ALERT2 Intelligent Network Device
Application Program Interface Specification

June 2020
Version 2.0

ALERT2 Technical Working Group of the National Hydrologic Warning Council

Revision history

Date	Version	Description	Author
2011/4/4	0.2	Application Layer API to TWG	R. Chris Roark
2011/4/12	0.3	Intelligent Network Device API	R. Chris Roark
2011/10/20	0.4	IND API, String Hdr	R. Chris Roark
2012/3/12	0.5	Revisions	R. Chris Roark
2013/4/5	1.0	Final Release	David Leader, R. Chris Roark
2015/6/23	1.1	Update	David Leader, R. Chris Roark
2016/2/5	1.1	Incorporated review comments	
2020/03/31	2.0	Major rework / API overhaul	Adam Torgerson, David Van Wie

Release history

Date	Version	Status	Audience	Approval
2011/4/12	0.3	Draft	TWG	R. Chris Roark, Don Van Wie
2013/4/5	1.0	Final Release	TWG	Don Van Wie
2015/6/23	1.1	Draft	TWG	David Haynes
2016/2/5	1.1	Final	TWG	David Haynes
2020/6/10	2.0	Final	TWG	David Haynes

Contact details

National Hydrologic Warning Council
2480 W. 26th Ave., Suite 156-B
Denver, CO 80211
ALERT2@hydrologicwarning.org
www.hydrologicwarning.org

ALERT2 Protocol Technical Working Group
Technical Working Group of the Standards and Guidance Committee of the
National Hydrologic Warning Council

Peter Acton
Ilse Gayl
David Haynes, Chair
Scott Holder
Dan Hurley
Brian Iserman
David Leader
Carrie Lery
James Logan
Steve Malers
Ron Marotto
Mark Moore
Andy Morin
Markus Ritsch
R. Chris Roark
James Slouber
Adam Torgerson
Sam Utley
David Van Wie
Don Van Wie

1. Content

Table of Contents

2. List of Tables

3. List of Figures

4. TLV Listings

5. Introduction

5.1. IND Feature Sets

An IND that encodes and transmits is referred to in this specification as an A2 Encoder & Modulator. Its primary function is to accept application protocol data units (APDUs), encapsulate them in a MANT PDU while providing the requested MANT services, and ultimately aggregate and transmit them via the AirLink protocol on an A2 radio network. An Encoder & Modulator device shall support at least the Core and Encode Feature Sets.

An IND that receives and decodes AirLink frames from an A2 architecture radio network is named an A2 Demodulator & Decoder in this specification. Its function is to receive AirLink radio frames, provide the AirLink and MANT protocol demodulation and decoding services requested or necessary, and present any application layer PDUs to the appropriate protocol port of a receiving APD. An A2 Demodulator & Decoder device shall support at least the Core and Decode Feature Sets.

An IND that implements both the Encoder & Modulator functionality and the Demodulator & Decoder functionality in a single device is an A2 Modem. An A2 Modem device shall support at least the Core, Encode, and Decode Feature Sets, and will potentially support all Feature Sets.

An IND that implements the Decode functionality and provides a mechanism for retransmitting received AirLink frames is named an A2 Repeater. An A2 Repeater device shall support at least the Core, Encode, Decode, and Repeat Feature Sets.

This specification splits the A2 IND functionality into several Feature Sets, which define the API:

Feature Set	Description
Core	Core functionality required on all ALERT2 devices
Encode	Encoding and modulation of ALERT2 for transmission on an AirLink network.
Decode	Demodulating and decoding ALERT2 data received from an AirLink radio network, and outputting data in the Message API format
Repeat	Retransmitting received messages on an ALERT2 network
ALERT CCN	Support for the ALERT2 ALERT Concentration protocol
GPS	The device has a GPS or other satellite-based receiver for timekeeping services
RS-232	Support for RS-232 Serial ports
TCP/IP	Support for TCP/IP connectivity

CFEC	Support for Configurable FEC
Status	Support for IND-initiated status reports
EERDS	Support for the end-to-end reliable datagram service
Encryption	Support for the ALERT2 Encryption and Authentication service

Table 1 ALERT2 IND API Feature Sets

Within a Feature Set, TLVs may be specified as Mandatory, Recommended, Optional, or Deprecated. TLVs specified as Mandatory shall be implemented by INDs implementing the relevant Feature Set. TLVs specified as Recommended should be implemented, and those specified as Optional or Deprecated may be implemented. TLVs specified as Deprecated may be removed in a future version of the API, and their use should be avoided.

5.2. API Specification Structure and Definitions

All content is exchanged in a binary Type, Length and Value (TLV) format (except for the ALERT concentration interface and the CSV data output).

This document specifies two different APIs: the **IND API**, for configuration and control of an ALERT2 IND device; and the **Message API**, for reporting and transmission of ALERT2 data from an IND to other devices not on the ALERT2 network.

The TLVs that make up the IND API have been divided into two types: **API commands**, which define an action to take, should be sent from the "top level", and not wrapped in another TLV; **API parameters**, which refer to specific elements of an IND, must be wrapped in a Set or Get API command in order for the IND to process them. API parameters may be read-only, requiring a Get, write-only, requiring a Set, or read-write, allowing either a Get or Set. This is shown in the Access field in the API parameter table as RO (read-only), WO (write-only), or RW (read-write). An API parameter that is not wrapped in a Set or Get TLV shall be ignored by the IND.

Much of the configuration information of an IND will be static, such as the source address and media access parameters. It is recommended every IND contain non-volatile memory storage where this static information can be saved. Dynamic, per Application PDU, configuration parameter changes are allowed. Absent any dynamic configuration parameter change the IND must create a MANT header and Payload in accordance with the configuration parameter values stored in the non-volatile storage. Any such dynamic configuration parameter change takes immediate effect, and are applied across all inputs to the IND, before processing subsequent parts of that Application PDU. Configuration parameters provided to the IND but not saved into non-volatile storage will apply until they are 1) subsequently changed by an API command, 2) overwritten by a "Reset configuration to defaults" command, 3) overwritten by a "Recall non-volatile configuration to current", or 4) lost due to a power-down event.

5.3. API Compatibility

This version of the IND API Specification breaks compatibility with version 1.0, which is widely in use. Specifically, the addition of the Get and Set API commands and the requirement to wrap a majority of other API commands within a Get or Set command are not compatible with Version 1.0. These were added in version 1.1 of the IND API specification, and this compatibility change was necessary in order to clear up ambiguities involved in using the IND API in the MANT Command & Control service. The updated specifications of the Message API, both binary and CSV formats, also break compatibility with older versions.

Regardless of the specification, existing deployed implementations of then Version 1.0 ALERT2 IND API do not implement the API Version TLV exchange. Further, deployed implementations use the prefix string "ALERT2", without an additional specifier for binary or ASCII format. Consequently, in order to remove any ambiguity, the message prefix used for version 2.0 has been changed.

Compatibility with future releases is addressed through the use of a versioning field in the message prefix string. When messages are sent in a streaming format (e.g., over a serial port or a TCP socket), the transmission begins with a prefix string defined as: AL2<*version*><*format*>. The *version* string shall be "2", matching the return from API 0x75; format shall be "a" for ASCII (Message API CSV format) or "b" for binary. If the message is in binary format, the version string shall be followed immediately by the message length encoded as an extensible value.

6. Examples of IND API Operation

<u>NOTE</u>*:* All values in the example tables are listed in hexadecimal.

6.1. Example: Self-Reporting Protocol Data Unit

6.1.1. Description

This example shows the binary message an APD sent to the IND, using version 2, to send a Self-Reporting Protocol Data Unit with Source Address 4403. All other IND configuration parameters are as previously configured.

This message begins with the API Prefix and length. Next is a "Set" command TLV, which contains a Source Address parameter TLV (0x18). The combination tells the IND to set its source address to 4403 (0x1133).

Finally, there is a Self-Reporting Protocol command TLV, which contains a single application layer PDU (see the Application Layer specification for more details on the APDU format). This APDU contains:

- A control byte of 0x70: APDU ID disabled, not test, no timestamp, version 0
- A tipping bucket rain gauge report with a 4-byte accumulator set to 104 and 4 tips at -32, -21, -16, and -2 seconds

- A general sensor report with two sensors:
  - pH Sensor: ID = 18, Format|Length = 1|2 (2-byte unsigned int), Value = 804
  - Water Temperature Sensor: ID = 19, Format|Length = 2|2 (2-byte signed int), Value = 630

6.1.2. Raw Data

Prefix and Length	Set Parameter		Self-Reporting Protocol			
		Source Addr		Application PDU		
				Ctl	Tipping Bucket	General Sensor Report
41 4C 32 32 62 1F	0A 04	18 02 11 33	00 17	70	02 0A 01 14 00 00 00 68 20 15 10 02	01 08 12 12 03 24 13 22 02 76

Figure 1 Example: Self-Reporting Protocol Data Unit Raw Data

6.1.3. Illustration

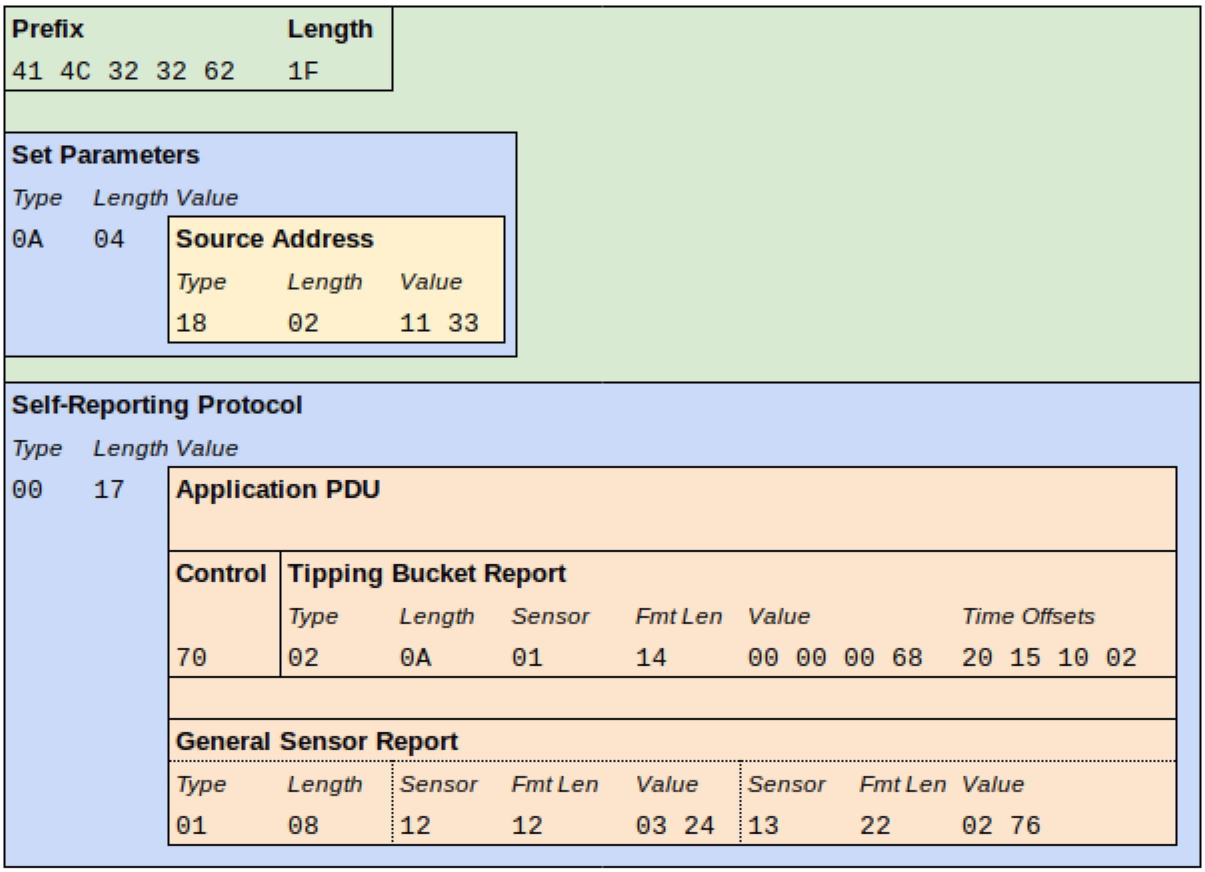

Figure 2 Example: Self-Reporting Protocol Data Unit Illustration

6.2. Example: Local Set TDMA Parameters

6.2.1. Description

This example shows the binary string an APD would send to the IND to set the following TDMA parameters:

- TDMA Frame Length (0x48) = 1 minute (60,000 milliseconds)

- TDMA Slot Start Offset (0x4B) = 30 seconds (30,000 in milliseconds)
- TDMA Slot Length (0x4A) = 1 second (1,000 in milliseconds)

The configuration is then saved to non-volatile configuration storage in the IND; until changed either temporarily or again into non-volatile storage, the IND uses these TDMA parameters for all communications.

6.2.2. Raw Data

Prefix and Length	Set Parameter				Save
		Slot Length	Frame Length	Slot Offset	
41 4C 32 32 62 12	0A 0E	4A 02 03 E8	48 03 00 EA 60	4B 03 00 75 30	78 00

Figure 3 Example: Local Set TDMA Parameters Raw Data

6.2.3. Illustration

Prefix	**Length**
41 4C 32 32 62	12

Set Parameters

Type	*Length*	*Value*							
		TDMA Slot Length			**TDMA Frame Length**			**TDMA Slot Start Offset**		
		Type	*Length*	*Value*	*Type*	*Length*	*Value*	*Type*	*Length*	*Value*
0A	0E	4A	02	03 E8	48	03	00 EA 60	4B	03	00 75 30

Save Configuration	
Type	*Length*
78	00

Figure 4 Example: Local Set TDMA Parameters Illustration

6.3. Example: Local TDMA Parameters Get Configuration

6.3.1. Description

This example shows the binary string an APD would send to the IND to get the TDMA Slot Length (0x4A), the TDMA Frame Length (0x48), and the TDMA Slot Start Offset (0x4B), as well as the binary string the IND would send to the APD in response.

6.3.2. Raw Data

Prefix and Length	Get Parameter	
		Params
41 4C 32 32 62 05	0B 03	4A 48 4B

Prefix and Length	Slot Length	Frame Length	Slot Start Offset
41 4C 32 32 62 0E	4A 02 03 E8	48 03 00 EA 60	4B 03 00 75 30

Figure 5 Example: Local TDMA Parameters Get Configuration Raw Data

6.3.3. Illustration

Prefix	**Length**			**Get Request**
41 4C 32 32 62	05			
Get Parameters				
0B 03	**Slot Length**	**Frame Length**	**Slot Start Offs**	
	Type	*Type*	*Type*	
	4A	48	4B	

Prefix	**Length**								**Get Response**
41 4C 32 32 62	0E								
TDMA Slot Length			**TDMA Frame Length**			**TDMA Slot Start Offset**			
Type	*Length*	*Value*	*Type*	*Length*	*Value*	*Type*	*Length*	*Value*	
4A	02	03 E8	48	03	00 EA 60	4B	03	00 75 30	

Figure 6 Example: Local TDMA Parameters Get Configuration Illustration

6.4. Example: Remote Set and Get

6.4.1. Description

This example shows the binary string an APD would send to an IND to perform configuration of a remote IND over the ALERT2 Network. In this case, the goal is to update the pass list on a repeater (address 10042) to include address 4242, and then ask the repeater to report the complete pass list.

There are three steps involved:

1. Configure the local IND to include destination address in the outgoing MANT PDU, and set the destination address to 10042.
2. Use the configuration and control command (type 0x02) to create a MANT PDU with the required payload. It will
   a. Set the current address list

b. Add the desired address (4242, or 0x1092 hex)
c. Save the configuration to non-volatile storage
d. Request the address list

3. Update the local IND configuration to not use the destination address

This example does not show the TLVs required to enable encryption. It is assumed that encryption is already configured and enabled for packets originating from this IND. If that were not the case, we could add extra TLVs to control encryption alongside those that enable and disable sending the destination address.

6.4.2. Raw Data

Prefix and Length	Set Parameter			ALERT2 Configuration and Control							Set Parameters	
		Dest Addr	Incl DA		Configuration and Control PDU							Incl DA
					Set	Addr List	Add Addr	Save	Get AL			
41 4C 32 32 62 1E	0A 07	19 02 27 3A	1B 01 01	02 0E	0A 07	31 01 00	35 02 10 92	78 00	0B 01 3B		0A 03	1B 01 00

Figure 7 Example: Remote Set and Get Raw Data

6.4.3. Illustration

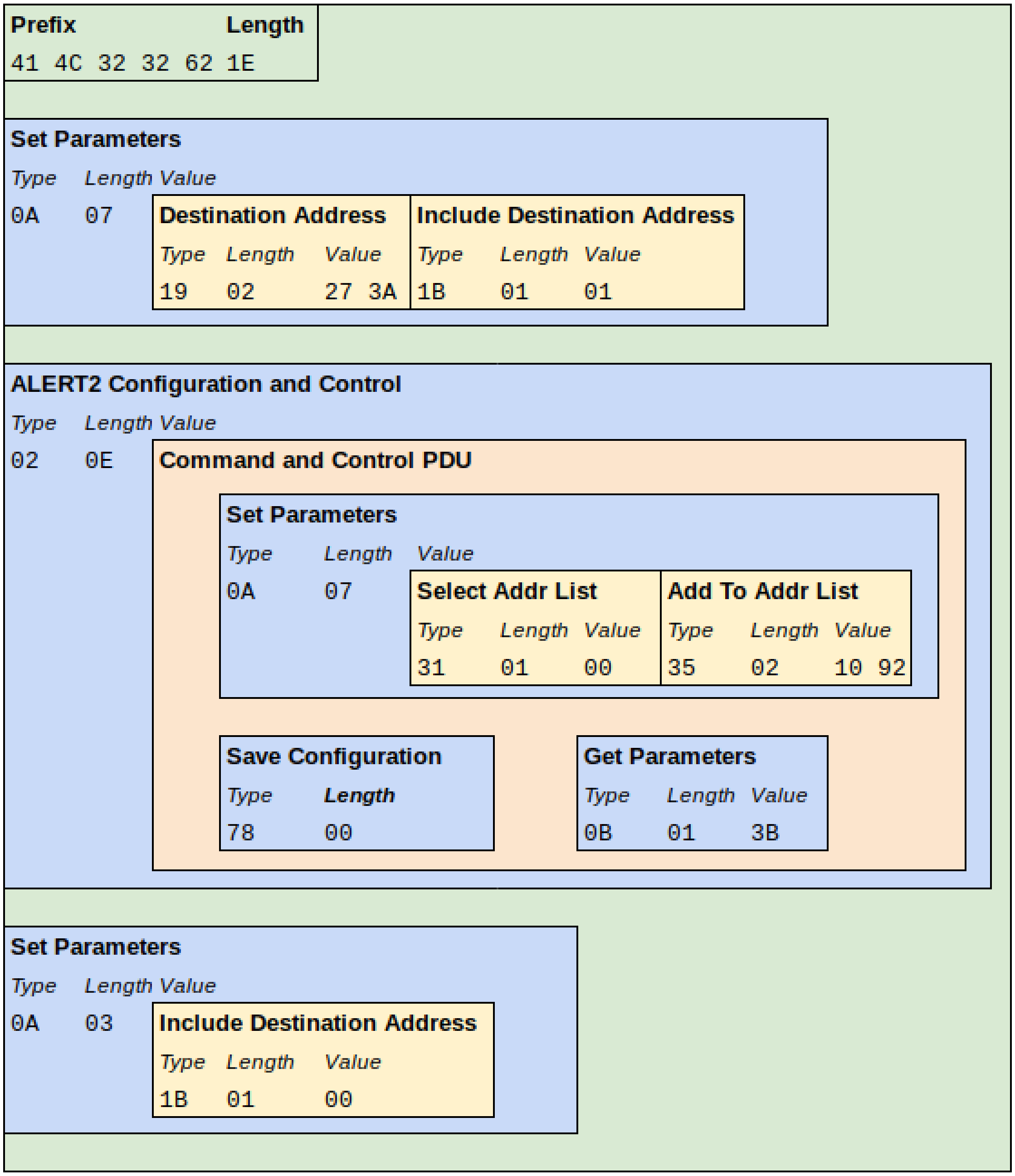

Figure 8 Example: Remote Set and Get Illustration

7. Examples of Message API Operation

In these examples, the fields are shown in three lines, the field TLV labels, the decimal values and the equivalent hex bytes sent down the wire.

7.1. Example: ALERT2 data without errors

7.1.1. Description

This example shows the binary string produced by the IND upon receipt of an AirLink message containing a single MANT PDU.

The TLVs show the following:

Agency ID	SACCO
Timestamp	2010-12-30 09:00:00.250 UTC
Time Sync Status	4 (NTP; Suitable for decoding, but not TDMA)
Decoding IND Address	8000
AirLink Frame Length	19
AirLink Total Sym Error Corrected	0
Per-Block Errors Corrected	0
Noise Level	24
FEC Mode	0
Number of MANT PDUs	1
MANT Authentic	0
MANT Header	00:00:70:0D:11:13
MANT Payload	24:7E:87:03:08:47:02:A8:3E:27:88:01:79

Table 2 Message API Data Without Errors Decoded TLVs

7.1.2. Illustration

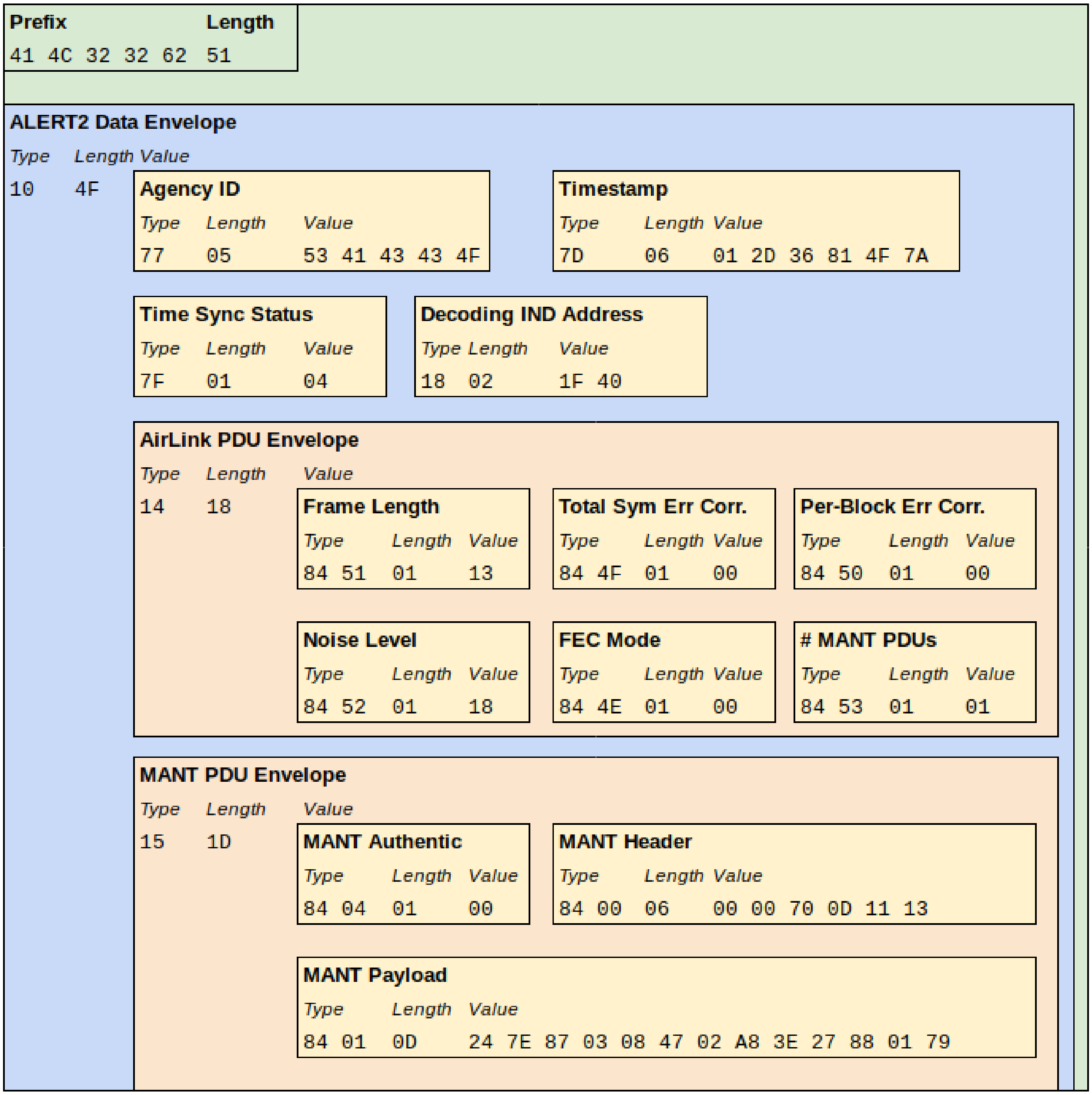

Figure 9 Message API Data Without Errors Illustration

7.2. Example: ALERT2 data with AirLink error

7.2.1. Description

This example shows the binary string produced by the IND upon receipt of an AirLink message with too many symbol errors to decode.

The TLVs show the following:

Agency ID	SACCO
Timestamp	2010-12-30 09:00:20.250 UTC
Time Sync Status	4 (NTP; Suitable for decoding, but not TDMA)
Decoding IND Address	8000
AirLink Frame Length	24
AirLink Total Sym Error Corrected	255 (-1)
Per-Block Errors Corrected	255 (-1)
Noise Level	84
FEC Mode	0
Number of MANT PDUs	0
AirLink Error	1 (Bad AirLink first block)
AirLink Header	00:02
AirLink Payload	00:7F:CC:1D:11:13:A4:08:3F:DF:01:24:55:80:FF:0A:04:99:00:00:55:55

Table 3 Message API Data With Errors Decoded TLVs

7.2.2. Illustration

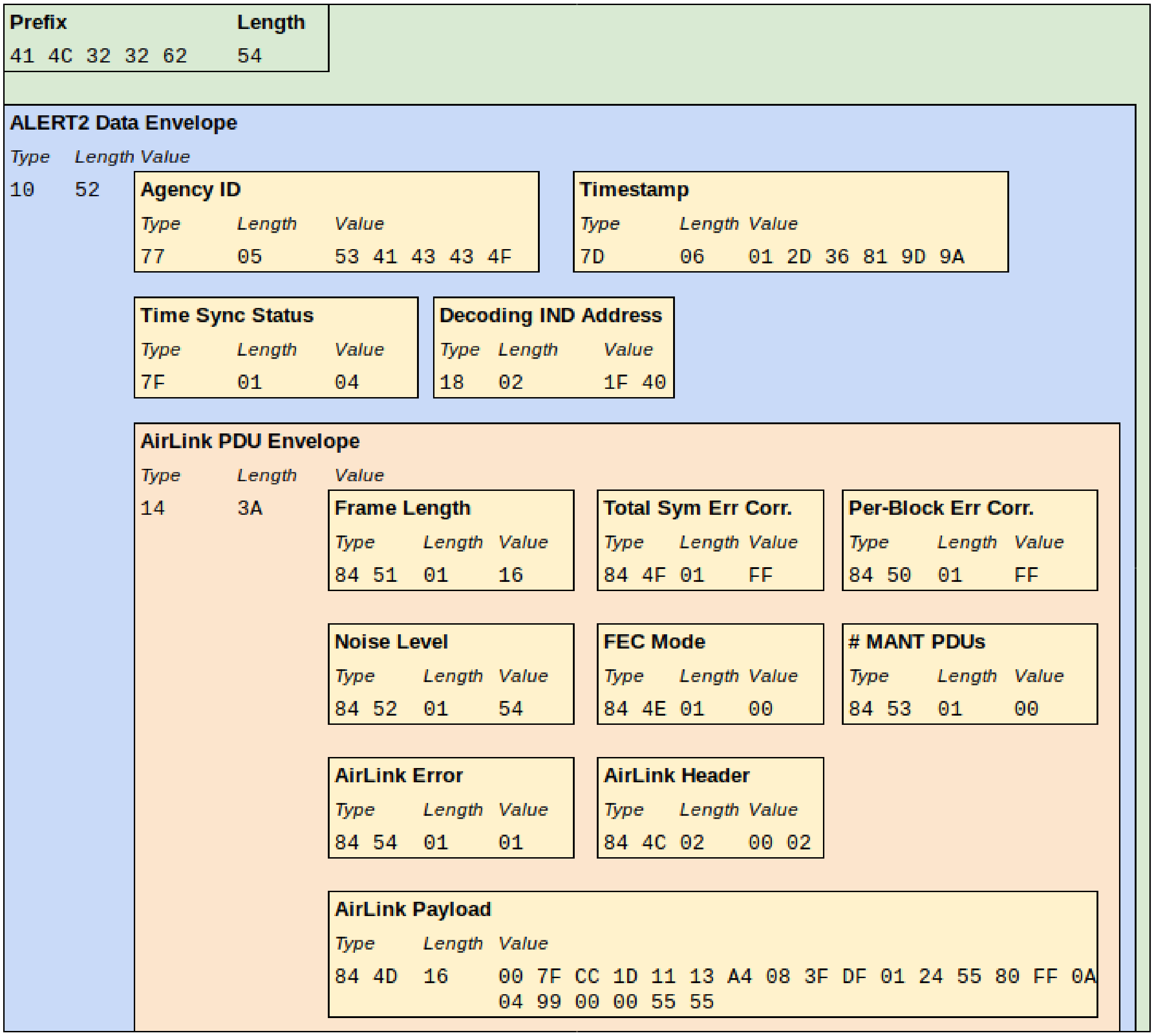

Figure 10 Message API Data With Errors Illustration

7.3. Example: ALERT2 Data without AirLink Envelope

7.3.1. Description

The following example shows an ALERT2 message transmitted without the AirLink envelope. This message might originate directly from a sensing device and be sent over cell modem, satellite, or other IP connection directly to an ALERT2 data collection application.

The message sent in this example is the same as the one shown in section 4.1, but the AirLink envelope TLV is absent. Because this message was never transmitted over an AirLink layer, it does not make sense to include that information.

In the future, if ALERT2 messages are routed across multiple transport media before leaving an ALERT2 network, the API definition may be extended to support additional envelope types beyond AirLink.

7.3.2. Illustration

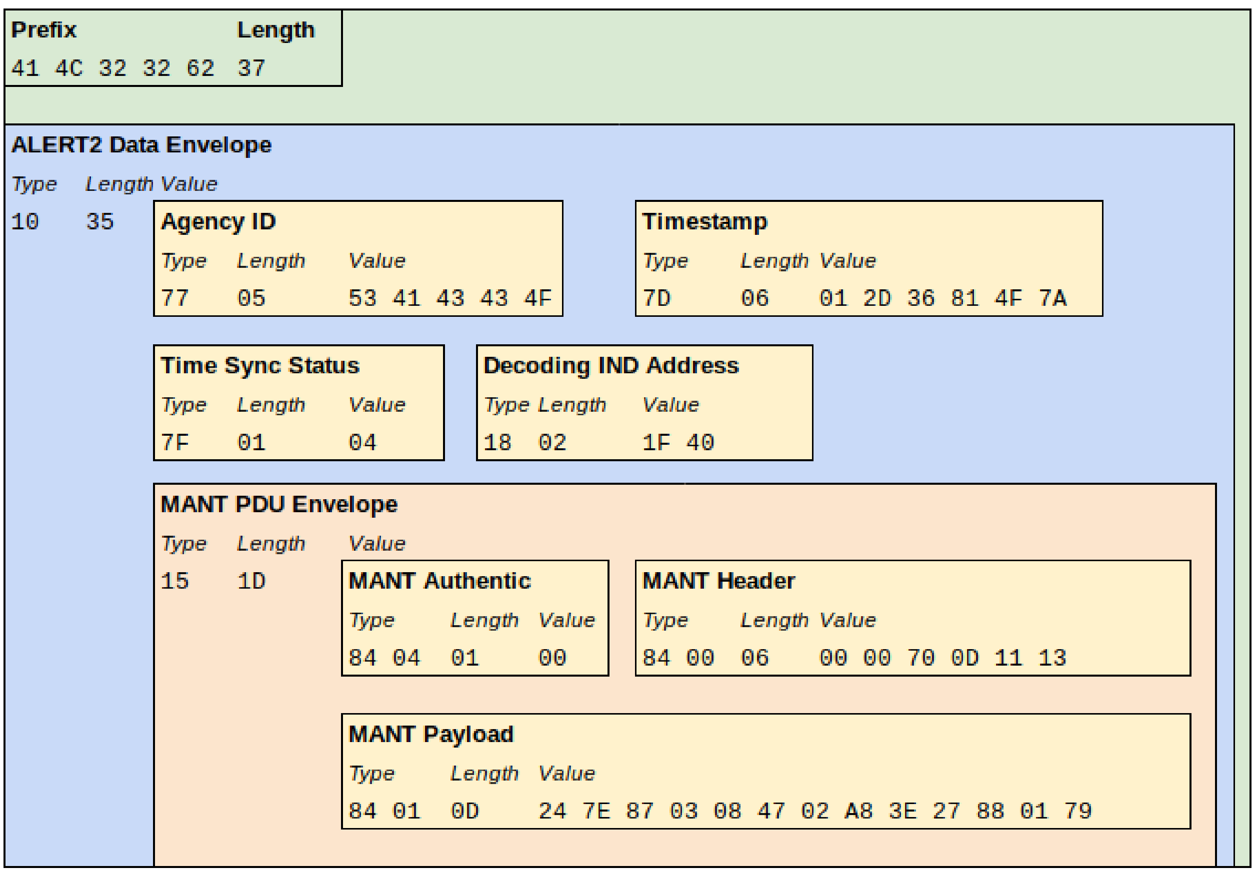

Figure 11 Message API Data Without AirLink Envelope Illustration

7.4. Example: CSV Output Format

The following examples show the output that would be produced from the receipt of three AirLink messages: first, the AirLink message shown in Example: ALERT2 data without errors; second, the AirLink message shown in Example: ALERT2 data with AirLink error; and, finally, an AirLink message received from UDFCD containing both a sensor report and an ALERT Concentration report. The optional sensor report output is shown.

In practice, an IND may produce all of these outputs, in CSV format, in a single ASCII stream in chronological order with the different message type records intermingled. A filtering tool, such as standard spreadsheet software, can quickly extract all of the records of a given type.

Please note that given width constraints in this document, each example is broken up into several tables, but in actual CSV output, each record -- beginning with the API version -- would be a single line, with commas separating the fields.

7.4.1. AirLink

API Version	Timestamp	Time Sync Status	Agency Identifier	Decoding IND Address	AirLink Message Type	AirLink Version	AirLink Reserved	AirLink Frame Length	AirLink Symbol Corrected	Total Errors
AL22a	2010-12-30 09:00:00.250	4	SACCO	8000	AirLink	0	0	19	0	
AL22a	2010-12-30 09:00:20.250	4	SACCO	8000	AirLink	0	0	24	-1	
AL22a	2018-07-04 00:12:32.091	4	UDFCD	10127	AirLink	0	0	30	0	

AirLink Error ID	AirLink Error Description	Raw Data	Block Error List	FEC Mode	Noise Level
0		00:13:00:00:70:0D:11:13:24:7E:87:03:08:47:02:A8:3E:27:88:01:79	0	0	24
1		00:02:00:7F:CC:1D:11:13:A4:08:3F:DF:01:24:55:80:FF:0A:04:99:00:00:55:55	-1	0	84
0		00:1E:02:00:00:07:27:3F:01:19:6A:04:02:D8:03:02:20:01:02:10:10:07:19:6A:00:14:02:F0:B7:D1:62:09	0/0	0	18

Table 4 CSV Format AirLink Data

7.4.2. MANT

API Version	Timestamp	Time Sync Status	Agency Identifier	Decoding IND Address	MANT Message Type	MANT Version	MANT Protocol ID	Time Stamp Request enabled	Add Path Service Request enabled	MANT Port
2	2010-12-30 09:00:00.250	4	SACCO	8000	MANT	0	0	0	0	0
2	2018-07-04 00:12:32.091	4	UDFCD	10127	MANT	0	0	0	1	0
2	2018-07-04 00:12:32.091	4	UDFCD	10127	MANT	0	0	0	1	1

MANT Payload Encrypted	MANT Reserved Bits	ACK	Hop Limit	Source Address	Destination Address	MANT PDU ID	Path Length	Repeater Path	Payload Length	Payload
0	0	0	7	4371			0		13	24:7E:87:03:08:47:02:A8:3E:27:88:01:79
0	0	0	0	10047			1	6506	7	04:02:D8:03:02:20:01
0	0	0	1	6506			0		7	14:02:F0:B7:D1:62:09

MANT Error ID	MANT Error Description	MANT Authentic
0		0
0		0
0		0

Table 5 CSV Format MANT Data

7.4.3. ALERT Concentration

API Version	Timestamp	Time Sync Status	Agency Identifier	Decoding IND Address	ALERT CCN Message Type	ALERT Timestamp	ALERT ID	ALERT Data Value
AL22a	2018-07-04 00:12:32.091	4	UDFCD	10127	ALERT CCN	2018-07-04 00:12:23	4535	1634

Table 6 CSV Format Concentration Data

7.4.4. Sensor

API Version	Timestamp	Time Sync Status	Agency Identifier	Decoding IND Address	Sensor Message Type	Reading Timestamp	Site Address	Sensor ID	Value
AL22a	2010-12-30 09:00:00.250	4	SACCO	8000	Sensor	2010-12-30 08:59:51	4371	1	680
AL22a	2010-12-30 09:00:00.250	4	SACCO	8000	Sensor	2010-12-30 08:59:51	4371	2	62
AL22a	2010-12-30 09:00:00.250	4	SACCO	8000	Sensor	2010-12-30 08:59:51	4371	3	10120
AL22a	2010-12-30 09:00:00.250	4	SACCO	8000	Sensor	2010-12-30 08:59:51	4371	7	3770
AL22a	2018-07-04 00:12:32.091	4	UDFCD	10127	Sensor	2018-07-04 00:12:08	10047	6	1

Table 7 CSV Format Sensor Data

8. Binary API

8.1. TLV Format

Each of the APIs, except the CSV Format of the Message API (found in Message API CSV Format), operates through the use of TLV formatted data. TLV data is 8 bit binary data in the form <type><length><value>, where the value field may be another recursively embedded <type><length><value> structure. Examples of the IND API may be found here. The Value field is optional; if the length of a TLV is 0, the Value field shall not be included.

The <type> and <length> fields are extensible: To encode a value greater than 127 requires a 2-byte field. To encode a 2-byte field Bit 7 (the high order bit) of the first byte sent (MSB) is set to 1 and the length value is encoded in the following 15 bits. A value of 127 or less is encoded in a single byte (whose high order bit is 0). On decoding, the MSB is read first, and if the high bit contains a 1, the value is read from the following 15 bits. If the high bit is 0, the value is read as the value of that byte. The one byte field may carry a value of 0 to 127 and the two-byte field may contain a number from 0 to 32,767. The format is shown below:

					Length	
MSB 7 bits Type	[LSB 8 bits Type]	MSB 7 bits Length	[LSB 8 bits Length]	data	...	data

Table 8 Type and Length Field Extensibility

All Parameter TLVs must be prefaced with either the Set Parameter TLV in order to write values or the Get Parameter TLV to read values. Multiple parameter TLV lists may be included in a single Set Parameter TLV or Get Parameter TLV. If a parameter TLV is not within a Set Parameter or Get Parameter TLV that parameter TLV shall be ignored.

Set Parameter and Get Parameter TLVs are not allowed to contain other command TLVs; e.g., no Set Parameter TLV can include a Get Parameter TLV within its length. If an IND detects a command TLV when a Parameter TLV is expected, the embedded command TLV shall be ignored.

A single Asynchronous Binary Interface string may include concatenated Set Parameter and Get Parameter TLVs: i.e. a Set Parameter TLV may be followed by a Get Parameter TLV followed by another Set or Get Parameter TLV.

For example, a Set Parameter TLV command would be:

First Parameter			Second Parameter			
Type	Length	Value	Type	Length	Value	...

← Total Length →

Type	Total Length	
10		...
0A		

Table 9 Set Parameter TLV Example

The value field of the Get Parameter command TLV should contain a list of TLV types to be returned; the Get Parameter command TLV is not recursive, and only the TLV types are specified while the length and value fields are omitted. The format of the Get Parameter TLV command is:

Type	Total Length	Type 1	Type 2	...
0B	...	XX	XX XX	...

Table 10 Get Parameter TLV Example

Some Parameter TLVs may be assigned Default Values. The initial value of any implemented Parameter TLV shall be assigned to the Default Value, if listed.

For any Parameter TLVs described with “enabled/disabled”, a setting of “enabled” is defined as the value 1, and “disabled” is defined as the value 0.

8.2. Error Handling

In the IND API, commands are not explicitly acknowledged and reported as successful or not. This has not changed in the 2.0 revision of the API. Instead, APD implementations are encouraged to verify that parameters were successfully set by sending a Get command immediately after the Set command. Similarly, when using the Save Configuration command, APDs can check if the configuration was successfully saved with the Flash Status command.

In the Message API, there is a need to be able to report errors encountered when decoding AirLink or MANT PDUs. These errors can occur either due to a bad radio path, where there is too much noise to be able to properly decode the message, or due to a faulty or misconfigured IND or APD introducing bad data to the ALERT2 network. The AirLink protocol ensures that all messages that are successfully decoded contain no errors - so radio noise cannot introduce errors by altering the payload of messages.

In the case of a decoding error, an error TLV with a descriptive code will be present in the relevant PDU envelope. There is no explicit top-level error envelope in the message API. The API also permits for vendor-defined error types and error descriptions.

8.3. API Extensibility

Some thought has been put into the separation of types in the TLV space. Type values less than 128 can be encoded in a single byte, and some care should be taken before exhausting that range. Where possible, types with meaning in a similar space should be grouped together in the numerical space.

In order to support extensibility, applications processing the Message API should not fail or take an action other than logging when they encounter a TLV that is not understood or implemented. This allows TLVs implemented in the future to behave predictably on older systems.

8.4. Vendor Specific TLVs

Many ALERT2 INDs will offer functionality beyond what is defined in the ALERT2 specification. In order to allow for that additional functionality to be configured in a consistent manner, TLV types **0x9B58** - **0x9D4C** (decimal 7000 - 7500) have been reserved for vendor-specific implementation. Vendors electing to make use of these TLVs shall make available documentation describing the TLVs and their function. These TLVs are not defined by the ALERT2 specification, and may implement different behaviour on different devices. If these TLVs implement functionality that would be of benefit to the ALERT2 community as a whole, vendors are encouraged to work with the TWG to standardize their definition and implementation.

Users should contact individual device manufacturers for more information on particular vendor-specific TLVs. Further, users should be aware that support for these TLVs will vary from vendor to vendor and device to device.

8.5. IND API Command TLVs

NOTE: All commands have Access type 'WO', because although some may generate a response, command themselves are being written.

Command	**ALERT2 Self-Report**	Type	**0x00**
Creates a MANT PDU with the supplied data as the payload, and enqueues it for transmission.		Default	-
		Range	-
		Support	Mandatory
		Feature Set	Encode

Command	**ALERT2 Concentration**	Type	**0x01**
Creates a MANT PDU with the supplied ALERT concentration payload, and enqueues it for transmission.		Default	-
		Range	-
		Support	Optional
		Feature Set	ALERT CCN

Command	**ALERT2 Configuration and Control**	Type	**0x02**
Creates a MANT PDU with the supplied configuration and control data as the payload, and enqueues it for transmission.		Default	-
		Range	-
		Support	Optional
		Feature Set	Encode

Command	**Set Parameter**	Type	**0x0A**
Request that the IND apply the enclosed TLVs		Default	-
		Range	-
		Support	Mandatory
		Feature Set	Core

Command	**Get Parameter**	Type	**0x0B**
Request to the IND to return TLVs associated with the enclosed list of TLV-types.		Default	-
		Range	-
		Support	Mandatory
		Feature Set	Core

Command	**Forward ALERT2 Messages**	Type	**0x10**
Enqueues messages described in the enclosed ALERT2 Data Envelope PDU for repeating. Messages will be subject to repeater processing as defined in the MANT specification, e.g., pass listing and decrementing hop limit, before being transmitted.		Default	-
		Range	-
		Support	Mandatory
		Feature Set	Repeat

Command	**Save Configuration**	Type	**0x78**
Saves current configuration to non-volatile storage.		Default	-
		Range	-
		Support	Recommended
		Feature Set	Core

Command	**Query Current Configuration**	Type	**0x79**
Returns current configuration. Similar to Get Parameter, but returns all configuration.		Default	-
		Range	-
		Support	Recommended
		Feature Set	Core

Command	**Reset Configuration to Defaults**	Type	**0x7A**
Resets configuration to factory defaults.		Default	-
		Range	-
		Support	Recommended
		Feature Set	Core

Command	**Load Configuration**	Type	**0x7B**
Loads configuration from non-volatile storage.		Default	-
		Range	-
		Support	Recommended
		Feature Set	Core

Command	**Initiate GPS Cycle**	Type	**0x70**
Immediately initiate a GPS time sync cycle. With a value of “1”, force re-acquisition of leap second information. With no value or a value of zero, the IND may determine if re-acquisition of leap seconds is required.		Default	-
		Range	-

Command	**TLV Exists**	Type	**0x8081**
Takes a list of TLV types, and returns an array of bytes containing 1 if the type is implemented on the IND or 0 if the type is not implemented on the IND.		Default	-
		Range	Varies
		Support	Mandatory
		Feature Set	Core

8.6. IND API Parameter TLVs

NOTE: In older revisions of this document, this section was titled "Protocol Services Configuration", and other specification documents may still reference this title.

Parameter	**API Version Number**	Type	**0x75**
Return a one-byte value representing the current API version: 0 = Version 1.0 1 = Version 1.1 2 = Version 2.0		Default	-
		Range	0 - 255
		Access	RO
		Support	Mandatory
		Feature Set	Core

Parameter	**Agency Identifier**	Type	**0x77**
A variable-length string that contains a unique identifier for the agency managing this IND.		Default	NONE
		Range	1 - 64 characters
		Access	RW
		Support	Mandatory
		Feature Set	Decode

Parameter	**Version String**	Type	**0x96**
Return a string describing the IND. The string format shall be a comma separated list of: ● 3-Byte IND API Version ("2.0") ● 3-12 Byte Vendor Name or Abbreviation ● 3-12 Byte Device Type Identifier ● 3-12 Byte Device Firmware Version String ● 0-13 Byte (Optional) Serial Number or Device ID String		Default	-
		Range	-
		Access	RO
		Support	Mandatory
		Feature Set	Core

Parameter	**IND Address**	Type	**0x18**
The address of this IND. This address will be used as the Source Address for messages originating from the IND, will be the address appended to the path list for messages repeated by this IND, and will be the decoder address for messages decoded by this IND.		Default	1000
		Range	1 - 65,534
		Access	RW
		Support	Mandatory
		Feature Set	Core

Parameter	**Destination Address**	Type	**0x19**
If the **Add Destination Address** parameter is enabled, this address will be included in the header of outgoing MANT PDUs in the Destination Address field.		Default	1
		Range	1 - 65,534
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**Add Path Service**	Type	**0x1A**
Determines the value of the **Add Path Service Request flag** in MANT PDUs originating from this IND.		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**Add Destination Address**	Type	**0x1B**
If enabled, include the **Destination Address** in MANT PDUs originating from this IND.		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**Application PDU Timestamp Service**	Type	**0x28**
If enabled, the IND will inspect the payload of Self-Reporting Protocol messages to determine if the application layer PDU contains a timestamp. If the PDU is missing a timestamp, the IND will insert one if it has a valid clock, or it will set the TSSR flag in the containing MANT PDU.		Default	0
		Range	0 - 1
		Access	RW
		Support	Recommended
		Feature Set	Encode

Parameter	**Hop Limit**	Type	**0x40**
The value placed in the Hop Limit field of MANT PDUs newly created by this IND. 0 - 6 : Maximum number of times this message may be repeated 7 : Hop Limit disabled (unlimited repeat count)		Default	1
		Range	0 - 7
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**TDMA Frame Length**	Type	**0x48**
TDMA Frame Length, in milliseconds; also constrained to be 1) an integral multiple of the IND's minimum slot length, and 2) evenly divisible into twelve hours		Default	15,000
		Range	5,000 - 3,600,000
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**TDMA Slot Length**	Type	**0x4A**
TDMA Slot Length, in milliseconds; also constrained to be an integral multiple of the IND's minimum slot length		Default	1,000
		Range	250 - 10,000
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**TDMA Slot Start Offset**	Type	**0x4B**
Number of milliseconds into the TDMA frame this IND starts its TDMA slot. Maximum value is determined by TDMA Frame Length - TDMA Slot Length.		Default	0
		Range	Varies
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**TDMA Slot Padding**	Type	**0x4E**
Number of milliseconds of padding to use in TDMA slots to account for clock drift. The more padding, the less accurate the clock needs to be.		Default	25
		Range	12 - 250
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**TDMA Center Transmission**	Type	**0x4F**
If enabled, transmission is centered in TDMA slot. If disabled, transmission begins at the start of the TDMA slot.		Default	0
		Range	0 - 1
		Access	RW
		Support	Optional
		Feature Set	Encode

Parameter	**Enable TDMA**	Type	**0x50**
Enable TDMA for media access. If not set, the IND will use random access mode.		Default	1
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**TDMA Slot Overrun Behavior**	Type	**0x51**
Determines behavior of IND when TDMA slot would be overrun: 0: Drop 1: Overrun		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**TDMA Bytes Remaining**	Type	**0x52**
Returns the maximum MANT payload allowable in the current TDMA slot assuming a single MANT protocol message and the current configuration.		Default	-
		Range	-
		Access	RO
		Support	Mandatory
		Feature Set	Encode

Parameter	**GPS Update Period**	Type	**0x4C**
The frequency, in minutes, with which the IND will attempt to synchronize its clock using the GPS system.		Default	30
		Range	5 - 1440
		Access	RW
		Support	Mandatory
		Feature Set	GPS

Parameter	**GPS Update Timeout**	Type	**0x4D**
The amount of time, in minutes, to attempt to get a valid GPS clock signal before giving up. Maximum value is **GPS Update Period** minus 1 minute.		Default	5
		Range	Varies
		Access	RW
		Support	Mandatory
		Feature Set	GPS

Parameter	**Get GPS Leap Seconds**	Type	**0x71**
Return the number of leap seconds used by the GPS for clock adjustment, or 0 if the correction is unknown.		Default	-
		Range	-
		Access	RO
		Support	Recommended
		Feature Set	GPS

Parameter	**IND Time - Day / Millisecond Format**	Type	**0x7C**
Query or set the clock on the IND. Time is specified and returned as a 4-Byte unsigned unsigned integer containing the number of days since Jan 1, 1970, UTC, minus leap seconds, followed by a 4-Byte unsigned integer containing the number of milliseconds since midnight UTC.		Default	-
		Range	-
		Access	RW
		Support	Mandatory
		Feature Set	Core

Parameter	**IND Time - Extended Format**	Type	**0x7D**
Query or set the clock on the IND. Time is specified and returned as an unsigned integer containing the number of milliseconds since midnight, Jan 1, 1970, UTC, minus leap seconds.		Default	-
		Range	-
		Access	RW
		Support	Mandatory
		Feature Set	Core

Parameter	**IND Time - Seconds Since 2010**	Type	**0x7E**
Query or set the clock on the IND. Time is returned as a 4-byte unsigned integer containing the number of seconds since midnight, Jan 1, 2010, UTC. Deprecated in favor of **IND Time - Extended Format**.		Default	-
		Range	-
		Access	RW
		Support	Deprecated
		Feature Set	Core

Parameter	**Clock Status**	Type	**0x7F**
0x0 = accurate time; TDMA is being used for media access 0x2 = stale time (time has been set since power on, but has since drifted); random mode is being used for media access. 0x3 = time has not been set since power on; random mode is being used for media access. 0x4 = clock is accurate for timestamping data, but not sufficient for TDMA mode (for example, NTP); random mode is being used for media access.		Default	-
		Range	0, 2-4
		Access	RO
		Support	Mandatory
		Feature Set	Core

Parameter	**Carrier Only Time**	Type	**0x60**
Time, in milliseconds, that carrier only signal will be sent after PTT is enabled to allow the radio to warm up. It is recommended that APDs set carrier only time to 0, and instead use AGC time. This parameter is preserved for backwards compatibility.		Default	5
		Range	0 - 1,000
		Access	RW
		Support	Deprecated
		Feature Set	Encode

Parameter	**AGC Time**	Type	**0x61**
Time, in milliseconds, that a sine wave signal will be sent to allow automatic gain algorithms to work.		Default	55
		Range	5 - 1,000
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**RF Tail Time**	Type	**0x62**
Time, in milliseconds, that the PTT signal remains enabled after the message is sent.		Default	5
		Range	0 - 100
		Access	RW
		Support	Recommended
		Feature Set	Encode

Parameter	**Invert Modulation**	Type	**0x63**
Invert modulation of audio sent to transmit radio. It is recommended that ALERT2 decoders implement support for both modulation polarities, however, some decoders may not provide this support. In this case, modulation polarity must be configured. 0x0 = normal modulation 0x1 = inverted modulation		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**Transmit Radio Always On**	Type	**0x65**
If enabled, transmit radio will be powered whenever the IND has power.		Default	-
		Range	-
		Access	RW
		Support	Recommended
		Feature Set	Encode

Parameter	**Transmit Radio Warm Up Time**	Type	**0x66**
Number of milliseconds that the transmit radio should be powered on before sending the PTT signal and audio data.		Default	750
		Range	25 - 2000
		Access	RW
		Support	Mandatory
		Feature Set	Encode

Parameter	**Transmit Audio Modulation Voltage**	Type	**0x68**
Peak to peak voltage level, in millivolts, for transmit audio output. This may be configured via software or via hardware using, e.g., a jumper or a potentiometer.		Default	400
		Range	Varies
		Access	RW
		Support	Mandatory, or HW Supported
		Feature Set	Encode

Parameter	**FEC Mode**	Type	**0x64**
0 = Highest error correction; lowest throughput 1 = Medium error correction; medium throughput 2 = Least error correction; highest throughput		Default	0
		Range	0 - 2
		Access	RW
		Support	Mandatory
		Feature Set	Configurable FEC

Parameter	**Concentration Test Flag**	Type	**0x1E**
If enabled, set the "Test Flag" in the Application Protocol Header for concentration messages originating from this IND.		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	ALERT CCN

Parameter	**Concentration PDU ID**	Type	**0x20**
If enabled, set the "PDU ID Flag" in the Application Protocol Header for concentration messages originating from this IND.		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	ALERT CCN

Parameter	**Status Report Interval (hours)**	Type	**0x56**
Frequency, in hours, of status reports from the IND, or 0 to disable status reports.		Default	1
		Range	0 - 48
		Access	RW
		Support	Mandatory
		Feature Set	Status Report

Parameter	**Status Report Offset (minutes)**	Type	**0x57**
Number of minutes after the start of the hour that the status report will be measured, then queued for the next available TX slot.		Default	0
		Range	0 - 59
		Access	RW
		Support	Mandatory
		Feature Set	Status Report

Parameter	**Last NV-Save Status**	Type	**0x8080**
0 = failure 1 = success in saving configuration to non-volatile storage		Default	-
		Range	0 - 1
		Access	RO
		Support	Mandatory (if HW Supported)
		Feature Set	Core

Parameter	**Address List Selection**	Type	**0x31**
Select address list for processing.		Default	0
		Range	Varies
		Access	RW
		Support	Mandatory
		Feature Set	Repeat

Parameter	**Address List Enabled**	Type	**0x32**
Enable or disable selected address list. Multiple address lists may be active.		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	Repeat

Parameter	**Address List Action**	Type	**0x33**
0 : Pass List 1 : Reject List		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	Repeat

Parameter	**Address List Type**	Type	**0x34**
0 : Source Address List 1 : Destination Address List 2 : Added Path List Address List		Default	0
		Range	0 - 2
		Access	RW
		Support	Mandatory
		Feature Set	Repeat

Parameter	**Address List: Add List**	Type	**0x35**
Takes a list of 2-byte IND Addresses and adds them to the current address list. If an address given is already in the Address List, this will have no effect.		Default	
		Range	
		Access	WO
		Support	Mandatory
		Feature Set	Repeat

Parameter	**Address List: Add Range**	Type	**0x36**
Takes a list of pairs of 2-byte IND Addresses (Low, High) and adds those ranges to the current address list. If an address in the given range is already in the Address List, this will have no effect.		Default	-
		Range	-
		Access	WO
		Support	Mandatory
		Feature Set	Repeat

Parameter	**Address List: Remove List**	Type	**0x37**
Takes a list of 2-byte IND Addresses and removes them from the current address list. If an address given is not in the Address List, this will have no effect.		Default	-
		Range	-
		Access	WO
		Support	Mandatory
		Feature Set	Repeat

Parameter	**Address List: Remove Range**	Type	**0x38**
Takes a list of pairs of 2-byte IND Addresses (Low, High) and removes those ranges from the current address list. If an address in the given range is already in the Address List, this will have no effect.		Default	-
		Range	-
		Access	WO
		Support	Mandatory
		Feature Set	Repeat

Parameter	**Address List: Query**	Type	**0x3B**
Returns contents of address as a series of ranges (2-byte low address, 2-byte high address).		Default	-
		Range	-
		Access	RO
		Support	Mandatory
		Feature Set	Repeat

Parameter	**Report Rejected Messages**	Type	**0x3A**
If enabled, the IND will include messages that are rejected by any active address list in its Message API output. If disabled, these messages will be ignored. Regardless of this setting, the IND shall not retransmit AirLink messages that are rejected by an activate address list.		Default	1
		Range	0 - 1
		Access	RW
		Support	Optional
		Feature Set	Repeat

Parameter	**Echo Suppression**	Type	**0x3F**
Before repeating a MANT PDU, check for this IND's address in the Added Path List. If present, do not repeat MANT.		Default	1
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	Repeat

Parameter	**Add Path Override**	Type	**0x39**
If enabled, set the "Add Path Service" flag and add this INDs address to all repeated MANT PDUs, regardless of the state of the "Add Path Service" flag on the incoming MANT PDU.		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	Repeat

Parameter	**EERDS Enable**	Type	**0x41**
Enable or disable end-to-end reliable datagram service.		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	EERDS

Parameter	**EERDS Retransmit Delay**	Type	**0x42**
Time, in milliseconds, that this IND will wait for a message to be acknowledged before enqueuing it for retransmission when EERDS is enabled.		Default	6,000
		Range	500 - 72,000
		Access	RW
		Support	Mandatory
		Feature Set	EERDS

Parameter	**EERDS Maximum Retransmissions**	Type	**0x43**
Maximum number of retransmissions before the IND gives up on an unacknowledged message.		Default	3
		Range	0 - 10
		Access	RW
		Support	Mandatory
		Feature Set	EERDS

Parameter	**Encrypt Outgoing Messages**	Type	**0x8082**
Enables encryption of outgoing messages.		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	Encrypt, Encode

Parameter	**Encryption: Address to Configure**	Type	**0x8083**
Set address to configure. Address 0 is the General Purpose Key.		Default	0
		Range	0 - 65,534
		Access	RW
		Support	Mandatory
		Feature Set	Encrypt

Parameter	**Encryption: Key Rotation Time**	Type	**0x8084**
4-byte POSIX Time, after which the encryption key will be rotated. - If no Address to Configure (**0x8083**) has been set, this affects the General Purpose Key - Only 1 future key may be defined for each address		Default	-
		Range	-
		Access	RW
		Support	Mandatory
		Feature Set	Encrypt

Parameter	**Encryption: Set Key**	Type	**0x8085**
16-byte string used to encrypt and decrypt messages. - If no Address to Configure (**0x8083**) has been set, this affects the General Purpose Key - If no Key Rotation Time (**0x8084**) has been set, this key is active immediately		Default	-
		Range	-
		Access	WO
		Support	Mandatory
		Feature Set	Encrypt

Parameter	**Encryption: Remove Key**	Type	**0x8086**
Takes a two-byte address, and removes all keys associated with that address.		Default	-
		Range	-
		Access	WO
		Support	Mandatory
		Feature Set	Encrypt

Parameter	**Encryption: EMID**	Type	**0x8087**
Get or set the EMID counter associated with the currently selected address.		Default	-
		Range	0 - 16,777,216
		Access	RW
		Support	Mandatory
		Feature Set	Encrypt

Parameter	**Encryption: List Addresses with Keys**	Type	**0x8088**
Returns a list of two-byte addresses that are associated with an encryption key.		Default	-
		Range	-
		Access	RO
		Support	Mandatory
		Feature Set	Encrypt

Parameter	**Serial Port to Configure**	Type	**0x9007**
Select which serial port to configure. Values range from 0 - N-1, where N is the number of ports available on the IND.		Default	0
		Range	Varies
		Access	RW
		Support	Mandatory
		Feature Set	RS-232

Parameter	**Baud Rate**	Type	**0x9000**
Acceptable Values: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200		Default	57600
		Range	Varies

		Access	RW
		Support	Mandatory
		Feature Set	RS-232

Parameter	**Parity**	Type	**0x9001**
0 : None 1 : Odd 2 : Even If not configurable, parity bit must be None.		Default	0
		Range	Varies
		Access	RW
		Support	Optional
		Feature Set	RS-232

Parameter	**Stop Bits**	Type	**0x9002**
If not configurable, stop bits must be one.		Default	1
		Range	1 - 2
		Access	RW
		Support	Optional
		Feature Set	RS-232

Parameter	**Flow Control**	Type	**0x9003**
0 : None 1 : Hardware 2 : Software If not configurable, defaults to 0 (None).		Default	0
		Range	0 - 2
		Access	RW
		Support	Optional
		Feature Set	RS-232

Parameter	**Timeout**	Type	**0x9006**
Time, in milliseconds, that the serial port will wait for input before timing out. A value of 0 disables the timeout.		Default	250
		Range	0 - 5000
		Access	RW
		Support	Optional
		Feature Set	RS-232

Parameter	**Serial Port Input Mode**	Type	**0x9008**
0 = API 1 = ALERT Concentration NOTE: Setting ALERT Concentration mode will prevent this port from being used for ALERT2 API input. It is recommended that users disable output from an ALERT Concentration serial port, using TLV 0x900B, Output Mode.		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	ALERT CCN

Parameter	**Independent Addressing Enabled**	Type	**0x9009**
If enabled, MANT PDUs originating from the selected serial port (see TLV 0x9007) will use the address from the **Port Address** field. Otherwise, the IND address will be used.		Default	0
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	RS-232

Parameter	**Port Address**	Type	**0x900A**
See **Independent Addressing Enabled**		Default	9000
		Range	1 - 65,534
		Access	RW
		Support	Mandatory
		Feature Set	RS-232

Parameter	**Output Mode**	Type	**0x900B**
0 : Disabled 1 : Binary 2 : CSV		Default	1
		Range	0 - 2
		Access	RW
		Support	Mandatory
		Feature Set	RS-232

Parameter	**Current Serial Port Number**	Type	**0x900C**
Returns the number of the current serial port		Default	-
		Range	Varies
		Access	RO
		Support	Mandatory
		Feature Set	RS-232

Parameter	**DHCP Enabled**	Type	**0x9010**
0: Disabled 1: Enabled		Default	1
		Range	0 - 1
		Access	RW
		Support	Mandatory
		Feature Set	TCP/IP

Parameter	**IP V4 Address (Static)**	Type	**0x9011**
The IP address to use if DHCP is disabled, in dotted decimal format.		Default	-
		Range	15 Byte String
		Access	RW
		Support	Mandatory
		Feature Set	TCP/IP

Parameter	**IP V4 Subnet Mask**	Type	**0x9012**
The subnet mask to use if DHCP is disabled, in dotted decimal format.		Default	-
		Range	15 Byte String
		Access	RW
		Support	Mandatory
		Feature Set	TCP/IP

Parameter	**IP V4 Gateway**	Type	**0x9013**
The gateway to use if DHCP is disabled, in dotted decimal format.		Default	-
		Range	15 Byte String
		Access	RW
		Support	Mandatory
		Feature Set	TCP/IP

Parameter	**IP V4 DNS Servers**	Type	**0x9014**
Comma separated list of hostnames or IP address to be used for DNS lookups, if DHCP is not enabled.		Default	-
		Range	255 Byte String
		Access	RW
		Support	Mandatory
		Feature Set	TCP/IP

Parameter	**IP V6 Address (Static)**	Type	**0x9018**
The IP address to use if DHCP is disabled, in colon separated hextet form.		Default	-
		Range	39 Byte String
		Access	RW
		Support	Mandatory
		Feature Set	TCP/IP

Parameter	**IP V6 Subnet Mask**	Type	**0x9019**
The subnet mask to use if DHCP is disabled, in colon separated hextet form.		Default	-
		Range	39 Byte String
		Access	RW
		Support	Mandatory
		Feature Set	TCP/IP

Parameter	**IP V6 Gateway**	Type	**0x901A**
The gateway to use if DHCP is disabled, in colon separated hextet form.		Default	-
		Range	39 Byte String
		Access	RW
		Support	Mandatory
		Feature Set	TCP/IP

Parameter	**IP V6 DNS Servers**	Type	**0x901B**
Comma separated list of hostnames or IP address to be used for DNS lookups, if DHCP is not enabled.		Default	-
		Range	255 Byte String
		Access	RW
		Support	Mandatory
		Feature Set	TCP/IP

Parameter	**Clock Source**	Type	**0x9015**
0: GPS 1: NTP 2: Disabled		Default	0
		Range	0 - 2
		Access	RW
		Support	Recommended
		Feature Set	Core

<table>
<tr><td>Parameter</td><td>NTP Servers</td><td>Type</td><td>0x9016</td></tr>
<tr><td colspan="2" rowspan="6">Comma separated list of hostnames or IP address to be used for NTP services.</td><td>Default</td><td>-</td></tr>
<tr><td>Range</td><td>255 Byte String</td></tr>
<tr><td>Access</td><td>RW</td></tr>
<tr><td>Support</td><td>Recommended</td></tr>
<tr><td>Feature Set</td><td>TCP/IP</td></tr>
</table>

<table>
<tr><td>Parameter</td><td>Current IP Address</td><td>Type</td><td>0x9017</td></tr>
<tr><td colspan="2" rowspan="6">Returns the device’s current IP address as allocated by the DHCP server or the static IP if DHCP is disabled. IP addresses are formatted in dotted decimal format (IPV4) or colon separated hextet format (IPV6). Returns an empty string if no IP address is configured. May return a comma separated list of IPV4 and IPV6 addresses if multiple addresses are assigned.</td><td>Default</td><td>-</td></tr>
<tr><td>Range</td><td>255 Byte String</td></tr>
<tr><td>Access</td><td>RO</td></tr>
<tr><td>Support</td><td>Mandatory</td></tr>
<tr><td>Feature Set</td><td>TCP/IP</td></tr>
</table>

8.7. Message API TLVs

Message	**ALERT2 Data Envelope**	Type	**0x10**
Recursive TLV; This type encapsulates received ALERT2 data. It will always contain an Agency ID, Timestamp, Time Quality. If the data was received from a source with a relevant Media Information Envelope (for example, an AirLink source), the associated envelope will be included. If MANT data was successfully decoded, the envelope will contain the MANT PDU envelopes. If any errors are encountered during processing, one or more Message Output Error Envelope(s) will be included		Default	-
		Range	Length Varies
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**AirLink PDU Envelope**	Type	**0x14**
Media Information Envelope for the AirLink protocol. Contains metadata about AirLink transmission, and optionally, full AirLink PDU.		Default	-
		Range	Length Varies
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**MANT PDU Envelope**	Type	**0x15**
Container for MANT PDU and related information.		Default	-
		Range	Length Varies
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**MANT Header**	Type	**0x8400**
MANT header bytes, first byte received, first.		Default	-
		Range	Length Varies
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**MANT Payload**	Type	**0x8401**
MANT payload bytes, first byte received, first.		Default	-
		Range	Varies
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**MANT Authentic**	Type	**0x8404**
This field will be 1 if the MANT PDU was sent using the MANT Encryption and Authentication protocol, and was successfully decrypted and authenticated. This field will be 0 if the MANT PDU has not been authenticated, e.g., if it was sent in plain text or if it was unable to be decrypted or authenticated.		Default	0
		Range	0 - 1
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**MANT Error**	Type	**0x8405**
If an error occurs while decoding the MANT PDU, a MANT Error TLV will be present in the MANT Envelope. MANT Error types are defined as: 0x00: No errors; MANT Error TLV may be omitted in this case 0x01: Inauthentic MANT; The MANT specified that authentication should be used and the IND had an appropriate encryption key, but authentication was unsuccessful 0x02: Concentration protocol message has a length not divisible by 4 0xFF: Freeform or vendor-specific error not covered by the specification; the AirLink Error Description field should be set		Default	-
		Range	Varies
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**MANT Error Description**	Type	**0x8406**
A human-readable string that describes the error		Default	-
		Range	Varies
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**AirLink Header**	Type	**0x844C**
2-Byte AirLink Header		Default	-
		Range	2-Bytes
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**AirLink Payload**	Type	**0x844D**
AirLink Payload, first byte received, first.		Default	-
		Range	Varies
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**AirLink FEC Mode**	Type	**0x844E**
FEC Mode (0 - 2) used to send this AirLink message.		Default	-
		Range	0 - 2
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**AirLink Total Symbol Errors Corrected**	Type	**0x844F**
1-Byte sum of the “AirLink Symbol Errors Corrected” array (TLV type 0x8450), or 255 if any block contained uncorrectable errors.		Default	-
		Range	0 - 255
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**AirLink Symbol Errors Corrected**	Type	**0x8450**
Array of bytes with length equal to the number of blocks in the AirLink message. Each byte indicates the number of symbol errors corrected in the associated block, or -1 if uncorrectable errors were present in the block. See the "Reed-Solomon Coding" section of the AirLink specification for more information.		Default	-
		Range	Varies
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**AirLink Frame Length**	Type	**0x8451**
Total length of AirLink frame, including two-byte header.		Default	-
		Range	0 - 1023
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**AirLink Noise Level**	Type	**0x8452**
Noise level of received AirLink PDU; 0-100. Different devices may perform decoding differently; the value and means of computing of the noise level parameter may vary from vendor to vendor or product to product.		Default	-
		Range	0 - 100
		Access	RO
		Support	Recommended
		Feature Set	Decode

Message	**Number MANT PDUs Successfully Decoded**	Type	**0x8453**
Number of MANT PDUs successfully decoded in this AirLink frame		Default	-
		Range	0 - 255
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**AirLink Error**	Type	**0x8454**
If an error occurs while decoding the AirLink message, an AirLink error TLV will be present and the AirLink Header and AirLink Payload TLVs should be populated. AirLink Errors types are defined as: 0x00: No errors; AirLink Error TLV may be omitted in this case 0x01: Bad AirLink first block; AirLink message was undecodable 0x02: Uncorrectable symbol errors in contained MANT header; one or more MANT PDUs were not decoded 0x03: Uncorrectable symbol errors in contained MANT payload; one or more MANT PDUs were not decoded 0x04: MANT PDU length exceeds AirLink length 0x05: Not enough data for MANT header 0x06: Invalid MANT header; one or more MANT PDUs was not decoded 0xFF: Freeform or vendor-specific error not covered by the specification; the AirLink Error Description field should be set		Default	-
		Range	0x00 - 0xFF
		Access	RO
		Support	Mandatory
		Feature Set	Decode

Message	**AirLink Error Description**	Type	**0x8455**
A human-readable string that describes the error.		Default	-
		Range	Varies
		Access	RO
		Support	Mandatory
		Feature Set	Decode

9. API Transport Methods

Both the IND API and the messaging API are suitable for transport over various media types. This section describes specific details of the transport types commonly in use.

9.1. ALERT2 Prefix String

Except for the "ALERT Compatibility Mode Interface", when data is exchanged with an IND over a transport interface that does not provide explicit segmentation of messages (e.g, over a serial port or a TCP socket), data must be formatted as strings constructed as follows:

- Begin with the ASCII character string prefix of "AL2",
- Followed by a single ASCII version character of "2",
- Followed by a single ASCII format character of "b" for binary or "a" for ASCII (Message Type API - CSV Format),
- Then, if using the binary API format,
  - a binary Total Length number (one or two bytes long) specifying the total length of the string following the Total Length number field,
  - TLV data.

The version character shall match the version reported by API TLV 0x75.

The total length number shall be encoded as an extensible value, such that, when the most significant bit is set, it shall be parsed at a 2-byte number, or a 1-byte number otherwise. Values decimal 0 to 127 are encoded as a single byte, and values from 127 to 32767 are encoded in the least significant bits of a 2-byte integer with the most significant bit set.

9.2. ALERT2 Asynchronous Serial Interface

For any ALERT2 IND implementing an asynchronous serial interface (UART), the following specifications apply.

The asynchronous serial port configuration must be:

- TIA RS-232-F standard DCE circuit and signal levels, using either a 3-wire or 5-wire configuration,
- 8 data bits, and
- least significant bit sent first.

The initial (power on or reset default) configuration parameters must be:

- 3-wire RS-232 circuit (no hardware handshaking),
- No software handshaking,
- 57600 baud,
- 8 data bits,
- no parity,
- 1 stop bit.

In order to retain communications with an IND during a serial port settings update, the settings should be sent in a single Set command, so the updated settings are applied at once.

9.3. ALERT2 OTA Configuration and Control Protocol

The MANT protocol defines a method for sending IND API commands to remote INDs over the ALERT2 network. To send a MANT Configuration and Control Protocol message to an IND, the APD uses command type 0x02; the IND then forms and sends a MANT PDU using port ID 0x8. Prior to enqueuing the configuration and control message, the destination address (type 0x19) and the include destination address (type 0x1B) parameters must be configured appropriately on the originating IND.

It is strongly recommended that ALERT2 INDs not process Configuration and Control protocol messages unless the requesting MANT PDU has been shown to be authentic using the MANT Encryption and Authentication protocol.

Because the MANT Configuration and Control protocol provides message framing information, the message prefix and total message length fields are not present in the MANT payload of OTA Configuration and Control Protocol messages.

9.4. ALERT2 TCP/IP Interfaces

While there is currently no standard defining the process, API data may be exchanged with an IND via TCP/IP. It is strongly recommended that any process for data exchange over a TCP/IP network implement encryption and authentication.

One common method of accepting data is via a raw TCP/IP socket. Encryption of TCP/IP data over a raw socket is possible with SSL, however without addressing authentication, encryption alone is of limited value. It is therefore recommended that any IND accepting TCP/IP over a raw socket do so only with a private, non-routable IP address that is not accessible from the general Internet.

For sending API data over the general Internet, one viable solution is that the data be sent in an SSH session. SSH provides encryption and authentication, and there are various commercial and freely available implementations.

9.5. Asynchronous Serial - ALERT Concentration

In order to support compatibility with legacy ALERT systems, an asynchronous serial interface may optionally support ALERT-style input instead of any of the API interfaces mentioned in this specification. See ALERT Concentration for more information.

10. Message API - CSV Format

The binary message API is an efficient and extensible format for machine to machine data exchange, but it is not suitable for human consumption or interaction. In order to allow for human observation of data in real time and analysis in a wide range of tools, ALERT2 data can be encoded in a comma-separated value (CSV) format in addition to the binary message API format.

Records in the CSV format will follow standard (RFC4180) CSV definitions. However, because ALERT2 data is often streamed rather than stored in files, different record types may be mixed and header rows may appear in the middle of the stream as well as at the start. Unless otherwise specified, field values will match the equivalent values from the binary output format. Other than the "Raw Data" fields, all fields shall be reported as human readable strings or as decimal numbers.

INDs should output appropriate header rows at a regular interval. In the "Message Type" field, the header rows should contain the appropriate message type as well as the string "Message Type". This will allow for the header rows to be included in filtering of the CSV stream using the desired message type as the filter.

Future extensibility is accomplished by appending new fields to the end of the record, while leaving all existing fields in place. In the case that a field cannot be populated or doesn't apply to a given message, it should be left blank.

10.1. Common data

All lines in the CSV format start with the same data, forming a record header. The header consists of the following fields.

1. Prefix String - AL2<version>a
2. Decode Timestamp - `YYYY-mm-dd HH:MM:SS.fff` format
3. Time Sync Status
4. Agency Identifier
5. Decoding IND Address
6. [Type] Message Type - AirLink, MANT, ALERT CCN, Sensor

10.2. AirLink

AirLink messages will contain the following, following the record header:

1. AirLink Version
2. AirLink Reserved Bits
3. AirLink Frame Length
4. AirLink Total Symbol Errors Corrected
5. AirLink Error ID (0 if no error)
6. AirLink Error Description
7. Raw Data - A single string of colon separated hexadecimal bytes (e.g., "0D:AB:14:52")
8. Block Error List - A slash separated list of decimal error counts (e.g., "0 / 4 / -1 / 3")
9. FEC Mode
10. Noise Level

10.3. MANT

MANT messages will contain the following, following the record header:

1. MANT Version
2. MANT Protocol ID
3. Time Stamp Request enabled
4. Add Path Service Request enabled
5. MANT Port
6. MANT Payload Encrypted
7. MANT Reserved Bits
8. ACK
9. Hop Limit
10. Source Address
11. Destination Address (or empty if not set)
12. MANT PDU ID (if EERDS enabled)
13. Path Length
14. Repeater Path (if APSR enabled) (“>” separated list)
15. Payload Length
16. Payload
17. MANT Error ID (0 if no error)
18. MANT Error Description
19. MANT Authentic

10.4. ALERT Concentration

ALERT Concentration messages will contain the following, after the record header:

1. ALERT Timestamp
2. ALERT ID
3. ALERT Data Value

10.5. Sensor Reading

Upon receipt of an ALERT2 Self-Reporting protocol PDU, the IND may decode the application layer PDU and present a CSV record for each sensor reading in the application layer PDU. The record for sensor readings contains the following, after the record header:

1. Reading Timestamp
2. Site Address
3. Sensor ID
4. Value

11. Appendix A: ALERT Concentration

To support legacy ALERT network architectures, it is recommended that an IND provide the Application layer Concentration Protocol service and support an asynchronous serial port configurable to accept ALERT decoder output. It is recommended that this port accept standard binary format, enhanced IFLOWS format and legacy ASCII format (see the Application Layer Protocol specification document). When supported, this is named the ALERT Compatibility Mode Interface.

When in this mode, the asynchronous serial port is uni-directional and shall only accept ALERT 4 byte messages with no other framing than asynchronous start and stop bits on each binary byte. There shall be no timing constraint; one or more ALERT 4 byte binary messages may be received at the serial communications speed, or the bytes may be spaced with delays between bytes. This input byte format that shall be accepted is shown below:

byte 0	...	byte 1	...	byte 2	...	byte 3

The asynchronous serial port configuration parameters that shall be configurable in this mode are baud rate, parity, number of stop bits, flow control, and serial port to configure. The “add Checksum” and “enable ACK/NAK” are ignored in this mode.

Once configured into the ALERT Compatibility Mode Interface all serial asynchronous bytes shall be parsed, if possible, into ALERT serial binary information. Some other mechanism (e.g. reconfiguration of the serial port from a different serial port) is necessary to reset the port.

When configured into the ALERT Compatibility Mode Interface, the IND must provide the application layer Concentration Protocol services for the ALERT messages. (See the ALERT2 Application Layer Protocols Specification Document.)

11.1. Example: ALERT Compatibility Binary

byte 0	...	byte 1	...	byte 2	...	byte 3
7B		6F		FC		FF

This is the 4 byte binary for an ALERT message with ID 3067 and data value 1022, sent in the “standard” binary format, e.g. using the encoding:

ALERT Byte 1	0	1	A5	A4	A3	A2	A1	A0
ALERT Byte 2	0	1	A11	A10	A9	A8	A7	A6
ALERT Byte 3	1	1	D4	D3	D2	D1	D0	A12
ALERT Byte 4	1	1	D10	D9	D8	D7	D6	D5

Table 11 ALERT Binary Format

11.2. Example: ALERT Compatibility ASCII

byte 0	...	byte 1	...	byte 2	...	byte 3
35		33		35		36

This is the 4 byte binary for an ALERT message with ID 35 and data value 65, sent in the "ALERT ASCII" format, e.g. using the encoding:

ALERT Byte 1	0	0	1	1	AU 3	AU 2	AU 1	AU 0
ALERT Byte 2	0	0	1	1	AT 3	AT 2	AT 1	AT 0
ALERT Byte 3	0	0	1	1	DU 3	DU 2	DU 1	DU 0
ALERT Byte 4	0	0	1	1	DT3	DT2	DT1	DT0

Table 12 ALERT ASCII Format

11.3. Example: ALERT Compatibility using the Enhanced IFLOWS Format

byte 0	...	byte 1	...	byte 2	...	byte 3
FB		2F		FF		per EIF

This is the 4 byte binary for an ALERT message with ID 3067 and data value 1022, sent in the "Enhanced IFLOWS Format", e.g. using the encoding:

ALERT Byte 1	1	1	A5	A4	A3	A2	A1	A0
ALERT Byte 2	D 0	A1 2	A1 1	A1 0	A9	A8	A7	A6
ALERT Byte 3	D 8	D7	D6	D5	D4	D3	D2	D1
ALERT Byte 4	C 0	C1	C2	C3	C4	C5	D10	D9

Table 13 ALERT EIF Format

Where (C0-C5) are the EIFS Frame Check Sequence bits.

12. Glossary

Abbreviation	Description
APD	Application Protocol Device – a device that implements the application layer protocols
API	Application Programming Interface – the means and specifications for communication between programs
APSR	Add Path Service Request – a 1-bit field in the MANT header used to request that each IND add its source address as it forwards a frame
DA	Destination Address – the Source Address of the IND to which a PDU is directed
DAI	Destination Address included in header – a 1-bit MANT header field used to indicate that the destination address is added to the header

EERDS	End-to-End Reliable Datagram Service – a MANT protocol used to confirm delivery of application PDUs
FEC	Forward Error Correction
IND	Intelligent Network Device – A device that implements both the AirLink and MANT protocols, e.g., a modulator/encoder, a demodulator/decoder, or a MODEM
MANT	The middle layer of the ALERT2 3-layer protocol stack. It is responsible for network and transport services
PDU	Protocol Data Unit – a unit of data containing a control header and a data payload that is exchanged between peer layers
SA	Source Address – the 16 bit identifier of the originating IND
TSSR	Time Stamp Service Request - a 1-bit MANT header field used to request that the receiving IND add a timestamp to certain MANT PDUs
UTC	Universal Coordinated Time, also known as Greenwich Mean Time (GMT)