
Action	0x0A
Data8	0x00
Data16	0x0000
Data32	0x00000000

LMU to Gateway Locate Report	
Field	**Contents**
Source	0x0002119EBFA2
Destination	0x000253C80122
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Proxy Length	0x08
Proxy	0x3FC80A0C 0x5014
Proxy Protocol Type	0x11
Forward Operation	0x00
Service Type	0x02
Message Type	0x08
Sequence Number	0x0000
Update Time	0x3FB54B33
Time of Fix	0x3FB54B33
Latitude	0x13B2953C
Longitude	0xBA18EBA3
Altitude	0x00000000
Speed	0x00000002
Heading	0x00D1
Satellites	0x04
Fix Status	0x00
Carrier	0x0004
RSSI	0xFFBF
Comm State	0x0F
HDOP	0x23
Inputs	0x07
Unit Status	0x00
Accums	0x00
Spare	0x00

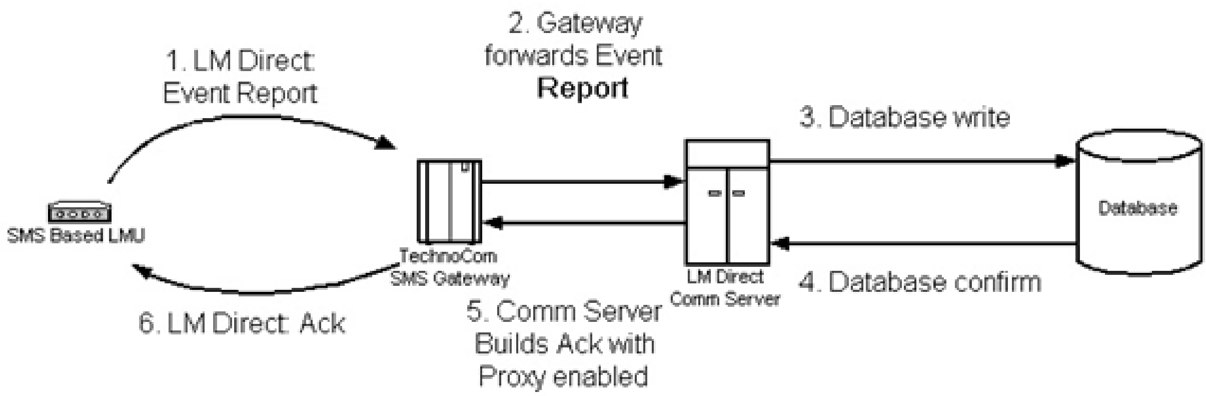
Gateway to LM Direct™ Locate Report	
Field	**Contents**
Source Address	0x3FC80A0A
Source Port	501E
Destination Address	0x3FC80A0C
Destination Port	0x5014
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Proxy Length	0x08
Proxy	0x0002119EBFA2
Proxy Protocol Type	0x11
Forward Operation	0x00
Service Type	0x02
Message Type	0x08
Sequence Number	0x0000
Update Time	0x3FB54B33
Time of Fix	0x3FB54B33
Latitude	0x13B2953C
Longitude	0xBA18EBA3
Altitude	0x00000000
Speed	0x00000002
Heading	0x00D1
Satellites	0x04
Fix Status	0x00
Carrier	0x0004
RSSI	0xFFBF
Comm State	0x0F
HDOP	0x23
Inputs	0x07
Unit Status	0x00
Accums	0x00
Spare	0x00

3.10 Message Proxy - Inbound

3.10.1 Message Flow (Inbound Proxy)

In an Inbound Proxy situation, a non-IP based LMU is attempting to deliver an Event Report Message (Message Type 2) to the LM Direct™ Comm Server via the CalAmp Gateway. This message flow assumes that the LMU is still using an Acknowledge Message (Message Type 1) so the LM Direct™ Comm Server must send an appropriate response (e.g., an ACK). In order to respond properly, the Comm Server must enable the Proxy settings on the outbound ACK.

The basic message flow is as follows:



1. The LMU creates an Event Report Message (Message Type 2) based on its PEG script and sends it to the CalAmp Gateway
2. The Gateway receives the message, adds the appropriate proxy information and forwards it to the Comm Server
3. The LM Direct™ Comm Server receives, interprets and writes the contents of the Event Report Message (Message Type 2) to the database
4. The database confirms the write operation
5. The LM Direct™ Comm Server sends an Acknowledge Message (Message Type 1) to the CalAmp Gateway with the appropriate Proxy information. Remember that this message is sent only if the Service Type of the event report is 0x01 (e.g., Acknowledged Request).
6. The Gateway forwards the ACK to the non-IP based LMU

3.10.2 Message Contents (Inbound Proxy)

Shown below are four LM Direct™ messages that could be seen in the above sequence. Highlighted fields indicate matching values in each message:

- Comm Server IP: 63.200.10.11 = 0x3FC80A0B
- Gateway IP: 63.200.10.10 = 0x3FX80A0A
- Gateway Phone Number: 9995551010 = 0x000253C80122
- LMU Phone Number: 8885551010 = 0x0002119EBFA2
- Port 20500 = 0x5014
- Port 20510 = 0x501E

LM Direct™ Event Report	
Field	**Contents**
Source	0x0002119EBFA2
Destination	0x000253C80122
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x01
Message Type	0x02
Sequence Number	0x000A
Update Time	0x3FB54B33
Time of Fix	0x3FB54B33
Latitude	0x13B2953C
Longitude	0xBA18EBA3
Altitude	0x00000000
Speed	0x00000002
Heading	0x00D1
Satellites	0x04
Fix Status	0x00
Carrier	0x0004
RSSI	0xFFBF
Comm Sta	0x0F
HDOP	0x23
Inputs	0x07
Unit Status	0x00
Event Index	0x01
Event Code	0x01
Accums	0x00
Spare	0x00

Gateway to LM Direct™ Comm Server Event Report	
Field	**Contents**
Source Address	0x3FX80A0A
Source Port	0x501E
Destination Address	0x3FC80A0B
Destination Port	0x5014
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Proxy Length	8
Proxy	0x0002119EBFA2
Proxy Protocol Type	0x11
Forward Operation	0x01
Service Type	0x02
Message Type	0x02
Sequence Number	0x000A
Update Time	0x3FB54B33
Time of Fix	0x3FB54B33
Latitude	0x13B2953C
Longitude	0xBA19EBA3
Altitude	0x00000000
Speed	0x00000002
Heading	0x00D1
Satellites	0x04
Fix Status	0x00
Carrier	0x0004
RSSI	0xFFBF
Comm State	0x0F
HDOP	0x23
Inputs	0x07
Unit Status	0x00
Event Index	0x01
Event Code	0x01
Accums	0x00
Spare	0x00

LM Direct™ to Gateway ACK	
Field	**Contents**
Source Address	0x3FC80A0B
Source Port	0x5014
Destination Address	0x3FX80A0A
Destination Port	0x501E
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0XF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Proxy Length	8
Proxy	0x0002119EBFA2
Proxy Protocol Type	0x11
Proxy Operation	0x01
Service Type	0x02
Message Type	0x01
Sequence Number	0x000A
Type	0x02
ACK	0x00
Spare	0x00
App Version	0x000000

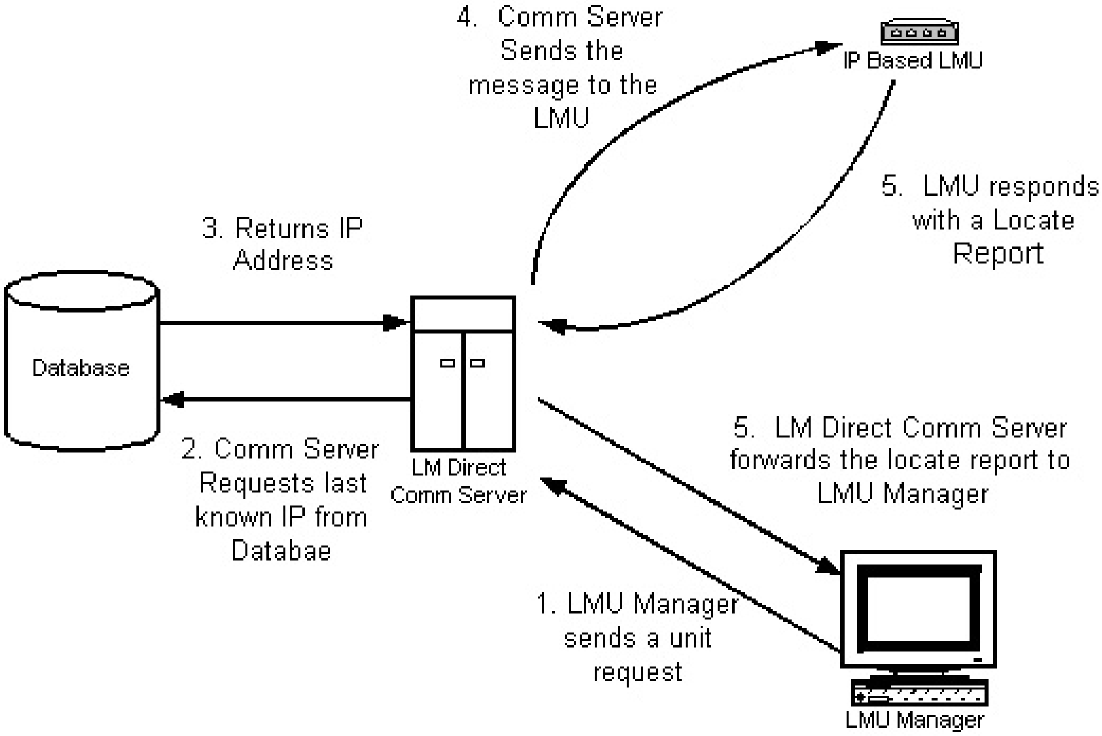
LM Direct™ ACK	
Field	**Contents**
Source	0x000253C80122
Destination	0x0002119EBFA2
Options Byte	0x83
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Service Type	0x02
Message Type	0x01
Sequence Number	0x000A
Type	0x02
ACK	0x00
Spare	0x00
App Version	0x000000

3.11 Message Forwarding With IP Lookup

3.11.1 Message Flow (Forwarding With IP Lookup)

Message forwarding with IP Lookup is similar to the basic Message Forwarding feature. The one key difference is that the LMU Manager program does not know the IP address of the destination LMU ahead of time; it only knows the Mobile ID. This type of request, therefore, requires that the Comm Server query the database for the last known IP address of the LMU in question. Assuming the IP is found, the Comm Server will then forward the message to the LMU. The incoming Locate Report Message (Message Type 8) should be handled as a normal Forwarding Message.

The basic message flow is as follows:



1. LMU Manager creates a Unit Request Message (Message Type 7) intended for a specific LMU. LMU Manager sends this Unit Request with the forwarding feature enabled to the Inbound Port of the LM Direct™ Comm Server
2. The LM Direct™ Comm Server receives this message and queries the data base for the IP address of the LMU
3. The database returns an IP address value to the Comm Server
4. The Comm Server sends the Unit Request Message (Message Type 7) to the LMU keeping the appropriate forwarding information
5. The LMU Responds to the Comm Server with a Locate Report Message (Message Type 8)
6. Based on the forwarding information the LM Direct™ Comm Server sends the Locate Report Message (Message Type 8) to LMU Manager

3.11.2 Message Contents (Forwarding With IP Lookup)

Shown below are four LM Direct™ messages that could be seen in the above sequence. Highlighted fields indicate matching values in each message.

- Comm Server IP: 63.200.10.11 = 0x3FC80A0B
- LMU Manager IP: 63.200.10.12 = 0x3FX80A0C
- LMU IP: 166.147.5.5 = 0xA6930505
- Port 20500 = 0x5014
- Port 20510 = 0x501E

LMU Manager to LM Direct™ Unit Request	
Field	**Contents**
Source Address	0x3FC80A0C
Source Port	0x5014
Destination Address	0x3FC80A0B
Destination Port	5014
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Forward Length	0x08
Forward Address	0x00000000
Forward Port	0x501E
Forward Protocol Type	0x11
Forward Operation	0x02
Service Type	0x01
Message Type	0x07
Sequence Number	0x0000
Action	0x0A
Data8	0x00
Data16	0x0000
Data32	0x00000000

LM Direct™ to LMU Unit Request	
Field	**Contents**
Source Address	0x3FC80A0B
Source Port	5014
Destination Address	0xA6930505
Destination Port	0x501E
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Forward Length	0x08
Forward Address	0x3FC80A0C
Forward Port	0x5014
Forward Protocol Type	0x11
Forward Operation	0x02
Service Type	0x01
Message Type	0x07
Sequence Number	0x0000
Action	0x0A
Data8	0x00
Data16	0x0000
Data32	0x00000000

LMU to LM Direct™ Locate Report	
Field	**Contents**
Source Address	0xA6930505
Source Port	0x501E
Destination Address	0x3FC80A0B
Destination Port	5014
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Forward Length	0x08
Forward Address	0x3FC80A0C
Forward Port	0x5014
Forward Protocol Type	0x11
Forward Operation	0x02
Service Type	0x02
Message Type	0x08
Sequence Number	0x0000
Update Time	0x3FB54B33
Time of Fix	0x3FB54B33
Latitude	0x13B2953C
Longitude	0xBA18EBA3
Altitude	0x00000000
Speed	0x00000002
Heading	0x00D1
Satellites	0x04
Fix Status	0x00
Carrier	0x0004
RSSI	0xFFBF
Comm State	0x0F
HDOP	0x23
Inputs	0x07
Unit Status	0x00
Accums	0x00
Spare	0x00

LM Direct™ to LMU Manager Locate Report	
Field	**Contents**
Source Address	0x3FC80A0B
Source Port	5014
Destination Address	0x3FC80A0C
Destination Port	0x5014
Options Byte	0x93
Mobid ID Length	0x02
Mobid ID	0xF999
Mobid ID Type Length	0x01
Mobid Type	0x01
Forward Length	0x08
Forward Address	0xA6930505
Forward Port	0x501E
Forward Protocol Type	0x11
Forward Operation	0x02
Service Type	0x02
Message Type	0x08
Sequence Number	0x0000
Update Time	0x3FB54B33
Time of Fix	0x3FB54B33
Latitude	0x13B2953C
Longitude	0xBA18EBA3
Altitude	0x00000000
Speed	0x00000002
Heading	0x00D1
Satellites	0x04
Fix Status	0x00
Carrier	0x0004
RSSI	0xFFBF
Comm State	0x0F
HDOP	0x23
Inputs	0x07
Unit Status	0x00
Accums	0x00
Spare	0x00

4 Appendix A — Using Mobile ID

The Mobile ID and Mobile ID Type fields in the Options Header can be used on both mobile to server messages (e.g., Event Reports) and Server to mobile messages (e.g., Parameter Requests). For wireless networks with dynamic IP addressing schemes, the Mobile ID is required for mobile to Server messages to provide the Server with the Mobile ID of the mobile unit. For Server to mobile messages, Mobile ID is optional, though it is recommended. For each message direction, the Mobile ID serves a slightly different purpose.

On mobile to Server messaging, the Mobile ID in the Options Header provides the static identifier (or name) of the unit. On Server to mobile messaging the Mobile ID is used as a filter; that is, the LMU or TetheredLocator will compare its own mobile ID and type with that of the incoming message. If they match, the message is processed; if neither match, the message is discarded. If either of these fields is not present in the Options Header, they will not be considered in the match process.

A specific scenario where the Server to mobile filter would be used is as follows:

1. LMU 1 registers to a network and gets IP address 166.10.20.30
2. LMU 1 sends in an event report with Mobile ID 5555
3. LMU 1 deregisters from the network thereby releasing its IP address
4. LMU 2 registers to the network and gets IP 166.10.20.30
5. The Server software sends a dispatch request to Mobile ID 5555
6. LMU 2 receives dispatch request for Mobile ID 5555

In this case, if the Mobile ID filter was not in place, LMU 2 would have received instructions meant for LMU 1. Of course, this could cause major problems and confusions in fleet management operations.

5 Appendix B — Using Authentication

The Authentication field in the Options Header allows the LMU to reject Server to mobile messages (e.g., Parameter Requests) based on an authentication key. The key is generated by an exclusive or operation between the LM Direct™ Interface's password and a 4 byte number based on the Mobile ID. This 4 byte number is created by taking the lower 9 digits of the Mobile ID.

If Mobile ID reporting has been disabled, the authentication key is generated by an exclusive or between the LM Direct™ Interface password and the current IP address of the LMU. The authentication feature is enabled and disabled based on the LM Direct™ Interface password. Authentication is enabled if the password is equal to or greater than 0x00000001. It is disabled if the password is 0x00000000.

5.1 Mobile ID Enabled Example

If the Mobile ID is 2957085360282 and the LM Direct™ Interface Password has been set to 21.12.21.12 then our Authentication key is created as follows:

1. Take the lower 9 digits of the mobile ID

Dec: 085360282
Hex: 0x5167E9A

2. Convert the password to a hexadecimal number

    Dotted Dec: 21.12.21.12
    Hex: 0x150C150C

3. XOR these two values to get the authentication key

    Hex: 0x101A6B96

5.2 Mobile ID Disabled Example

If the current IP Address of the LMU is 166.147.64.158 and the LM Direct™ Interface Password has been set to 21.12.21.12 then the Authentication key is created as follows:

1. Take the IP Address of the LMU

    Dotted Dec: 166.147.64.158
    Hex: 0xA693409E

2. Take the unit's password

    Dotted Dec: 21.12.21.12
    Hex: 0x150C150C

3. XOR these two values to get the authentication key

    Hex: 0x B39F5592

6 Appendix C — Processing the Options Header

The following pseudo-code snippet illustrates one possible way of processing the Mobile ID information from the Options Header of an LM Direct™ message.

Our basic steps are as follows:

1. Check the options byte to see if there is header information to process
2. Check each bit in the options byte and process the data accordingly

```
/*Check the Options Byte of our incoming packet
 *remember bit 7 will be set if there
 *are options to process
```

```
 */
IF (PacketIN->Options & 0x80) THEN
{
/* Assign a Pointer to the top of the packet
 */
       ptrPacket = &PacketIN->Options;
       /*  Create an offset to walk through the packet
        */
       packetOffset = 0;
       /* We look at each bit of the Options Byte
        * to determine which values we need to extract.
       .* We will walk through each bit using a left shift
        * bitwise operator
        */
       FOR (Options_Bit = 1; Options_Bit 128; OptionsBit <<= 1)
       {
               /*  Check if the options bit has been set.
                */
               IF (PacketIN->Options & Options_Bit) THEN
               {
                       /* move our offset to the next byte to process
*/
                       packetOffset ++;
                       /* Process Mobile ID
                       */
IF (Options_Bit == 1) THEN
{
/* BCDtoASCII (Destination, Source, Length)
 */
       BCDtoASCII (MyPacket->MobileID,
        &ptrPacket[packetOffSet+1],
        ptrPacket[PacketOffset])
} ENDIF
                               /*  Process Mobile ID Type
                                */

IF (Options_Bit == 2) THEN
                               {
                               /* NOTE Mobile ID Type Length is
                                * always 1 byte.
                               */
       MyPacket->MobileIDType =  ptrPacket[packetOffset + 1];
                               } ENDIF
                               /*  Process Authentication field
                               */
IF (Options_Bit == 4) THEN
                               {
                               /* NOTE Authentication Length is
                                * always 4 bytes.
                               */
       MyPacket->Auth =
ptrPacket[packetOffset + 1];
                               } ENDIF
                               /*  move the Offset past the length of data
                                */
                               packetOffset += ptrPacket[packetOffset];
                       }  ENDIF
               } END LOOP
} END IF
/*  Set index to the beginning of the location data
 */
packetOffset ++
```

7 Appendix D — Routing Field and Forwarding Field Definitions and Uses

The Routing Field can be included in a request made to an LMU to assist the Server or programs in routing the response from the LMU to the appropriate destination. For example, if an Application has several user terminals, the routing field can be used to identify the terminal making the request so that the response returned to the Application can be forwarded to the requesting terminal. The Routing field is optional and is not required for normal, single source application communications with the LMU. It is entirely up to the Application to define the use of the Routing field. Up to 8 bytes are available, which is sufficient to include an IP address and port number as a means of identifying the message source.

The Forwarding field is similar to the Routing field except that it has a specific purpose and formats. Like the Routing field, the Forwarding field included in a request from the Application is returned, byte-for-byte, in the Options header of the response sent back from the LMU. This field has two basic uses, Forward and Proxy.

The Forward Operation is intended for use by a forwarding agent located within the Application that can accept packets from another program such as LMU Manager and forward them to the LMU through the IP/UDP port already in use by the Application. Forwarding is often necessary to circumvent routing issues created by firewalls put in place by the wireless carriers. With these firewalls, typically only messages originated by the address/port destination of an LMU inbound message is permitted to pass through to the LMU. In other words, the only parties that can send to a given LMU are the parties that are currently receiving messages from the LMU. For the Forwarding operation this field comprises four sub fields: A 4-byte address followed by a 2-byte port followed by a 1-byte protocol type followed by a one byte Operation Type.

The Proxy Operation is intended to be used in conjunction with a CalAmp Gateway program. Gateways are programs that will take packets from an IP based network and forward them to LMUs on non-IP based networks (e.g. the 900MHz Point to Multi-Point LMUs). For the Proxy operation this field comprises three sub fields: A 6-byte address followed by a 1-byte protocol type followed by a one byte Operation Type.

8 Appendix E — Application Message Types and Formats

Application Message (Message Type 5) are a class of messages that are used for management of the LMU and for special functions supported by the Server. This section defines the Application Message types and formats. Each of the Application Message formats listed are contained within an Application Message (Message Type 5).

8.1 Types

Types	
IP Request	10
IP Report	11
Time Sync	50
Download ID Report	100
Download Authorization	101
Download Request	102
Download Update	103
Download Complete	104
Download, HTTP, LMU FW	105
Download, HTTP, File	106
OTA Download	107
AT Command	110
Version Report	111
GPS Status Report	112
Message Statistics Report	113
State Report	115
Geo-Zone Action Message	116
Geo-Zone Update Message	117
Capture Report	120
Motion Log Report	122
VBus Data Report	130
Vehicle ID Report	131
VBus DTC Report	132

8.2 Formats

8.2.1 IP Request

This message is supported by the Server to allow external programs to request the Server provide the IP Address and UDP Port number from which the LMU last reported. The message contains the Mobile ID of the LMU. The Server will respond with an Application Message ‘IP Report’. The format for this message is TBD.

8.2.2 IP Report

Sent by the Server in response to receiving an IP Request Application Message from an external program. This message provides the external program the IP Address and UDP Port number from which the requested LMU last reported. The format for this message is TBD.

Used by the LMU to request time synchronization from the server (Request/Response byte equals 1) and used by the server to send time to the LMU (Request/Response byte equals 2). The 'Time' field is a 4-byte UTC (GMT) time (1-sec LSB, 0=12:00AM Jan 1, 1970). The 'Spare fields should be set to 0. The 'Time' field is set 0 by the LMU when sending a request.

15	**0**
Request (1) / Response (2)	Spare
Spare	Spare
Time(msByte)	Time
Time	Time(lsByte)

Time Sync App Message Format

8.2.3 Time Sync

Used by the LMU to request time synchronization from the server (Request/Response byte equals 1) and used by the server to send time to the LMU (Request/Response byte equals 2). The 'Time' field is a 4-byte UTC (GMT) time (1-sec LSB, 0=12:00AM Jan 1, 1970). The 'Spare fields should be set to 0. The 'Time' field is set 0 by the LMU when sending a request.

15	**0**
Request (1) / Response (2)	Spare
Spare	Spare
Time(msByte)	Time
Time	Time(lsByte)

Time Sync App Message Format

8.2.4 Download App Messages (100 through 106)

These messages are reserved for use in downloading a new program into the LMU.

8.2.5 OTA Download App Message (107)

The OTA Download App message is sent by the downloading service to inform the LMU that an Over-the-Air download file is available for the unit. This App message describes the identity and location of that file.

The message is a collection of null-terminated strings, which have a ‘key:value’ structure.

DEV:<n>
FILE:<n>
FLEN:<n>
VER:<s>
PROT:<s>
SRVR:<s:n>
PATH:<s>
UNAM:<s>
PWD:<s>

DEV
The device ID of the file to be downloaded. The device ID can refer to the LMU itself or the other field upgradeable devices that are connected to the LMU.

Device Type	Value
LMU	0
JPOD	1
OBDII	2
Reserved (TC Radio)	3

Example:

DEV:2<Null Byte>

FILE
The ID of the file to be downloaded.

File Type	Value
Firmware	0
Configuration	1
Bootloader	2
System Manager	3
DB	4

Example:

FILE:2<Null Byte>

FLEN
The length of the file to be updated in bytes.

Example:

FLEN:13418<Null Byte>

VER
The version of the file as a string.

Example:

VER:2.0.2<Null Byte>

PROT
The ID of ther protocol to use when retrieving the file.

Download Protocol	ID
UDP	0
HTTP	1
FTP	2

Example:

PROT:HTTP1<Null Byte>

SRVR
The name of the server, with an optional port, to connect to download the file.

Example:

SRVR:puls.calamp.com:80<Null Byte>

PATH
The full path to the location of the file on the server

Example:

PATH:dnld/ota<Null Byte>

UNAM (As necessary)
The username to use when connecting to various protocols, for example, FTP

PWD (As necessary)
The password to use with the username above

LMU Response – HTTP OTA transfers
The LMU uses this method to construct a URL to the file.
For HTTP requests, the LMU constructs query parameters based on the values in the message. The following is an example of the query parameters :

?esn=4471001004&DeviceType=0&FileType=1&Ver=FileVersion

Where 'esn' is the CalAmp ESN.

And a full URL will look like:
http://puls.calamp.com/dnld?esn=4471001004&DeviceType=0&FileType=1&Ver=FileVersion

8.2.6 AT Command

Used by Server to send an AT Command Set|AT Command to the LMU. Currently, the only AT Command accepted on this interface is a "SENDTO" command. This App Message holds the AT Command string in the 'Msg' field. The 'Msg Length' field contains the 2-byte length of the AT Command string. The 'Msg Type' field should have a value of 110. The LMU will respond with an Acknowledge Message (Message Type 1).

8.2.7 Version Report

Sent by the LMU in response to a Server initiated Unit Request Message (Message Type 7) for a Version Report. This App Message holds the version string in the 'Msg' field. The 'Msg Length' field contains the 2-byte length of the version string. The 'Msg Type' field should have a value of 111.

8.2.8 GPS Status Report

Sent by the LMU in response to a Server initiated Unit Request Message (Message Type 7) for a GPS Report. This App Message reports the current GPS output information which may be different from what is reported through normal report messages. It also contains the PRN, Azimuth, Elevation, SNR and usage indication for each satellite being tracked by LMU's GPS receiver. The 'App Msg Type' field should have a value of 112 and the 'App Msg Length' field should be 124.

15	0
Time of Fix (msByte)	Time of Fix
Time of Fix	Time of Fix (lsByte)
Latitude (msByte)	Latitude
Latitude	Latitude(lsByte)
Longitude (msByte)	Longitude
Longitude	Longitude (lsByte)
Altitude (msByte)	Altitude
Altitude	Altitude(lsByte)
Speed (msByte)	Speed
Speed	Speed (lsByte)
Heading (msByte)	Heading (lsByte)
Satellites	Fix Status
HDOP	AGC
Antenna Status	GPS Power
The following fields repeat for each of 16 possible satellites.	
PRN	Used
Elevation	SNR
Azimuth(msByte)	Azimuth (lsbyte)

GPS Status App Message Format

Time of Fix (4 bytes)
The time of fix from the GPS receiver. This value is reported in seconds from Jan. 1, 1970

Latitude (4 bytes)
The latitude reading from the GPS receiver measured in degrees with a 1x10^-7 degree lsb, signed 2's complement.

Longitude (4 bytes)
The longitude reading from the GPS receiver measured in degrees with a 1x10^-7 degree lsb, signed 2's complement.

Altitude (4 bytes)
The altitude reading from the GPS receiver measured in centimeters above the WGS-84 Datum, signed 2's complement.

Speed (4 bytes)
The speed as reported by the GPS receiver, measured in centimeters per second

Heading (2 bytes)
The heading value reported in degrees from true North

Satellites (1 byte)
The number of satellites used in the GPS fix solution

Fix Status (1 byte)
The current fix status of the GPS receiver bitmapped as follows:
Bit 0 – Predicted

> Bit is set when the position update has a horizontal position accuracy estimate that is less that the Horizontal Position Accuracy Threshold defined in S-Register 142 (and the threshold is non-zero).

Bit 1 – Differentially Corrected

> This bit is set when WAAS DGPS is enabled (S-Register 139) and the position has been differentially corrected

Bit 2 – Last Known

> This bit is set when the current GPS fix is invalid but a previous fix's value is available.

Bit 3 – Invalid Fix

> This bit is set only after a power-up or reset before a valid fix is obtained.

Bit 4 – 2D Fix

> This bit is set when 3 or fewer satellites are seen/used in the GPS fix. (i.e. with 3 satellites or less, an altitude value cannot be calculated)

Bit 5 – Historic

> This bit is set when the message has been logged by the LMU.

Bit 6 – Invalid Time

> This bit is set only after a power-up or reset before a valid time-sync has been obtained.

HDOP (1 byte)
The GPS Horizontal Dilution of Precision: It is a unit-less value reported with a 0.1 lsb.

AGC (1 byte)
The current gain level of the AGC in the GPS receiver - Represented in percent (0-100). 0=not supported

Antenna Status (1 byte)
Electrical status of the GPS Antenna:
0=Active Antenna On and OK
1=Open Circuit
2=Short Circuit
3=Active Antenna off
4=Passive Antenna
5=Unknown
6-255 – Unused
GPS Power (1 byte)
0=GPS Receiver is Off
Nonzero = GPS Receiver is On

The following fields repeat for each of 16 possible satellites:

PRN (1 byte)
Identification number of the satellite, 1-32, 0 if this satellite set of fields is not used

Used (1 byte)
Indicates if this satellite is used in the location fix calculation, ASCII 'U'(0x55)=yes, ASCII <space> (0x20)=no

Elevation (1 byte)
Elevation of satellite above the horizon, degrees 0-90,

SNR (1 byte)
Satellite receive signal strength (C/No) in dBHz, 0-99;

Azimuth (2 bytes)

Azimuth of satellite from true North, degrees 0-359

8.2.9 Message Statistics Report

Sent by the LMU in response to a Server initiated Unit Request Message (Message Type 7) for a Message Statistics Report. This App Message reports various message counters in the LMU. The 'App Msg Type' field should have a value of 113 and the 'App Msg Length' field should be 30 bytes.

15	**0**
LocReq Count (msByte)	LocReq Count (lsByte)
Received User Msgs(msByte)	Received User Msgs (lsByte)
Inbound Reports (msByte)	Inbound Reports (lsByte)
Sent User Msgs (msByte)	Sent User Msgs (lsByte)
Spare1 (msByte)	Spare 1 (lsByte)
Logged Records (msByte)	Logged Records
Logged Records	Logged Records (lsByte)
Spare2 (msByte)	Spare2
Spare2	Spare2 (lsByte)
Spare3 (msByte)	Spare3
Spare3	Spare3 (lsByte)
Spare4 (msByte)	Spare4
Spare4	Spare4 (lsByte)
Spare5 (msByte)	Spare5
Spare5	Spare5 (lsByte)

Message Statistics App Message Format

LocReq Count (2 bytes)

The number of Locate Requests received by the LMU since the last power-up, wakeup, or stats reset

Received User Msgs (2 bytes)
The number of User Messages received by the LMU since the last power-up, wakeup, or stats reset

Inbound Reports (2 bytes)
The number of Inbound Messages (i.e., Event Reports) sent by the LMU since the last power-up, wakeup or stats reset

Sent User Msgs (2 bytes)
The number of User Messages sent by the LMU since the last power-up, wakeup, or stats reset.

SPARE1 (2 bytes)
This field is reserved for future use

Logged Records (4 bytes)
This field is the current number of Event Reports and User Messages stored in the LMU's log

SPARE2 (4 bytes)
This field is reserved for future use

SPARE3 (4 bytes)
This field is reserved for future use

SPARE4 (4 bytes)
This field is reserved for future use

SPARE5 (4 bytes)
This field is reserved for future use

8.2.10 State Report

15	0
Zone States (msByte)	Zone States
Zone States	Zone States (lsByte)
Environment State (msByte)	Environment State
Environment State	Environment State (lsbyte)
PEG Flags (msByte)	PEG Flags (lsByte)
Input States	Output States
Spare 1 (msByte)	Spare 1 (lsByte)
Spare 2	Accumulator Count
Accumulator 0 (msByte)	Accumulator 0
Accumulator 0	Accumulator 0 (lsByte)
…	…
Accumulator 15 (msByte)	Accumulator 15
Accumulator 15	Accumulator 15 (lsByte)

LMU State Report Format

Zone States (32 byte)

This is a bit mapped field indicating if the LMU is either inside (bit set) or outside (bit cleared) of a defined zone. If the zone is undefined, the bit is also cleared. The mapping is as follows:

```
Bit 0          Set = Inside Zone 0           Cleared = Outsize Zone 0
Bit 1          Set = Inside Zone 1           Cleared = Outsize Zone 1
Bit 2          Set = Inside Zone 2           Cleared = Outsize Zone 2
Bit 3          Set = Inside Zone 3           Cleared = Outsize Zone 3
…
Bit 30         Set = Inside Zone 30          Cleared = Outsize Zone 30
Bit 31         Set = Inside Zone 31          Cleared = Outsize Zone 31
```

Environment State (32 byte)

This is a bit mapped field indicating the environment state of the LMU. The bit mapping is as follows:

```
Bit 0           Set = PEG Flag 0 set          Cleared = Peg Flag 0 cleared
Bit 1           Set = PEG Flag 1 set          Cleared = Peg Flag 1 cleared
Bit 2           Set = PEG Flag 2 set          Cleared = Peg Flag 2 cleared
Bit 3           Set = PEG Flag 3 set          Cleared = Peg Flag 3 cleared
Bit 4           Set = PEG Flag 4 set          Cleared = Peg Flag 4 cleared
Bit 5           Set = PEG Flag 5 set          Cleared = Peg Flag 5 cleared
Bit 6           Set = PEG Flag 6 set          Cleared = Peg Flag 6 cleared
Bit 7           Set = PEG Flag 7 set          Cleared = Peg Flag 7 cleared
Bit 8           Set = Inside Zone 0           Cleared = Outsize Zone 0
Bit 9           Set = Inside Zone 1           Cleared = Outsize Zone 1
Bit 10          Set = Inside Zone 2           Cleared = Outsize Zone 2
Bit 11          Set = Inside Zone 3           Cleared = Outsize Zone 3
Bit 12          Set = Inside Zone 4           Cleared = Outsize Zone 4
Bit 13          Set = Inside Zone 5           Cleared = Outsize Zone 5
Bit 14: spare
Bit 15: spare
Bit 16          Set = Above Speed 0           Cleared = Below Speed 0
Bit 17          Set = Above Speed 1           Cleared = Below Speed 1
Bit 18          Set = Above Speed 2           Cleared = Below Speed 2
Bit 19          Set = Above Speed 3           Cleared = Below Speed 3
Bit 20          Set = Day-of-Week 0 True      Cleared = Day-of-Week 0 False
Bit 21          Set = Day-of-Week 1 True              Cleared = Day-of-Week 1 False
Bit 22          Set = Day-of-Week 2 True              Cleared = Day-of-Week 2 False
Bit 23          Set = Day-of-Week 3 True      Cleared = Day-of-Week 3 False
Bit 24          Set = Ignition On             Cleared = Ignition Off
Bit 25          Set = Input 1 High            Cleared = Input 1 Low
Bit 26          Set = Input 2 High            Cleared = Input 2 Low
Bit 27          Set = Input 3 High            Cleared = Input 3 Low
Bit 28          Set = GPS Acquired            Cleared = No GPS
Bit 29          Set = Comm Acquired           Cleared = Comm Lost
Bit 30          Set = Log Active              Cleared = Log Inactive
Bit 31          Set = Moving                  Cleared = Not Moving
```

PEG Flags (2 byte)

This is a bit mapped field indicating the state of each of the available PEG Flags. The bit mapping is as follows:

```
Bit 0           Set = PEG Flag 0 set          Cleared = Peg Flag 0 cleared
Bit 1           Set = PEG Flag 1 set          Cleared = Peg Flag 1 cleared
Bit 2           Set = PEG Flag 2 set          Cleared = Peg Flag 2 cleared
Bit 3           Set = PEG Flag 3 set          Cleared = Peg Flag 3 cleared
Bit 4           Set = PEG Flag 4 set          Cleared = Peg Flag 4 cleared
Bit 5           Set = PEG Flag 5 set          Cleared = Peg Flag 5 cleared
Bit 6           Set = PEG Flag 6 set          Cleared = Peg Flag 6 cleared
Bit 7           Set = PEG Flag 7 set          Cleared = Peg Flag 7 cleared
Bit 8           Set = PEG Flag 8 set          Cleared = Peg Flag 8 cleared
Bit 9           Set = PEG Flag 9 set          Cleared = Peg Flag 9 cleared
Bit 10          Set = PEG Flag 10 set         Cleared = Peg Flag 10 cleared
Bit 11          Set = PEG Flag 11 set         Cleared = Peg Flag 11 cleared
Bit 12          Set = PEG Flag 12 set         Cleared = Peg Flag 12 cleared
Bit 13          Set = PEG Flag 13 set         Cleared = Peg Flag 13 cleared
Bit 14          Set = PEG Flag 14 set         Cleared = Peg Flag 14 cleared
Bit 15          Set = PEG Flag 15 set         Cleared = Peg Flag 15 cleared
```

Input States (1 byte)

This is a bit-mapped field indicating the state of each of the LMU's available input lines, including those on the ioPOD. It does not apply to the A/D inputs. The bit mapping is as follows:

```
Bit 0           Set = Ignition On              Cleared = Ignition Off
Bit 1           Set = Input 1 High             Cleared = Input 1 Low (Or Used as Output)
Bit 2           Set = Input 2 High             Cleared = Input 2 Low (Or Used as Output)
Bit 3           Set = Input 3 High             Cleared = Input 3 Low
Bit 4           Set = Input 4 High             Cleared = Input 4 Low
Bit 5           Set = Input 5 High             Cleared = Input 5 Low
Bit 6           Set = Input 6 High             Cleared = Input 6 Low
Bit 7           Set = Input 7 High             Cleared = Input 7 Low
```

Outputs (1 bytes)

This is a bit-mapped field indicating the state of each of the LMU's available outputs, including those on the ioPOD. The bit mapping is as follows:

```
Bit 0           Set = Output 0 Set             Cleared = Output 0 Cleared
Bit 1           Set = Output 1 Set             Cleared = Output 1 Cleared (Or used as Input)
Bit 2           Set = Output 2 Set             Cleared = Output 2 Cleared (Or used as Input)
Bit 3           Set = Output 3 Set             Cleared = Output 3 Cleared
Bit 4           Set = Output 4 Set             Cleared = Output 4 Cleared
Bit 5           Set = Output 5 Set             Cleared = Output 5 Cleared
Bit 6           Set = Output 6 Set             Cleared = Output 6 Cleared
Bit 7: spare
```

Spare 1 (2 bytes)

This field is not used. It should be set to 0

Spare 1 (1 byte)

This field is not used. It should be set to 0

Accums (1 byte)

The number of 4-byte values in the AccumList and the Accumulator Reporting Format Type (upper 2 bits). Refer to Appendix G, 'Accumulator Reporting Formats' for details.

AccumList (4 bytes x ‘n’)
A list of ‘n’ 4-byte fields where ‘n’ is defined in the Accums field. The format for this list is dependent upon the Accumulator Reporting Format Type also defined in the Accums field. Refer to Appendix G, ‘Accumulator Reporting Formats’ for details.

8.2.11 Geo-Zone Action Message

15	0
Action Code	Response Code
Block Number	Spare
Start Index (msByte)	Start Index (lsByte)
Record Count (msByte)	Record Count (lsByte)
Major Version	Minor Version

Geo-Zone Action Message Format

Action Code (1 byte)
The value of this field indicates the action being performed (or requested) of the Geo-Zone memory space. The following values are defined:
0 – No Action
1 – Erase
2 – Read Records
3 – Update Complete
4 – Write Record (response only)
5 – Read Version
6 – Write Version
7 – 255 –Unknown/Reserved

Response Code (1 byte)
The value of this field indicates any error condition reported during one of the Actions. The following values are defined:
0 – No Error
1 – Write Verification Failure (resend previous request)
2 – Write Error
3 – Invalid Length on Update Requests or Read Requests
4 – Block Error on Update Request or Read Request
5 – 255 – Unknown/Reserved

Block Number (1 byte)

The memory block in which the Geo-Zone action is taking place. This value should normally be 0.

Spare (1 byte)
Reserved for future use. Set to 0.

Start Index (2 byte)
This field contains the record index (location) of the first record Requested or 0 for all other Actions

Record Count (2 byte)
The meaning of this field will change depending on the Action code. The actions that contain a value other than 0 are:

- Read Version: The Number of records contained in the LMU(response only)
- Read Records: The Number of records to return in the update message

Major Version (1 byte)
The value of this field contains an unsigned integer value indicating the major version of the Geo-Zone configuration. Versions are optional, but are available to assist in tracking changes to sets of configured Geo-Zones. For example version 1.5 indicates the current configured Geo-Zone set is under Major Version 1. This field is only valid for action types 6 and 7.

Minor Version (1 byte)
The value of this field contains an unsigned integer value indicating the minor version of the Geo-Zone configuration. Versions are optional, but are available to assist in tracking changes to sets of configured Geo-Zones. For example version 1.5 indicates the current configured Geo-Zone set is under Minor Version 5. This field is only valid for action types 6 and 7.

8.2.12 Geo-Zone Update Message

15 **0**

Block Number	Record Count
Start Index (msByte)	Start Index (lsByte)
Geo-Zone 0 Type	Geo-Zone 0 ID
Geo-Zone 0 Range (msByte)	Geo-Zone 0 Range (lsByte)
Geo-Zone 0 Latitude (msByte)	Geo-Zone 0 Latitude
Geo-Zone 0 Latitude	Geo-Zone 0 Latitude (lsByte)
Geo-Zone 0 Longitude (msByte)	Geo-Zone 0 Longitude
Geo-Zone 0 Longitude	Geo-Zone 0 Longitude (lsByte)
…	…
Geo-Zone N Type	Geo-Zone N ID
Geo-Zone N Range (msByte)	Geo-Zone N range (lsByte)
Geo-Zone N Latitude (msByte)	Geo-Zone N Latitude
Geo-Zone N Latitude	Geo-Zone N Latitude (lsByte)
Geo-Zone N Longitude (msByte)	Geo-Zone N Longitude
Geo-Zone N Longitude	Geo-Zone N Longitude (lsByte)

Geo-Zone Update Message Format

Block Number (1 byte)
The memory block in which the Geo-Zone action is taking place. This value should normally be 0.

Record Count (1 byte)
This contains the number of records included in this message. This field has a maximum value of 70.

Start Index (2 byte)
This field contains the record index (location) of the first record in this message

Geo-Zone Type (1 byte)
This field contains two values, the type of geo-zone point being defined and the Super Group to which the

Geo-Zone point belongs. The type value is contained in bits 0 – 3. The super group ID is contained in bits 4 – 7. The available settings are:

```
Bits 0-3: Geo-Zone Type
0               reserved
1               PointZone
2               Polygon
3               Route
4-14            spare
15              reserved
Bits 4-5 Super-group ID
0-3             Super Group ID
Bits 6-7 Reserved
```

Geo Zone ID (1 byte)
The value of this field is the reference number used for reporting Geo-Zone states within PEG. This field may range from 0 – 255. Note that point for a geo-zone must be grouped together and in order for route and polygon definitions. Zone IDs, must also be grouped under the Geo-Zone Super Group ID.

Geo Zone Range (2 bytes)
The value of this field will change based on the type of Geo-Zone being used.
For Point Zones, this value defined the radius of the Point Zone.
For Routes it defines the width of the route's corridor (eg the width of the road).
This value is not used for Polygon Geo-Zones.
The field value can range from 0 – 65535 meters.

Geo-Zone Latitude (4 bytes)
The latitude value of the Geo-Zone point being defined measured in degrees with a 1x10^-7 degree lsb, signed 2's complement

Geo-Zone Longitude (4 bytes)
The longitude of the Geo-Zone point being defined measured in degrees with a 1x10^-7 degree lsb, signed 2's complement

NOTE: A single update message can contain up to 70 Geo-Zone point definitions.

8.2.13 Capture Message

The Capture Message is sent by the LMU in response to the Flush Capture Buffer PEG Action. The Capture Message uses an Application Message (Message Type 5) Type of 120 and may contain one or more

Capture Messages.

15	**0**
Timestamp (msByte)	Timestamp
Timestamp	Timestamp (lsByte)
Latitude (msByte)	Latitude
Latitude	Latitude(lsByte)
Longitude (msByte)	Longitude
Longitude	Longitude (lsByte)
Accumulator N (msByte)	Accumulator N (lsByte)
Accumulator N+1(msByte)	Accumulator N+1 (lsByte)
…	
Accumulator N+X (msByte)	Accumulator N+X (lsByte)

Repeats for Each Capture Report in the Capture Message

Capture Report Message Format (2 Byte Accumulators)

15	**0**
Timestamp (msByte)	Timestamp
Timestamp	Timestamp (lsByte)
Latitude (msByte)	Latitude
Latitude	Latitude(lsByte)
Longitude (msByte)	Longitude
Longitude	Longitude (lsByte)
Accumulator N (msByte)	Accumulator N
Accumulator N	Accumulator N (lsByte)
Accumulator N+1(msByte)	Accumulator N+1 (lsByte)
Accumulator N+1	Accumulator N+1 (lsByte)
…	
Accumulator N+X (msByte)	Accumulator N+X
Accumulator N+X	Accumulator N+X (lsByte)

Repeats for Each Capture Report in the Capture Message

Capture Report Message Format (4 Byte Accumulators)

Timestamp(4 bytes)

The timestamp from the GPS receiver when the capture report was created. This value is reported in seconds from Jan. 1, 1970

Latitude (4 bytes)

The latitude reading from the GPS receiver measured in degrees with a 1x10-7 degree lsb, signed 2's complement.

Longitude (4 bytes)
The longitude reading from the GPS receiver measured in degrees with a 1x10-7 degree lsb, signed 2's complement.
Accumulator N to Accumulator N+X (2 bytes or 4 bytes)
The value of Accumulator N using either a 2 byte or 4 byte width. The accumulator width is defined by the S-Register 171 setting of the LMU. N and X are defined by the Flush Capture Buffer Action Modifier. Please refer to the PEG Programming Guide for details.

8.2.14 Motion Log Report

This App message is used by the LMU to report the contents of either the Short Motion Log or the Long Motion Log. The payload includes a sub-header section describing the type and number of records that follow. The 'Msg Type' field should have a value of 122.

Record Type
Group Count
Message Count
Spare (0)
Record Count (MsByte)
Record Count (LsByte)
Sample Interval (MsByte)
Sample Interval (LsByte)
Time-Date (MsByte)
Time-Date
Time-Date
Time-Date (LsByte)

Sub-Header

Record Type (1 byte)
This field identifies whether this report contains Short Motion Log records, a value of 2, or Long Motion Log records, a value of 1. All other values are reserved.

Group Count (1 byte)
This field contains the number of Motion Log Reports within this group of Motion Log Reports required to deliver the desired contents of the Long or Short Log buffers. See Message Count.

Message Count (1 byte)

This field contains the count of this Motion Log Report within the Group of Motion Log Reports required to deliver the desired contents of the Long or Short Log buffer. See Group Count.

Spare (1 byte)
Set to zero (0).

Record Count (2 bytes)
This field contains the number of count of Long or Short Log records contained in this Motion Log Report.

Sample Interval (2 bytes)
This field represents the time between record samples for the Long or Short Log records contained in this Motion Log Report. If the Record Type is Long ('1') then the sample interval is in seconds. If the Record Type is Short ('2') then the sample interval is in milliseconds.

Time-Date (4 bytes)
The device system time at the time this report was generated. This value is reported in seconds from Jan. 1, 1970

Short Motion Log
The Short Motion Log buffer will hold 34 24-byte messages, providing a history of 8.5 seconds since the GPS receiver updates normally occur every 250 mS (4Hz). New updates overwrite the oldest updates in the Short Log buffer so the contents of the buffer always represents the most recent 34 updates. Note that the Time-of-Fix for the Short Log is based on the number of mS in the day and does not include date information.

Latitude (MsByte)
Latitude
Latitude
Latitude (LsByte)
Longitude (MsByte)
Longitude
Longitude
Longitude (LsByte)
Speed (MsByte)
Speed (LsByte)
Heading
nSats FixStat
Time-of-Fix (MsByte)
Time-of-Fix
Time-of-Fix
Time-of-Fix (LsByte)
Accel-Longitudinal (MsByte)
Accel-Longitudinal (LsByte)
Accel-Lateral (MsByte)
Accel-Lateral (LsByte)
Accel-Vertical (MsByte)
Accel-Vertical (LsByte)
TimeTick (MsByte)
TimeTick (LsByte)

Short Motion Log Report

Latitude (4 bytes)
The latitude reading of the GPS receiver, measured in degrees with a 1x10-7 degree lsb, signed 2's complement

Longitude (4 bytes)
The longitude reading of the GPS receiver, measured in degrees with a 1x10-7 degree lsb, signed 2's complement

Speed (2 bytes)
The speed as reported by the GPS receiver, measured in centimeters per second

Heading (1 byte)
The heading value reported in 1.40625 degrees increments (360/256) from true North

nSats_FixStat (1 byte)
This field contains two values. The 4 least significant bits (Bits 0-3) represent the satellite count for this GPS reading (0-15, >15 is reported as 15). The most significant 4 bits represent the Fix Status of this GPS reading with each bit representing the following:
Bit 4 – Predicted

> Bit is set when the position update has a horizontal position accuracy estimate that is less that the Horizontal Position Accuracy Threshold defined in S-Register 142 (and the threshold is non-zero).

Bit 5 – Differentially Corrected

> This bit is set when SBAS DGPS is enabled (S-Register 139) and the position has been differentially corrected

Bit 6 – Dead-Reckoning (not supported)

> This bit indicates of the current GPS fix is a result of dead-reckoning calculations instead of a true GPS position determination.

Bit 7 – Invalid Fix

> This bit indicates the reported fix is invalid and should be discarded or disregarded.

Time of Fix (4 bytes)
The field is generated by the GPS receiver and represents the time-of-day the GPS fix was made. This field has an LSB of 1 mS.

Acceleration-Longitudinal (2 bytes)
The field represents the acceleration of the vehicle along its longitudinal axis (forwards/backwards) as measured by the accelerometer and calibrated by the device software. The signed value is scaled to be cm/s/s.

Acceleration-Lateral (2 bytes)
The field represents the acceleration of the vehicle along its lateral axis (side-to-side) as measured by the accelerometer and calibrated by the device software. The signed value is scaled to be cm/s/s.

Acceleration-Vertical (2 bytes)

The field represents the acceleration of the vehicle along its vertical axis (up/down) as measured by the accelerometer and calibrated by the device software. The signed value is scaled to be cm/s/s.

Tick (2 bytes)

The field represents the least 16-bits of the devices internal clock. It has a 1mS resolution.

Long Motion Log

The Long Motion Log will store up to 96 records, and allows updates to be received from the GPS receiver that are 'n' or more seconds old. 'n' is a 16-bit configurable parameter (Param 1043, Index 0) defined by the user to control how often update are stored on the Long Motion Log. 'n' can have a value of 0 through 65535 seconds (default = 1). If zero (0), the Long Motion Log will be updated at the same rate as the Short Motion Log (typically 4Hz). Note that the Time-of-Fix value for the Long Log includes date information since it is defined as the number of seconds since Jan 1 1970.

Latitude (MsByte)
Latitude
Latitude
Latitude (LsByte)
Longitude (MsByte)
Longitude
Longitude
Longitude (LsByte)
Speed (MsByte)
Speed (LsByte)
Heading
nSats FixStat
TimeDate-of-Fix (MsByte)
TimeDate-of-Fix
TimeDate-of-Fix
TimeDate-of-Fix (LsByte)

Sub-Header

Latitude (4 bytes)

The latitude reading of the GPS receiver, measured in degrees with a 1x10^-7 degree lsb, signed 2's complement

Longitude (4 bytes)

The longitude reading of the GPS receiver, measured in degrees with a 1x10^-7 degree lsb, signed 2's complement

Speed (2 bytes)
The speed as reported by the GPS receiver, measured in centimeters per second

Heading (1 byte)
The heading value reported in 1.40625 degrees increments (360/256) from true North

nSats_FixStat (1 byte)
This field contains two values. The 4 least significant bits (Bits 0-3) represent the satellite count for this GPS reading (0-15, >15 is reported as 15). The most significant 4 bits represent the Fix Status of this GPS reading with each bit representing the following:
Bit 4 – Predicted

> Bit is set when the position update has a horizontal position accuracy estimate that is less that the Horizontal Position Accuracy Threshold defined in S-Register 142 (and the threshold is non-zero).

Bit 5 – Differentially Corrected

> This bit is set when SBAS DGPS is enabled (S-Register 142) and the position has been differentially corrected

Bit 6 – Dead-Reckoning (not supported)

> This bit indicates of the current GPS fix is a result of dead-reckoning calculations instead of a true GPS position determination.

Bit 7 – Invalid Fix

> This bit indicates the reported fix is invalid and should be discarded or disregarded.

TimeDate of Fix (4 bytes)
The field represents the time and date of the reported GPS fix. This value is reported in seconds from Jan. 1, 1970

8.2.15 VBus Data Report

Sent by the LMU upon receipt of data from the Vehicle Bus capture device. This App message is used to package the VBus data for consumption by the server. Its format and usage is dependent upon the script running on the VBus capture device. The ‘Msg Type’ field should have a value of 130.

8.2.16 Vehicle ID Report

Sent by the LMU to report the Vehicle specific information related to the Vehicle Bus Interface. This message is typically used to report the VIN (Vehicle ID Number) of the vehicle, the vehicle interface bus type and a list of parameters detected and supported by the vehicle. The message is formatted using multiple null-terminated strings, each beginning with a Tag:

VIN:<17-char VIN><NULL Char>
PROTO:<OBD-II-Protocol ID><NULL Char>
PARAMS:<Comma Separated Supported Parameter IDs><NULL Char>
INDCTRS:<Supported Indicator IDs><NULL Char>

VIN
17-character string

Example:

VIN:1G1JC5444R7252367<NULL Char>

OBD II Protocol Identifier
0 - None
1 - J1850VPW
2 - J1850PWM
3 - ISO9141-2
5 - KWP2000
6 - CAN 11 bit
7 - CAN 29 bit

Example:

PROTO:7<NULL Char>

Supported Parameters and Indicators
Supported Parameter ID’s and Indicator ID’s are formatted as a string of decimal ID numbers separated by commas. For Indicators, the Indicator ID is immediately followed in parenthesis “(…)” by the bit map of the indicator members supported by the vehicle.

Example:

PARAMS:1,3,5,6,10,11,12<NULL Char>
INDCTRS:0(000001101100011),1(0000001100011001)<NULL Char>

Note: In some vehicles VIN may not be available (see Supported Vehicles/parameters list). In this case the VIN string will be empty.

8.2.17 VBus DTC Report

This App message is used by the LMU to report either all Diagnostic Trouble Codes (DTCs) or previously unreported (DTCs) currently being generated on the vehicle bus. The payload for this report is a list of 5-byte DTC codes preceded by the DTC Report Type field (0='all' DTCs, 1='unreported' DTCs) followed by the number of DTCs included in the report.

DTC Report Type
Number DTCs Reported
1st DTC Code (MSB)
1st DTC Code
1st DTC Code
1st DTC Code
1st DTC Code (LSB)
2nd DTC Code (MSB)
2nd DTC Code
2nd DTC Code
2nd DTC Code
2nd DTC Code (LSB)
…
nth DTC Code (MSB)
nth DTC Code
nth DTC Code
nth DTC Code
nth DTC Code (LSB)

9 Appendix F — Geo-Zone Messaging

9.1 Using Geo-Zones

Unlike most configuration items within the LMU, the Geo-Zone definitions are stored in Flash memory. From an external application stand point, this means that a specific sequence of events must be followed in order to read or write geo-zones.

For Polygon definitions, geo-zone point ordering is also important. Points must be listed in either clock-wise or counter clockwise order as you travel around the perimeter of the polygon. The first point definition of the polygon is also used as the last. Polygons points must appear in successive definitions, for instance, if you have a 6 point polygon and a 12 point polygon, the first 6 records must be polygon #1, the next 12 must be polygon #2. Polygon zones cannot share points between each other.

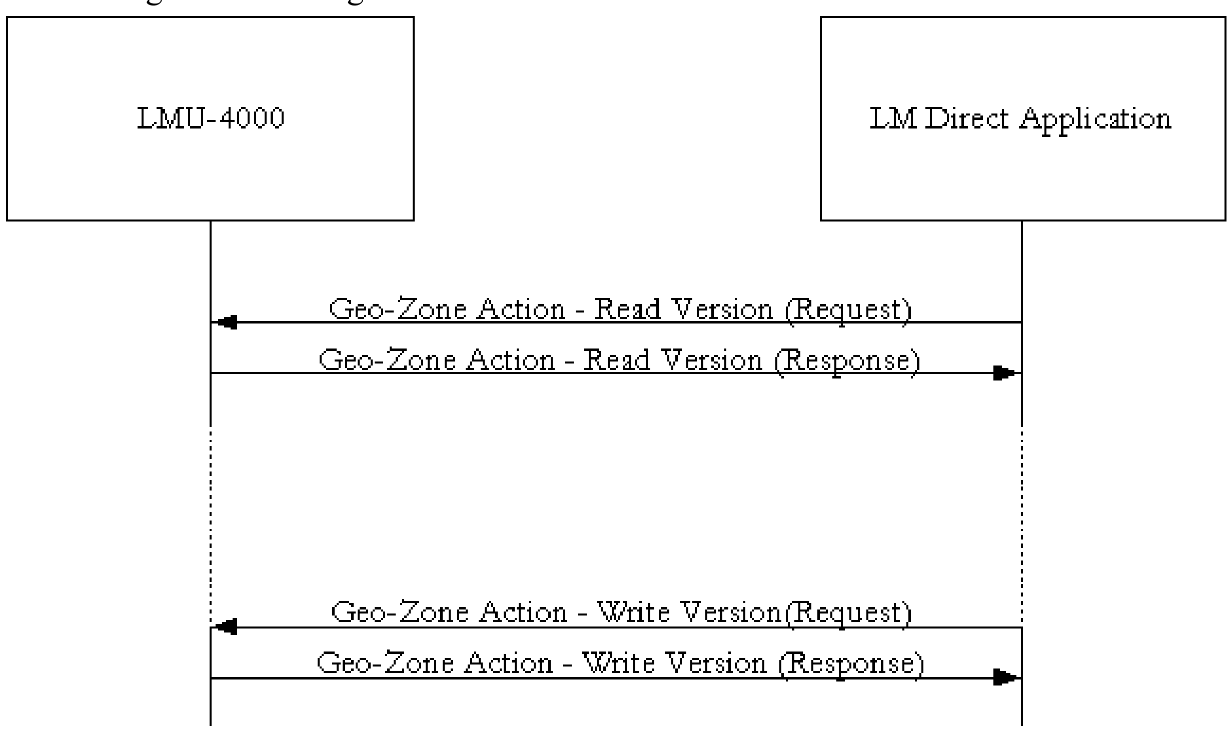
Point zones, since they are stand-alone entities may appear in any order and in any combination. Keep in mind that they cannot appear in the middle of a polygon definition as that would violate the above rule.

The LMU can store up to 5400 Geo-Zone records.

9.1.1 Geo-Zone Read & Write Version Requests

To read the current version of the Geo-Zone settings, an LM Direct™ Application will send a Geo-Zone Action Message with the Action Type set to 5 (Read Version Request). The LMU will respond with another Action message containing Action Type 7 as well as populating the Major and Minor version fields.
To write a new version the LM Direct™ App will send a Geo-Zone Action Message of type 6 (Write Request) remembering to populate the Major and Minor version fields with the new version information. The LMU will respond with an Action message of type 4 (general response).

Both message flows are diagramed below:

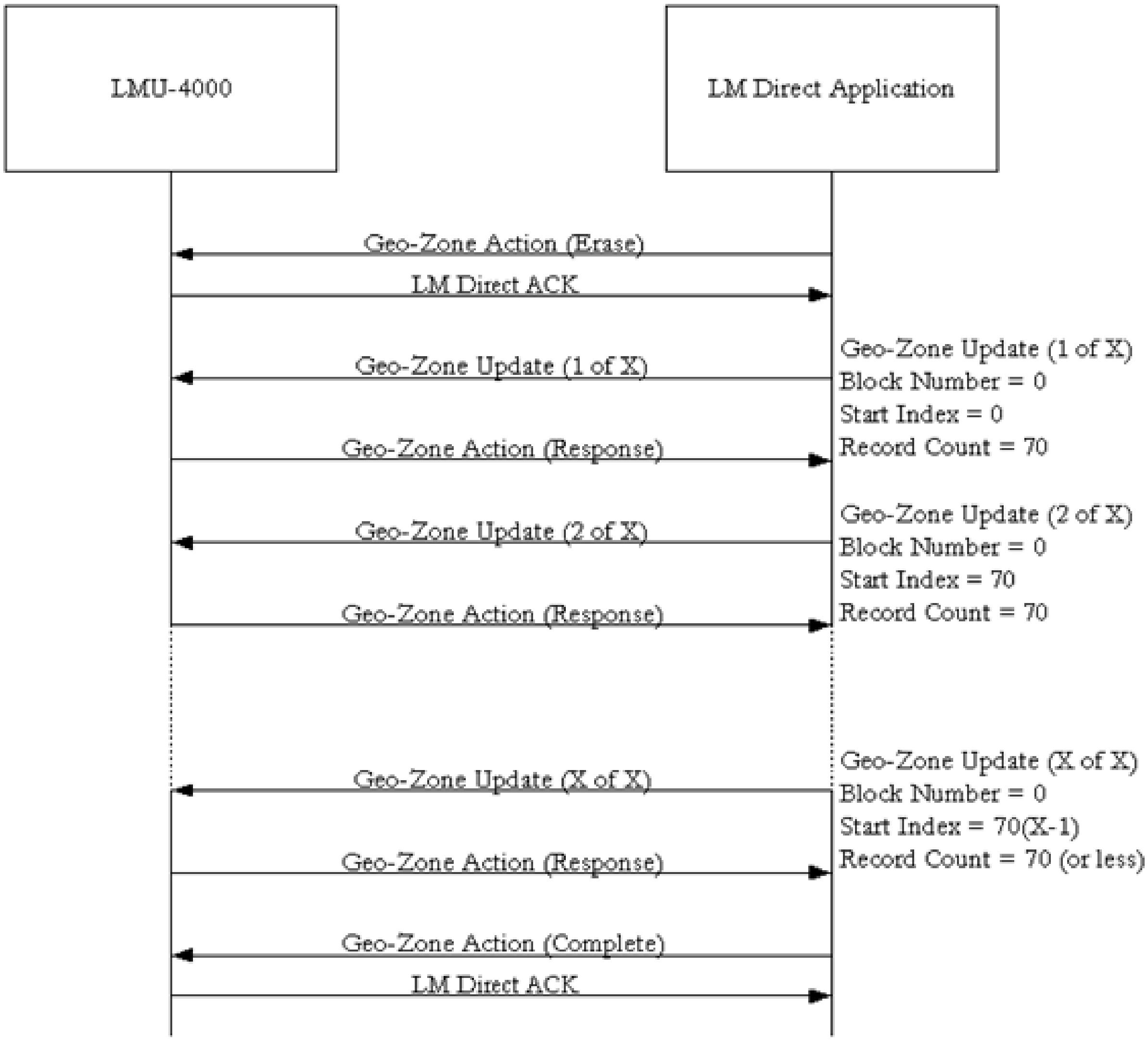


Geo-Zone Read/Write Version Process

9.1.2 Geo-Zone Write Requests

The basic write procedure for geo-zones is as follows.

1. Send an Erase Request
2. Wait for ACK
3. Send a Write Request
4. Wait for Write Complete Report
5. Repeat steps 3&4 until all points have been sent
6. Send a Write Complete Request
7. Wait for ACK



Geo-Zone Write Process

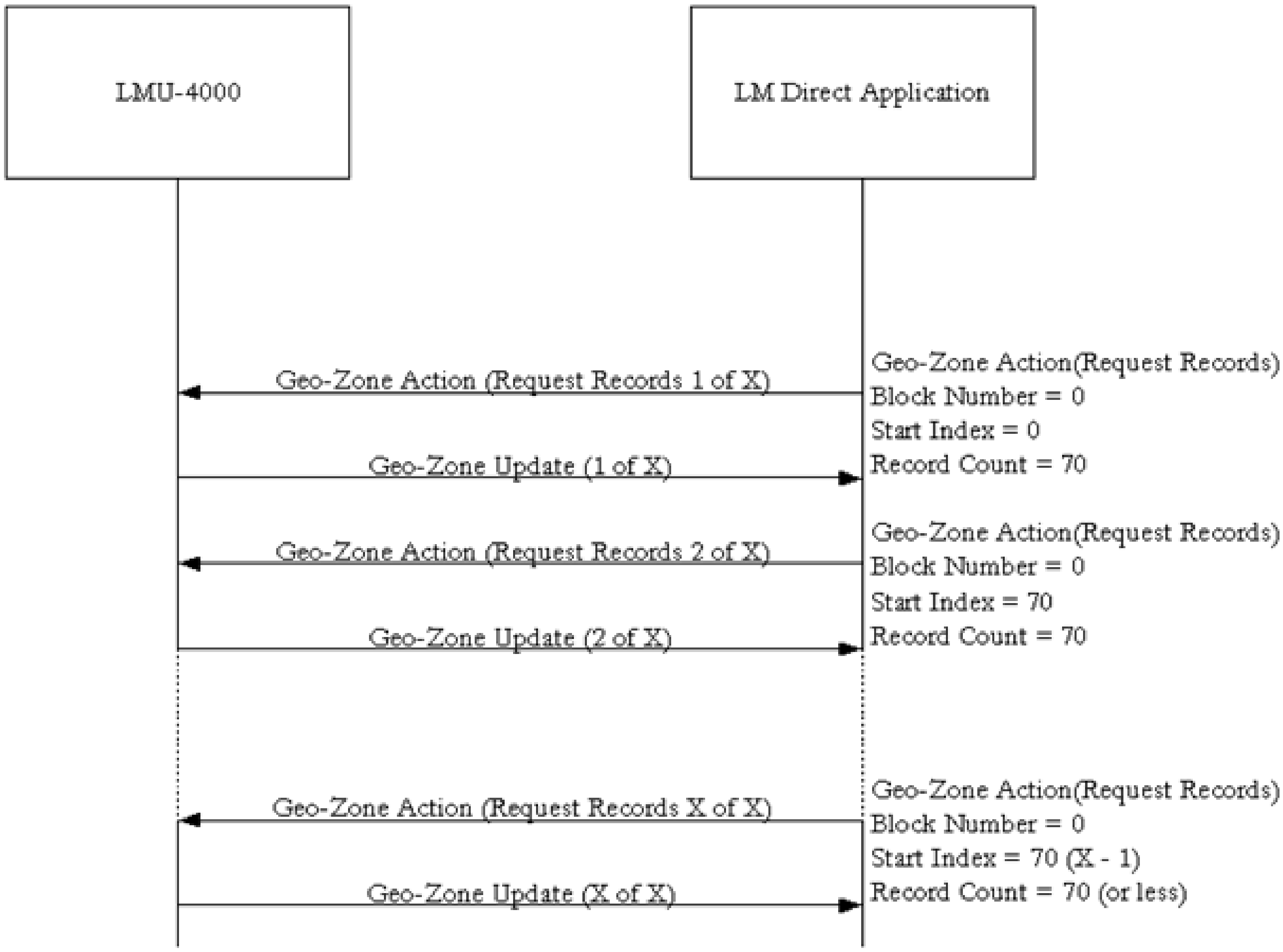
It is important to note that geo-zone processing is stopped when the Erase request is received. It is only restarted when the Write Complete request is received (or the LMU is power cycled).

Application developers must also keep in mind that each write request can hold a maximum of 70 Geo-Zone point definitions.

9.1.3 Geo-Zone Read Requests

The Read process for geo-zones is somewhat simpler since geo-zone processing does not need to be halted.

1. Send a Read Request
2. Wait for Update Message
3. Repeat steps 1&2 until all desired records are read



Geo-Zone Read Process

Again, it is important to note that Update messages will contain a maximum of 70 Geo-Zone.
Also note that the LMU will return a Geo-Zone Action Message if the read request contains an invalid Start Index, Record Count or Block Number.

10 Appendix G — Accumulator Reporting Formats

10.1 Accumulator Reporting Overview

Accumulator contents are included in several LMDirect message types: Event Report, User Messages, etc. Each of these messages has a 1-byte Accumulator Count field and a list of 4-byte fields. The Accumulator Count field is broken up into two sub-fields, the count (lower 6-bits) which indicates the number of 4-byte fields in the following list, and the accumulator reporting format type (upper 2 bits) which specifies the contents and organization of the following list. The following sections describe each format type. Reminder: All fields are transmitted Most-Significant Byte first.

10.2 Standard Accumulator Reporting

The standard format for reporting Accumulators within a message consists of the first ‘n’ Accumulators starting with Accumulator 0 and ending with Accumulator ‘n-1’. The Accumulator Count field in the message is broken up into two sub-fields, the count (lower 6-bits) which specifies the number of accumulators and has the value ‘n’, and the format type (upper two bits) which has a value of zero (0) for the standard format.

Accum 0 (msByte)	Accum 0
Accum 0	Accum 0 (lsByte)
Accum 1 (msByte)	Accum 1
Accum 1	Accum 2 (lsByte)
...	...
Accum 15 (msByte)	Accum 15
Accum 15	Accum 15 (lsByte)

Standard Accumulator Reporting Format

10.3 Enhanced Accumulator Reporting (32-bit)

The Enhanced32 format (format type 1) for reporting Accumulators within a message consists of two 32-bit (4-byte) bit-mapped fields followed by a variable list of 4-byte Accumulator values. This format is intended for use when up to 32 accumulators are in use on the device.

The first of the 32-bit bit-mapped fields indicate which accumulators in the device are currently equaling or exceeding their configured threshold. The least significant bit (bit-0) represents the threshold status of Accumulator 0. The next bit, bit-1, represents the threshold status of Accumulator 1 and so on. If a threshold status bit is set, the value in the respective accumulator equals or exceeds that accumulator’s threshold value.

The second of the two 32-bit bit-mapped fields indicate which accumulators are reported in the following list. The least significant bit (bit-0) represents Accumulator 0. The next bit, bit-1, represents Accumulator 1 and so on. If a bit is set, the respective accumulator’s contents are included in the following list. If an accumulator does not have a bit set, it will not be present in the list.

The list of 4-byte accumulator contents follows the two 32-bit bit-mapped fields. The order of accumulators in the list is increasing from the lowest accumulator with a bit set in the 32-bit bit-mapped ‘reported’ field to the highest accumulator with a bit set. For instance, if only bits 3, 10 and 24 are set, then only Accumulators 3, 10 and 24 will be included in the list and they will occur in the order 3 first, 10 seconds and 24 last. In this case, the list would be 12 bytes long (3 accumulators X 4-bytes-per- accumulator).

Acc0-31 Threshold (ms-Byte)	Acc0-31 Threshold
Acc0-31 Threshold	Acc0-31 Threshold (ls-Byte)
Reported Acc0-31 (ms-Byte)	Reported Acc0-31
Reported Acc0-31	Reported Acc0-31 (ls-Byte)
Accum A (msByte)	Accum A
Accum A	Accum A (lsByte)
Accum B (msByte)	Accum B
Accum B	Accum B (lsByte)
...	...

Enhanced 32 Accumulator Reporting Format