MISB MOTION IMAGERY STANDARDS BOARD	STD 0903.3
Standard	**04 October 2012**
Video Moving Target Indicator and Track Metadata	

Contents

1 Scope

This Standard defines Local Data Sets (LDSs) that may be used to deliver Video Moving Target[1] Indicator (VMTI) metadata[2] and related Track metadata in accordance with SMPTE (Society of Motion Picture Television Engineers) ST 336:2007 [8], which defines an encoding method called "Key-Length-Value" (KLV).

This document also provides abstract data models for the motion imagery VMTI and Track metadata (Appendix C and Appendix D, respectively). These data models, which are defined using the Unified Modeling Language (UML), are provided in anticipation of defining future additional encoding methods, such as, Extensible Markup Language (XML). Definition of an abstract data model provides a consistent, common specification, regardless of the encoding method used. However, the KLV encoding specified herein is currently the normative definition for this Standard.

This document also lays out the relationship between this Standard and other relevant Standards, and provides implementation guidance.

The intent is to provide VMTI and Track metadata to downstream clients for the purpose of populating Situational Awareness products and Common Operating Pictures, for generating VMTI and Track overlays on video players, and for providing input to tracking and data fusion systems (*e.g.*, STANAG 4676 *ISR Tracking* compliant systems). In the interests of data efficiency, these LDSs only include elements relevant to VMTI and Tracks that are not available in any other Universal or Local Data Set. For example, it does not include or repeat the sensor model parameters detailed in MISB Standard 0601 *UAS Datalink Local Data Set* [6] (henceforth, "*0601*").[3]

2 References

This Standard references the following documents and standards:

2.1 Normative References

[1] DMA TM8358.1: *Datums, Ellipsoids, Grids, and Grid Reference Systems, 20 September 1990*

[2] IEEE 1003.1, *Information Technology - Portable Operating System Interface (POSIX), 2004*

[3] ISO/IEC 13818-1, *Information technology - Generic Coding of Moving Pictures and Associated Audio Information Part 1: Systems, 2007*

---

1 The term "target", rather than "object", is used for a mode of radar operation that discriminates a "target" from its background. Commonly, radar Doppler shift is used to detect moving objects, resulting in "moving target indications", MTI. In this context, "target" has become widely adopted and is used here for consistency.

2 VMTI metadata is analogous to, but distinct from, NATO STANAG 4607 radar-derived Ground Moving Target Indicator (GMTI) metadata. VMTI metadata contains elements to describe imagery-derived characteristics (e.g., color, shape, features, and unique identity) for which STANAG 4607 GMTI has no counterpart.

3 MISB Standard 0601 *UAS Datalink Local Data Set* and NATO STANAG 4609 Standard 0801 *UAS Datalink Local Data Set* are for all intents and purposes identical, although the MISB Standard can sometimes be a version ahead of the NATO Standard because of latency in the ratification process. STANAG 4609 Standard 0801 is not to be confused with MISB EG 0801 *Photogrammetry Metadata Set for Digital Motion Imagery*.

[4] ISO 19136:2007 *Geographical Information - Geography Markup Language (GML)*

[5] MISB STD 0807, *KLV Metadata Dictionary (latest version)*

[6] MISB STD 0601, *UAS Datalink Local Data Set (latest version)*

[7] MISB RP 0701 *Common Metadata System: Structure, Aug 2007*

[8] SMPTE ST 336:2007 *Data Encoding Protocol Using Key-Length-Value*

[9] SMPTE ST 335:2001 *Television - Metadata Dictionary Structure*

[10] SMPTE RP 210.11:2007 *Metadata Dictionary Registry of Metadata Element Descriptions*

[11] ISO/IEC 646:1991, *Information Technology - ISO 7-bit coded character set for information interchange*

[12] MISB STD 0107 *Bit and Byte Order for Metadata in Motion Imagery Files and Streams*, Oct 2001

[13] Internet Standard Request for Comments (RFC) 3986 *Uniform Resource Identifier (URI): Generic Syntax*

[14] MISB RP 1204 *Motion Imagery Identification System*

[15] MISB RP 1201 *Floating Point to Integer Mapping*

[16] ISO/IEC 9834-8:2005 *Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components*

2.2 Informative References

[17] MISB EG 0104.5, *Predator UAV Basic Universal Metadata Set, Dec 2006*

[18] MISB STD 0102, *Security Metadata Universal and Local Data Sets for Digital Motion Imagery (latest version)*

[19] MISB STD 0604, *Time Stamping and Transport of Compressed Motion Imagery and Metadata (latest version)*

[20] MISB TRM 1007, *Key-Length-value (KLV) User's Guide (latest version)*

[21] STANAG 4676 *NATO Intelligence, Surveillance and Reconnaissance Tracking Standard, EDITION 1 (DRAFT)*, August 2011

[22] Allied Engineering Documentation Publication AEDP-12 *NATO Intelligence, Surveillance and Reconnaissance (ISR) Tracking Standard (NITS) STANAG 4676 Implementation Guide, Edition 1 (DRAFT)*, October 2011

3 Modifications and Changes

Date	Release	Change Summary
10/02/2012		Version 3.0 – The document is reorganized and expanded to include

Date	Release	Change Summary
		motion imagery derived Track metadata. • Retitled document *Video Moving Target Indicator and Track Metadata*. • Added data elements Target_Centroid_Pixel_Row and Target_Centroid_Pixel_Column to the VMTI LDS, for use as an alternative to Target_Centroid_Pixel_Number to avoid the need for conversion, albeit at an increase in size. • Added data element WAMI_FPA_Index to the VMTI LDS, to specify which Focal Plane Array (FPA) in an array of Wide Area Motion Imagery (WAMI) FPAs is the "frame" in which a VTarget detection occurred. • Added the VTrack LDS to provide a "track-centric" representation of detections, a series of VTrackItems, which comprise a track. VTrack data elements are generally aligned with elements of the data model for NATO STANAG 4676 Edition 1 (DRAFT), the emerging NATO ISR Tracking Standard, although as of this date additional harmonization of the two standards is required. With few exceptions, the data elements of the VTrack LDS are the same as those defined for the VMTI LDS. • Moved paragraphs describing Key-Length-Value (KLV) to Appendix A. • Moved the definition of KLV Complex Types used for the VMTI LDS and the VTrack LDS into Appendix B. • Added informative UML data models for the VMTI LDS and the VTrack LDS (Appendix C and Appendix D, respectively). • Added an informative mapping between the STANAG 4676 Edition 1 (DRAFT) data model and the VTrack LDS data model (Appendix E). • Moved paragraphs describing operational considerations to Appendix F.
04/22/2011		Version 2.0 – Clarifications and typo corrections • Corrected erroneous cross-references, incorrect values in examples, and assorted typographical errors. • Modified integer encoding examples in Section 1.1 to better illustrate when to choose signed versus unsigned encoding. • Added clarifying language to Location (Section 1), Velocity (Section 1), and Acceleration (Section 1), stating that truncation is allowed only at a group boundary and filler values are not allowed. • Added clarifying language to Boundary (Section 1), stating that vertex Location elements need not all be of the same length. Removed the requirement to close the Boundary by re-specifying the first vertex. • Assigned a 16-byte Universal Label to VTargetSeries (Section 7.10.13). • Restored Tags for VObject, VFeature, and VTracker to the

Date	Release	Change Summary
		values assigned in EG 0903.0. Assigned a new Tag to VChip. (Note that VObject is redefined to be target type or class, and VChip assumes the earlier definition of VObject, image chip.) • Updated References
02/24/2011	1.0	Version 2.0 – For Review and Approval
01/17/2011	0.9	Version 2.0 – Final Draft

4 Introduction

MISB Standard 0601 *UAS Datalink Local Data Set* ("*0601*") [6] has become the accepted standard Local Data Set (LDS) within Defense agencies for the transmission of metadata elements within motion imagery streams. *0601* includes numerous individual elements and a few local data subsets but, to date, none of these elements or subsets allows the effective inclusion of VMTI or Track metadata.

VMTI and tracks may be indicators of adversarial activities. As such they represent a source of "geospatial transaction" that can be a rich source of intelligence for unraveling adversarial activities and networks. With the increasing appreciation of the importance of tracks and associated indicators of motion, it has become important to provide standard ways to describe and disseminate this type of information so it can be effectively shared, reused, and amplified in the course of a mission or ongoing intelligence analysis.

This document provides a set of standards for tracks and associated indicators of motion for the motion imagery domain. It specifies the formats for reporting indications of the motions of entities, the history of their motions, and the types of the entities being reported. In addition, it provides ways to disseminate the evidential support for inferred movements, tracks, and the characteristics of entities observed in the motion imagery stream. As such, it is designed both for stand-alone roles for motion imagery-based analytics, and for roles requiring conjunctive analytics with other sensor modalities and within multi-source methodologies.

The role of this Standard, therefore, is to define LDSs for VMTI and Track metadata that complement *0601* metadata. Although VMTI and Track information are derived from video, they are meaningful as products in their own rights. Therefore, the VMTI and VTrack LDSs are designed to be transmitted "standalone", independent of any motion imagery essence. Furthermore, the transport stream is permitted to contain just a "naked" VMTI LDS or a VTrack LDS, which contains no other metadata (the *0601* LDS, in particular).

5 Operational Considerations

The VMTI LDS is designed to support a large gamut of systems ranging from those producing thousands of moving targets to those producing great detail about a small number of targets. Its design seeks to use bandwidth efficiently and not replicate or duplicate tags or information. It does not include information that can be calculated, derived, or associated using other information already defined.

The VTrack LDS has similar characteristics.

Relevant operational considerations are described in Appendix F.

6 VMTI Local Data Set Structure

As is the case for all MISB Standards, Recommended Practices (RP), Engineering Guidelines (EG), and Technical Reference Manuals (TRM), *0903* is based upon the SMPTE Standard 336 Key-Length-Value (KLV) construct. We encourage readers unfamiliar with KLV to review MISB TRM 1006 *Key-Length-Value (KLV) Users Guide* [20], which provides a succinct description of those KLV principles particularly relevant to MISB specifications.

Appendix A contains an overview of KLV concepts and constructs that are particularly relevant to this Standard.

6.1 VMTI LDS Structure

VMTI is not a simple set of unrelated metadata elements. It is a complex data system where information exists not just in the data elements, but also in the relationships and structures within which those elements exist. The SMPTE standards and documents covering Key-Length-Value (KLV) encoding of metadata provide the constructs needed to capture these complex data systems in a bandwidth efficient manner.

For bandwidth efficiency, the VMTI LDS contains several data elements expressed as offsets from the frame center geographic coordinate provided in *0601*. Thus, the VMTI LDS is generally paired with a *0601* LDS, and Tag 74 from *0601* has been assigned for the VMTI LDS. However, the VMTI LDS also contains corresponding data elements expressed in absolute geographic coordinates, allowing the VMTI LDS to be independent of *0601*.[4] (Geographic coordinates in the VTrack LDS are absolute, so the VTrack LDS is always independent of *0601*.)

The base structure of the VMTI data system is the VMTI Local Data Set. This LDS contains the core information applicable to all reported phenomena within a video frame. The VMTI LDS makes use of a subordinate VTargetSeries that, in itself, contains one or more VTarget Packs which, in turn, may make use of five subordinate Local Data Sets arranged as shown in Figure 1.

VMTI LDS		
VTargetSeries		
VTarget Pack$_1$	• • •	VTarget Pack$_N$
VMask LDS		VMask LDS
VChip LDS		VChip LDS
VObject LDS		VObject LDS
VFeature LDS		VFeature LDS
VTracker LDS		VTracker LDS

Figure 1. VMTI Local Data Set

[4] The MISB has assigned a full 16-byte key to the VMTI LDS for use in this situation, with value 06.0E.2B.34.02.0B.01.01.0E.01.03.03.06.00.00.00.

Target unique information is contained within the VTargetSeries. VTargetSeries is of type Series (defined in Appendix B), which is a variation on the SMPTE Variable Length Pack.[5] As shown in Figure 2, the VTargetSeries contains one or more VTarget[6] Packs. The metadata associated with individual targets is delivered using VTarget Packs (one for each target). The value field of each VTarget pack is preceded by the short or long BER-encoding of its length. The first, mandatory, element in the value field of each VTarget Pack is a BER-OID encoded value to convey the Target ID Number of the target. The following elements form an LDS-like structure containing one or more TLV triplets that convey information about the target. No particular TLV triplet is mandatory, but at least one must be present.

Note that the VTarget Pack is very unusual in that the first element of the value field must be the BER-OID encoded Target ID Number, without a Tag or a Length, yet the remaining elements adhere to the more conventional TLV structure, with a Tag and a Length. Since the Target ID Number must always be specified, whereas other elements are optional, this construct has been adopted as a bandwidth saving measure. In the simplest case, a VTarget Pack can consist of just the Target ID Number and the Target Centroid Pixel Number.[7]

The TLV triplets may refer to subordinate data structures as nested LDSs. These LDSs include the VMask, VChip, VObject, VFeature, and VTracker Local Data Sets.

In the VMTI LDS packet example shown in Figure 2, the packet begins with a Tag having the value 74. This indicates the presence of a VMTI LDS within the *0601* UAS Datalink LDS. The Tag is followed by a Length value, which is the sum of the lengths of all elements in the VMTI LDS. In this example, the VMTI LDS contains one element with a Tag of 5 (which specifies the Total Number of Targets Detected in the Frame), followed by a VTargetSeries, indicated by Tag 101. The VTargetSeries contains two VTarget Packs. The Value of the first VTarget Pack consists of a BER-OID encoded Target ID # and a Pixel # element (Tag 1, Length 1). The Value of the second VTarget Pack is similar, except it also contains target latitude and longitude elements.

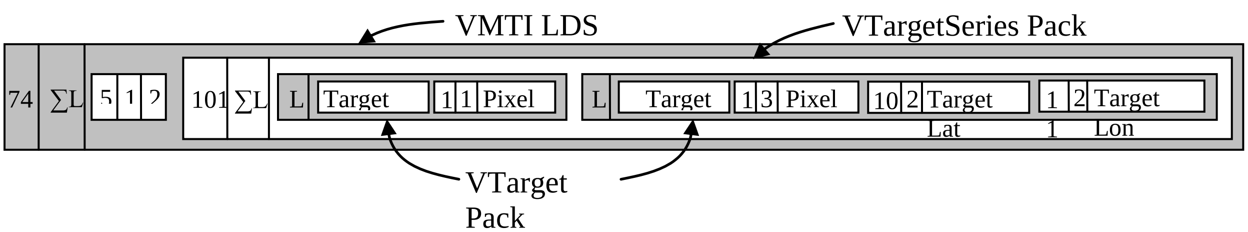


Figure 2. Example VMTI Local Data Set Packet

Figure 3 shows the general format of how the VMTI LDS is configured. It is recommended, but not mandatory, that each VMTI LDS packet contain a Unix-based Coordinated Universal Time

5 No key is required. Each element, a VTarget Pack, consists of only a BER-encoded Length and a Value, which contains metadata elements describing a target.

6 It is assumed that the LDS or Pack name is unique at all levels; therefore references are to VTarget Pack rather than VMTI.VTarget Pack.

7 The Target Centroid Pixel Number can be specified using either the data element Target_Centroid_Pixel_Number or the pair of data elements Target_Centroid_Pixel_Row and Target_Centroid_Pixel_Column, but one representation must appear.

(UTC) Time Stamp that represents the time of birth of the metadata within the LDS packet. (See §6.2.)

It is permitted, but not mandatory, that a Checksum metadata item be included in each VMTI LDS packet. Since the VMTI LDS packet must, except for very exceptional cases, be partnered with an *0601* packet, the Checksum should be omitted in deference to the *0601* Checksum. (See §6.3.)

Any combination of metadata items can be included in a VMTI Local Data Set packet. The items within the VMTI LDS can be arranged in any order except that the Unix-based UTC Time Stamp, if present, must come first, and the Checksum, if present, must come last.

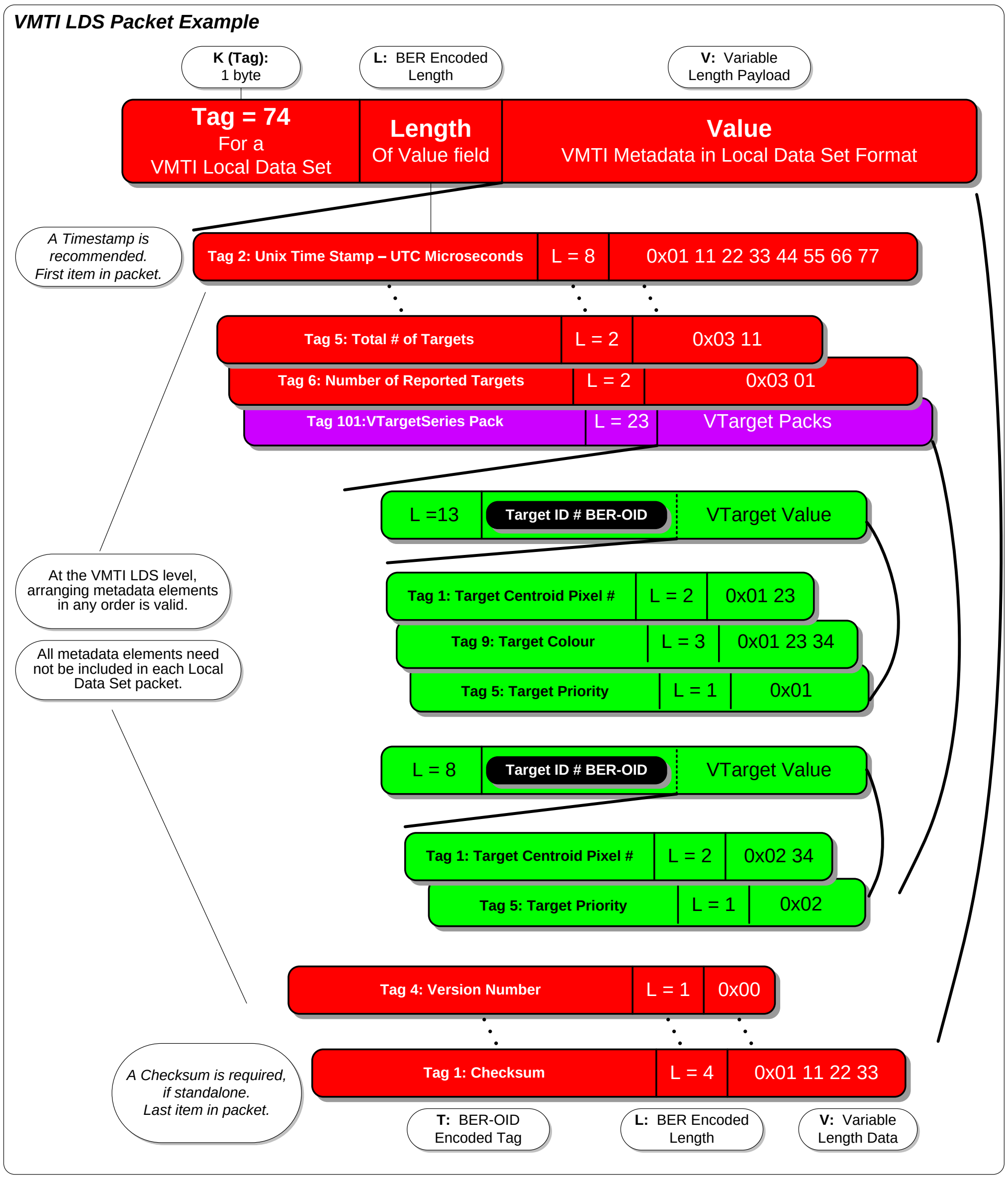


Figure 3. Example VMTI Local Data Set Packet

6.2 Time Stamping

It is recommended that every VMTI LDS packet include a Unix-based UTC time stamp (VMTI LDS Tag 2) as a way to correspond the metadata with a standardized time reference. UTC time is useful to associate metadata with frames, and for reviewing time-critical events at a later date. This section describes how to include a time stamp within a Local Data Set packet.

Metadata sources are coordinated to operate on the same standard time, which is typically GPS derived. The metadata source provides a time stamp for inclusion in an LDS packet and the time stamp assists the accuracy of synchronizing each frame to its corresponding metadata set.

The UTC time stamp, identified by VMTI LDS Tag 2, is an 8-byte unsigned integer that represents the theoretical number of microseconds that have elapsed since midnight, January 1, 1970. This date is known as the UNIX Epoch and is described in the *IEEE POSIX Standard* [2].

It is recommended that the Packet Time Stamp be inserted at the beginning of a VMTI LDS packet. This time stamp corresponds to the time of birth of all the data within the LDS packet. This time can be used to associate the metadata with a particular video frame and can be displayed or monitored appropriately.

An example of an LDS packet containing a time stamp is shown in Figure 4.

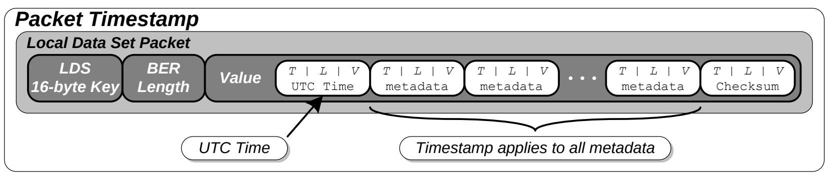


Figure 4. Packet Time Stamp Example

6.3 Error Detection

To help prevent erroneous metadata from being presented, a 16-bit Checksum (VMTI LDS Tag 1) may be included as the last item in every VMTI Local Data Set packet. The Checksum is a running 16-bit sum through the entire LDS packet, starting with the 16-byte Local Data Set Key[8] and ending with summing the Length field of the Checksum data item. Although required in "naked" VMTI LDS packets, it is recommended that when embedded within an *0601* LDS in bandwidth constrained environments, the VMTI LDS Checksum be omitted in deference to the new parent *0601* checksum which has been recalculated as a result of the inclusion.

Figure 5 shows the data range to which the checksum applies for the "naked" case.

[8] If embedded within an *0601* LDS, then the summation starts from the *0601* Tag 74 and ends with the summing of the length field of the Checksum data item.

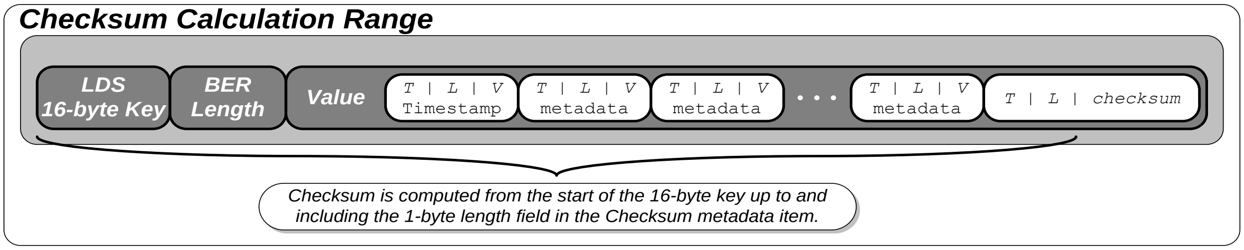


Figure 5. Example Checksum Computation Range for "Naked" VMTI LDS

6.4 Required Tags

The VMTI LDS is permitted to contain a Checksum (VMTI LDS Tag 1). The Checksum is mandatory if the VMTI LDS is not partnered with a *0601* LDS. The UNIX-based UTC Time Stamp (VMTI LDS Tag 2) is recommended, but not mandatory, because it is expected that some VMTI systems will not have access to a time source. Data from such systems is still considered useful even if only aligned with the video by time of arrival.

In the interests of bit efficiency, the VMTI LDS Checksum and UTC Time Stamp may be omitted when the VMTI LDS is embedded within a *0601* LDS, which must always include a time stamp and a checksum.

Other elements may be required under certain circumstances, which are described in various sections throughout this document.

6.5 KLV Complex Types

The KLV coding of data items and groups of data items is defined in SMPTE Standard 336, which requires that each data element is registered in a Metadata Dictionary. The structure of the Metadata Dictionary, defined in SMPTE ST 335 [9] includes the requirement for a Type entry for each data element. In most cases, these are primitive types, such as, integer or string. However, SMPTE provides for more common complex types and allows for the definition of additional complex types.[9] Complex types must have an assigned 16-byte Universal Label (UL) and be registered in a Types Dictionary. *0903* takes advantage of this capability to define some common complex structures that are used in VMTI, as well as other community data models.

The following complex types used in *0903* are defined in Appendix B.

1) Series Type: The Series type is a SMPTE ST 336 Variable-Length Pack Structure containing elements of the same type. In this specification, the VTargetSeries data element is of type Series. VTargetSeries is an unbounded list of VTarget Pack data elements. (Similarly, VTrackItemSeries is a Series of VTrackItem Packs.)
2) Location Type: The Location type describes the geospatial location of an object. This type includes latitude, longitude and height; standard deviations of those measurements; and their correlation coefficients. Locations are structured as a Defined-Length

[9] Common complex types include Universal Sets, Global Sets, Local Sets, Variable Length Packs, Defined Length Packs (formerly called Fixed Length Packs), Truncation Packs, and Floating Length Packs. These are defined in MISB Recommended Practice 0701 [6].

Truncation Pack. There are no Tags or Length fields. Low-priority elements can be omitted from the structure.

3) Velocity Type: The Velocity type describes the velocity of an object along three right-handed Cartesian coordinate axes. This type includes the velocity components in X (West to East), Y (South to North), and Z (vertical); standard deviations of those measurements; and their correlation coefficients. Velocities are structured as Defined-Length Truncation Packs. There are no Tags or Length fields. Low-priority elements can be omitted from the structure.

4) Acceleration Type: The Acceleration type describes the acceleration of an object along three right-handed Cartesian coordinate axes. This type includes the acceleration components in X (West to East), Y (South to North), and Z (vertical); standard deviations of those measurements; and their correlation coefficients. Accelerations are structured as Defined-Length Truncation Packs. There are no Tags or Length fields. Low-priority elements can be omitted from the structure.

5) Boundary Type: The Boundary type describes the geospatial extent of an area or volume of interest. This type consists of Series of Location data elements, one for each vertex of a bounding area or volume. The structure of this type is similar to that of a Variable-Length Pack in that each element has a Length and a Value field. It differs from the Variable-Length Pack in that each element is an instance of the same type. In this specification, the Boundary type is generally used for planar bounding boxes. However, it can support the specification of multifaceted (triangulated) volumes, as well, and it is used in this manner to describe the bounding volume of a set of VMTI detections in VTracker.

7 VMTI Data Element Descriptions

This section provides a description of each VMTI metadata element, its KLV encoding, and the method for encoding and decoding the Value.

A set of tables summarizing the elements defined in this section appears first. We encourage the reader to examine the information in those tables before reviewing the detailed descriptions that follow. A quick look at the Key Name column will provide an excellent overview of the VMTI LDS.

Following the summary tables is a more detailed, structured, descriptive table for each element. Each of these tables consists of two sections. The Data Element Definition section provides a "computer science" view of the data element. It captures the information needed by a software developer or system designer to use this data element. The KLV Encoding section addresses the representation of the data element in KLV and the rules for encoding and decoding the element.

7.1 VMTI Local Data Set

Table 1: VMTI LDS

VMTI LDS							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.0B.01.01 0E.01.03.03.06.00.00.00	VMTI KLV Dictionary	N/A	None	N/A	Variable	This is the Universal Key for the VMTI LDS
01	06.0E.2B.34.04.01.01.01 0E.01.02.03.01.00.00.00	Checksum	Unsigned Integer	None	Uint16	F2	Checksum used to detect errors within a VMTI LDS packet. Defers to *0601* checksum when VMTI LDS packet included inside *0601* packet. Lower 16-bits of summation. Performed on entire VMTI LDS packet, including 16-byte UDS key and 1-byte checksum length.
02	06.0E.2B.34.01.01.01.03 07.02.01.01.01.05.00.00	UNIX Time Stamp	Unsigned Long	Micro-seconds	Uint64	F8	Microseconds elapsed since midnight (00:00:00 UTC), January 1, 1970 (the UNIX Epoch). Derived from the POSIX IEEE1003.1 standard. Resolution: 1 microsecond.
03	06.0E.2B.34.01.01.01.01 0E.01.02.02.7C.00.00.00	VMTI System Name / Description	String	String	ISO7	V32	Text string to allow the inclusion of the name and/or description of the VMTI system.

VMTI LDS							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
04	06.0E.2B.34.01.01.01.01 0E.01.02.05.04.00.00.00	VMTI LDS Version Number	Unsigned Integer	None	Uint16	V2	Version number of the VMTI LDS document used to generate the VMTI metadata. 0 is pre-release, initial release (090x.0), or test data. 1..65535 corresponds to document revisions 1 through 65535.
05	06.0E.2B.34.01.01.01.01 0E.01.02.03.36.00.00.00	Total Number of Targets Detected in the Frame	Unsigned Integer	None	Uint24	V3	The total number of targets detected in the frame. Range 1 to 16,777,215. 0 represents no targets detected – also implied by no value at all and VMTI LDS is superfluous and should be discarded. 16,777,215represents 16,777,215or more detections
06	06.0E.2B.34.01.01.01.01 0E.01.02.03.37.00.00.00	Number of Reported Targets	Unsigned Integer	None	Uint24	V3	The number of targets reported following a culling process. For use, for example, where bandwidth limits the number of targets that can be sent. Range 0 to 16,777,215.
07	06.0E.2B.34.01.01.01.01 0E.01.01.03.1F.00.00.00	Video Frame Number	Unsigned Integer	Frames	Uint24	V3	The video frame number corresponding to the frame in which the targets were detected. Use of the time stamp is preferred but frame number can be used where a time stamp is not available. Range 0 to 2^{24}-1 which equates to approximately 155 hours at 30fps.
08	06.0E.2B.34.01.01.01.01 0E.01.01.02.07.00.00.00	Frame Width	Unsigned Integer	Pixels	Uint24	V3	Width of the video frame in pixels. Range 1 to 16777215 Value of zero is meaningless and should not be used
09	06.0E.2B.34.01.01.01.01 0E.01.01.02.08.00.00.00	Frame Height	Unsigned Integer	Pixels	Unit24	V3	Height of the video frame in pixels. Range 1 to 16777215 Value of zero is meaningless and should not be used

VMTI LDS							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
10	06.0E.2B.34.01.01.01.01 04.20.01.02.01.01.00.00	VMTI Source Sensor	String	String	ISO7	V127	String of VMTI source sensor. *E.g.*, 'EO Nose', 'EO Zoom (DLTV)', 'EO Spotter', 'IR Mitsubishi PtSi Model 500', 'IR InSb Amber Model TBT', 'LYNX SAR Imagery', 'TESAR Imagery', etc. Value field is Free Text. Maximum 127 characters. Similar to *0601* Tag 11 Used to identify the imagery source for the VMTI process if more than one bore-sighted video source is captured simultaneously.
11	06.0E.2B.34.01.01.01.02 04.20.02.01.01.08.00.00	VMTI Sensor Horizontal Field of View	Float	Degrees	Uint16	V2	Horizontal field of view of imaging sensor input to VMTI process. Required if VMTI process is run on a different imaging sensor to that described by the parent *0601* packet. Can be used with HFOV (Tag 16) from *0601* to scale VMTI X,Y coordinates. Map 0..(2^16-1) to 0..180. Resolution: ~2.7 milli degrees.
12	06.0E.2B.34. 01.01.01.07 04.20.02.01.01.0A.01.00	VMTI Sensor Vertical Field of View	Float	Degrees	Uint16	V2	Vertical field of view of imaging sensor input to VMTI process. May be required if VMTI process is run on a different imaging sensor to that included in the MPEG-2 TS [3] . Can be used with VFOV (Tag 17) from *0601* to scale VMTI X,Y coordinates. Typically only required to cater for aspect ratio variation. Map 0..(2^16-1) to 0..180. Resolution: ~2.7 milli degrees.
101	06.0E.2B.34.02.04.01.01 0E.01.03.03.1B.00.00.00	VTargetSeries	Array	None	N/A	V	Series of target metadata, each of which is a. VTarget Pack as defined in Table 2 below. The length field for VTargetSeries is the sum of the lengths of all the contained target metadata. The length field for each VTarget Pack is the size of all elements within that pack including the Target ID #.

7.2 VTarget Pack

Table 2: VTarget Pack

VTarget Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.05.01.01. 0E.01.03.03.07.00.00.00	VTarget Pack	N/A	None	N/A	Variable	This is the Universal Key for the VTarget Pack. This is a Truncation Pack
N/A	N/A	Target ID Number	Unsigned Integer		Uint24	V3	This element is mandatory and it must come first in the VTarget Pack. It is BER-OID encoded to convey the length but has no Tag or Length field. Range 1.. 2,097,151
01	06.0E.2B.34.01.01.01.01 0E.01.02.03.38.00.00.00	Target Centroid Pixel Number	Unsigned Integer	Pixels	Uint48	V6	Defines the position of the target within the video frame in pixels. Range 1 to 2^{48}-1. Numbering commences from one denoting the Top Left pixel
02	06.0E.2B.34.01.01.01.01 0E.01.02.03.39.00.00.00	Bounding Box Top Left Pixel Number	Unsigned Integer	Pixels	Uint48	V6	Defines the position of the top left corner of the target bounding box within the video frame in pixels. Range 1 to 2^{48}-1. Numbering commences from one denoting the Top Left pixel
03	06.0E.2B.34.01.01.01.01 0E.01.02.03.3A.00.00.00	Bounding Box Bottom Right Pixel Number	Unsigned Integer	Pixels	Uint48	V6	Defines the position of the bottom right corner of the target bounding box within the video frame in pixels. Range 1 to 2^{48}-1. Numbering commences from one denoting the Top Left pixel
04	06.0E.2B.34.01.01.01.01 0E.01.02.03.3B.00.00.00	Target Priority	Unsigned Integer	None	Uint8	F1	Priority or validity of target based on criteria within the VMTI system. The target(s) with the highest priority may not have the highest confidence level. Potential for use in limited bandwidth scenarios to only send highest priority targets. Range 1 to 255 where 1 is the highest priority. Multiple targets may have the same priority.

VTarget Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
05	06.0E.2B.34.01.01.01.01 0E.01.02.03.3C.00.00.00	Target Confidence Level	Unsigned Integer	None	Uint8	F1	Confidence level of target based on criteria within the VMTI system. The target(s) with the highest confidence may not have the highest priority value. Potential for use in limited bandwidth scenarios to only send highest confidence targets. Range 0 to 100, as a percentage, where 100 percent is the highest confidence. Multiple targets may have the same confidence level.
06	06.0E.2B.34.01.01.01.01 0E.01.02.03.3D.00.00.00	New Detection Flag / Target History	Unsigned Integer	Frames	Uint16	V2	The number of previous times the same target has been detected. Range 0 to 65535 frames. Where a value of 0 denotes the target as a new detection. Detections are not required to be in consecutive frames.
07	06.0E.2B.34.01.01.01.01 0E.01.02.03.3E.00.00.00	Percentage of Target Pixels	Float	None	Uint8	F1	The percentage of pixels within the bounding box that are detected to be target pixels rather than background pixels. Range 1 to 100, where 100 signifies that the target completely fills the bounding box.
08	06.0E.2B.34.01.01.01.01 0E.01.02.03.3F.00.00.00	Target Color	Unsigned Integer	None	Uint24	F3	Dominant color of the target. For use when metadata is transmitted in the absence of the underlying video. RGB color value. VFeature LDS can be used for more comprehensive color information.
09	06.0E.2B.34.01.01.01.01 0E.01.02.03.40.00.00.00	Target Intensity	Float	None	Uint24	V3	Dominant Intensity of the target. For use when metadata is transmitted in the absence of the underlying video. Range 0..2^{24}-1. VFeature LDS can be used for more comprehensive temperature information.

VTarget Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
10	06.0E.2B.34.01.01.01.01 0E.01.02.03.41.00.00.00	Target Location Latitude Offset	Float	Degrees	Int24	V3	Latitude offset for target from frame center latitude (*0601*). Based on WGS84 ellipsoid. Use with Frame Center Latitude. Map $-(2^{23}-1)..(2^{23}-1)$ to +/- 19.2 degrees. Use $-(2^{23})$ as an "error" indicator. Resolution: ~2.3micro deg, ~0.25 meters at equator. Range: +/- 2136 km.
11	06.0E.2B.34.01.01.01.01 0E.01.02.03.42.00.00.00	Target Location Longitude Offset	Float	Degrees	Int24	V3	Longitude offset for target from frame center longitude (*0601*). Based on WGS84 ellipsoid. Use with Frame Center Longitude. Map $-(2^{23}-1)..(2^{23}-1)$ to +/- 19.2 degrees. Use $-(2^{23})$ as an "error" indicator. Resolution: ~2.3micro deg, ~0.25 meters at equator. Range: +/- 2136 km.
12	06.0E.2B.34.01.01.01.01 0E.01.02.03.43.00.00.00	Target Height	Float	Meters	Uint16	V2	Height of target in meters above WGS84 Ellipsoid. Map $0..(2^{16}-1)$ to -900..19000 meters. Resolution: ~0.3 meters.
13	06.0E.2B.34.01.01.01.01 0E.01.02.03.44.00.00.00	Bounding Box Top Left Latitude Offset	Float	Degrees	Int24	V3	Latitude offset for top left corner of target bounding box. Use with *0601* Frame Center Latitude. Map $-(2^{23}-1)..(2^{23}-1)$ to +/- 19.2 degrees. Use $-(2^{23})$ as an "error" indicator. Resolution: ~2.3micro deg, ~0.25 meters at equator. Range: +/- 2136 km.
14	06.0E.2B.34.01.01.01.01 0E.01.02.03.45.00.00.00	Bounding Box Top Left Longitude Offset	Float	Degrees	Int24	V3	Longitude offset for top left corner of target bounding box. Use with *0601* Frame Center Longitude. Map $-(2^{23}-1)..(2^{23}-1)$ to +/- 19.2 degrees. Use $-(2^{23})$ as an "error" indicator. Resolution: ~2.3micro deg, ~0.25 meters at equator. Range: +/- 2136 km.

VTarget Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
15	06.0E.2B.34.01.01.01.01 0E.01.02.03.46.00.00.00	Bounding Box Bottom Right Latitude Offset	Float	Degrees	Int24	V3	Latitude offset for bottom right corner of target bounding box. Use with *0601* Frame Center Latitude. Map -(2^{23}-1)..(2^{23}-1) to +/- 19.2 degrees. Use -(2^{23}) as an "error" indicator. Resolution: ~2.3micro deg, ~0.25 meters at equator. Range: +/- 2136 km.
16	06.0E.2B.34.01.01.01.01 0E.01.02.03.47.00.00.00	Bounding Box Bottom Right Longitude Offset	Float	Degrees	Int24	V3	Longitude offset for bottom right corner of target bounding box. Use with *0601* Frame Center Longitude. Map -(2^{23}-1)..(2^{23}-1) to +/- 19.2 degrees. Use -(2^{23}) as an "error" indicator. Resolution: ~2.3micro deg, ~0.25 meters at equator. Range: +/- 2136 km.
17	06.0E.2B.34.02.05.01.01 0E.01.03.03.14.00.00.00	Target Location	Structure	None	Location	V	Earth-Centered Earth-Fixed (ECEF) location of the target with associated sigma and rho values.
18	06.0E.2B.34.02.04.01.01 0E.01.03.03.17.00.00.00	Target Boundary	Structure	None	Boundary	V	Boundary around the target, typically defined as a bounding box in two or more ECEF locations.
19	06.0E.2B.34.01.01.01.01 0E.01.02.03.58.00.00.00	Target Centroid Pixel Row	Unsigned Integer	Pixels	Uint32	V4	Specifies the row of the target centroid within the video frame in pixels. Range 1 to 2^{32}-1. Numbering commences from 1, denoting the top row.
20	06.0E.2B.34.01.01.01.01 0E.01.02.03.59.00.00.00	Target Centroid Pixel Column	Unsigned Integer	Pixels	Uint32	V4	Specifies the column of the target centroid within the video frame in pixels. Range 1 to 2^{32}-1. Numbering commences from 1, denoting the left column.

VTarget Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
21	06.0E.2B.34.02.03.01.01 0E.01.03.03.1D.00.00.00	WAMI FPA Index	Structure	None	WAMI FPA Index	V	Specifies the row and the column of a Wide Area Motion Imagery (WAMI) sensor Focal Plane Array (FPA) in a two-dimensional array of FPAs. Specifies the FPA in which detection of the target has occurred. An FPA is considered to be the equivalent of a “frame” for VMTI data elements that use the frame as a reference.
101	06.0E.2B.34.02.03.01.01 0E.01.03.03.08.00.00.00	VMask LDS	Structure	None	N/A	V	Local set tag to include a mask for delineating the perimeter of the target. It may be used to extract the target and populate the VChip LDS. Use the VMask Local Set Tags. The length field is the size of all VMask items to be packaged within this tag.
102	06.0E.2B.34.02.03.01.01 0E.01.03.03.09.00.00.00	VObject	Structure	None	N/A	V	Local set tag to specify the class or type of a target. Uses an ontology to describe the set of allowed class or type values.
103	06.0E.2B.34.02.03.01.01 0E.01.03.03.0A.00.00.00	VFeature LDS	Structure	None	N/A	V	Local set tag to include features about the target. More than one VFeature LDS can be included within the VTarget Pack. Use the VFeature Local Set Tags. The length field is the size of all VFeature items to be packaged within this tag.
104	06.0E.2B.34.02.03.01.01 0E.01.03.03.0B.00.00.00	VTracker LDS	Structure	None	N/A	V	Local set tag to include track information about the target. Use the VTracker Local Set Tags. The length field is the size of all VTracker items to be packaged within this tag.
105	06.0E.2B.34.02.03.01.01 0E.01.03.03.13.00.00.00	VChip LDS	Structure	None	N/A	V	Local set tag to include underlying pixel values for the target. The VChip LDS contains a target chip extracted from the video. Use the VChip Local Set Tags. The length field is the size of all VChip items to be packaged within this tag.

7.3 VMask Local Data Set

Table 3: VMask LDS

VMask LDS							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.03.01.01 0E.01.03.03.08.00.00.00	VMask LDS	N/A	None	N/A	Variable	This is the Universal Key for the VMask Local Data Set.
01	06.0E.2B.34.02.04.01.01 0E.01.03.03.18.00.00.00	Polygon	Array of unsigned integers	NA	Series of Unsigned Integers	V	This is a Series of at least three pixel numbers that specify the vertices of a polygon representing the outline of the target. Numbering commences from 1, denoting the Top Left pixel. Values are encoded using the Length-Value construct of a Variable-Length Pack.
02	06.0E.2B.34.02.04.01.01 0E.01.03.03.19.00.00.00	Bit Mask	Array of unsigned integers	NA	Series of Unsigned Integers	V	This element describes the area of the frame occupied by a target, using a run-length encoded bit mask, with 1 to indicate that a pixel subtends a part of the target and 0 to indicate otherwise. A Series of pixel-number-plus-run-length pairs, each describing the starting pixel number and the number of pixels in a run. Numbering commences from 1, denoting the Top Left pixel. Pixel numbers are encoded using the Length-Value construct of a Variable-Length Pack. The length of each run is encoded using BER Length encoding.

7.4 VChip Local Data Set

Table 4: VChip LDS

VChip LDS							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.03.01.01 0E.01.03.03.13.00.00.00	VChip LDS	N/A	None	N/A	Variable	This is the Universal Key for the VChip Local Data Set.
01	06.0E.2B.34.01.01.01.01 0E.01.01.03.30.00.00.00	Image Type	String	NA	ISO-7	V	Multi-Purpose Internet Extension (MIME) type specifying the VChip image type (limited to “jpeg”, and “png”).
02	06.0E.2B.34.01.01.01.01 0E.01.01.03.31.00.00.00	Image URI	String	NA	ISO-7	V	Uniform Resource Identifier (or Uniform Resource Locator) that refers to an image stored on a server.
03	06.0E.2B.34.01.01.01.01 0E.01.01.03.32.00.00.00	Embedded Image			Wrapper		An image “chip” of the image type specified by Tag 01.

7.5 VObject Local Data Set

Table 5: VObject LDS

VFeature LDS							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.03.01.01 0E.01.03.03.09.00.00.00	VObject LDS	N/A	None	N/A	Variable	This is the Universal Key for the VObject Local Data Set.
01	06.0E.2B.34.01.01.01.01 0E.01.01.03.31.00.00.00	Ontology	URI	NA	ISO-7	V	This element is a Uniform Resource Identifier (URI) which refers to a VObject ontology. The ontology shall be expressed using the Web Ontology Language OWL.
02	06.0E.2B.34.01.01.01.01 0E.01.01.03.33.00.00.00	Class	String	NA	ISO-7	V	The name of the target class or type, as defined in the VObject Ontology

7.6 VFeature Local Data Set

Table 6: VFeature LDS

VFeature LDS							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.03.01.01 0E.01.03.03.0A.00.00.00	VFeature LDS	N/A	None	N/A	Variable	This is the Universal Key for the VFeature Local Data Set.
01	06.0E.2B.34.01.01.01.01 0E.01.01.03.31.00.00.00	Schema	URI	NA	ISO-7	V	Uniform Resource Identifier (URI) which refers to an OGC Geography Markup Language (GML) Observations and Measurements (O&M) application schema.
02	06.0E.2B.34.01.01.01.01 0E.01.01.03.34.00.00.00	Feature		NA	Wrapper	V	OGC GML document structured according to the schema specified by Tag 01. Intended to capture properties (values) observed for a feature of interest.

7.7 VTracker Local Data Set

Table 7: VTracker LDS

VTracker LDS							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.03.01.01 0E.01.03.03.0B.00.00.00	VTracker LDS	N/A	None	N/A	Variable	This is the Universal Key for the VTracker Local Data Set.
01	06.0E.2B.34.01.01.01.01 0E.01.01.03.35.00.00.00	Target ID	Unsigned Integer	NA	Uint64	V8	A unique identifier for all VMTI detections known or believed to be the same entity, insofar as that is possible.
02	06.0E.2B.34.01.01.01.01 0E.01.01.03.36.00.00.00	Detection Status	Unsigned Integer	NA	Uint8	F1	An enumeration indicating the current state of VMTI detections for a given entity (Inactive, Active, Dropped, Stopped).
03	06.0E.2B.34.01.01.01.03 07.02.01.01.01.05.00.00	Start Time Stamp	Unsigned Long	Micro-seconds	Uint64	V8	Date and time for the first observation of the entity.
04	06.0E.2B.34.01.01.01.03 07.02.01.01.01.05.00.00	End Time Stamp	Unsigned Long	Micro-seconds	Uint64	V8	Date and time of the most recent observation of the entity.

VTracker LDS							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
05	06.0E.2B.34.02.04.01.01 0E.01.03.03.17.00.00.00	Bounding Box	Structure	NA	Boundary	V	Set of Boundary vertices that specify a minimum bounding area or volume, which encloses the full extent of VMTI detections for the entity.
06	06.0E.2B.34.01.01.01.01 0E.01.02.03.49.00.00.00	Algorithm	String	NA	ISO-7	V	Name or description of the algorithm or method used to create or maintain object movement reports or intervening predictions of such movement. The intent of this element is to identify uniquely the VMTI algorithm or method used.
07	06.0E.2B.34.01.01.01.01 0E.01.01.03.37.00.00.00	Confidence	Unsigned Integer	NA	Uint8	F1	An estimation of the certainty or correctness of VMTI movement detections. Larger values indicate greater confidence. Zero indicates no confidence.
08	06.0E.2B.34.01.01.01.01 0E.01.01.03.38.00.00.00	Number of Points	Unsigned Integer	NA	Uint16	V2	Number of coordinates of type Location that describe the locus of VMTI detections.
09	06.0E.2B.34.02.04.01.01 0E.01.03.03.1A.00.00.00	Locus	Structure	NA	Series	V	Points of type Location that represent the locations of VMTI detections.
10	06.0E.2B.34.02.05.01.01 0E.01.03.03.15.00.00.00	Velocity	Structure	NA	Velocity	V	Velocity of the entity at the time of last observation.
11	06.0E.2B.34.02.05.01.01 0E.01.03.03.16.00.00.00	Acceleration	Structure	NA	Acceleration	V	Acceleration of the entity at the time of last observation.

7.8 WAMI FPA Index Pack

Table 8: WAMI FPA Index Pack

WAMI FPA Index Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.03.01.01 0E.01.03.03.1D.00.00.00	WAMI FPA Index	N/A	None	N/A	Variable	This is the Universal Key for the WAMI FPA Index Pack.
01	06.0E.2B.34.01.01.01.01 0E.01.02.03.5A.00.00.00	FPA Row	Unsigned Integer	NA	Uint8	F1	Specifies the row of a Wide Area Motion Imagery (WAMI) sensor Focal Plane Array (FPA) in a two-dimensional array of FPAs, numbering from 1, top to bottom, starting at the upper left of the array. Specifies the FPA in which detection of the target has occurred. An FPA is considered to be the equivalent of a “frame” for VMTI data elements that use the frame as a reference.
02	06.0E.2B.34.01.01.01.01 0E.01.02.03.5B.00.00.00	FPA Column	Unsigned Integer	NA	Uint8	F1	Specifies the column of a Wide Area Motion Imagery (WAMI) sensor Focal Plane Array (FPA) in a two-dimensional array of FPAs, numbering from 1, left to right, starting at the upper left of the array. Specifies the FPA in which detection of the target has occurred. An FPA is considered to be the equivalent of a “frame” for VMTI data elements that use the frame as a reference.

7.9 KLV Complex Types

Details of the KLV Complex Types used in 0903 appear in Appendix B.

7.9.1 *Location*

Table 9: Location Truncation Pack

Location Truncation Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.05.01.01 0E.01.03.03.14.00.00.00	Location Truncation Pack	N/A	None	N/A	Variable	This is the Universal Key for the Location Truncation Pack type
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.28.00.00.00	Latitude	Float	Degrees	Uint32	F4	Latitude of point in degrees with respect to the WGS84 datum. Map 0..(2^{32}-1) to -90.0..+90.0 degrees. Resolution: ~0.042 micro-degrees. Range: -90.0..+90.0 degrees.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.29.00.00.00	Longitude	Float	Degrees	Uint32	F4	Longitude of point in degrees with respect to the WGS84 datum. Map 0..(2^{32}-1) to -180.0..+180.0 degrees. Resolution: ~0.083 micro-degrees. Range: -180.0..+180.0 degrees.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2A.00.00.00	Height	Float	Meters	Uint16	F2	Height of point in meters above the WGS84 Ellipsoid (HAE). Map 0..(2^{16}-1) to -900..19000 meters. Resolution: ~0.3 meters. Range: -900..19000 meters.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2D.00.00.00	Sigma_ Latitude	Float	Meters	Uint16	F2	Standard deviation of Latitude. Map 0..(2^{16}-1) to 0.0..650.0 meters. Resolution: ~0.01 meters. Range: 0.0..650.0 meters.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2D.01.00.00	Sigma_ Longitude	Float	Meters	Uint16	F2	Standard deviation of Longitude. Map 0..(2^{16}-1) to 0.0..650.0 meters. Resolution: ~0.01 meters. Range: 0.0..650.0 meters.

Location Truncation Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2D.02.00.00	Sigma_ Height	Float	Meters	Uint16	F2	Standard deviation of Height. Map 0..(2^{16}-1) to 0.0..650.0 meters. Resolution: ~0.01 meters. Range: 0.0..650.0 meters.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2E.00.00.00	Rho_Lat_Lon	Float	N/A	Uint16	F2	Correlation coefficient between Latitude and Longitude. Map 0..(2^{16}-1) to -1.0..+1.0. Resolution: ~3 x 10^{-5}. Range: -1.0..+1.0.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2E.01.00.00	Rho_Lat_Ht	Float	N/A	Uint16	F2	Correlation coefficient between Latitude and Height. Map 0..(2^{16}-1) to -1.0..+1.0. Resolution: ~3 x 10^{-5}. Range: -1.0..+1.0.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2E.02.00.00	Rho_Lon_Ht	Float	N/A	Uint16	F2	Correlation coefficient between Longitude and Height. Map 0..(2^{16}-1) to -1.0..+1.0. Resolution: ~3 x 10^{-5}. Range: -1.0..+1.0.

7.9.2 *Velocity*

Table 10: Velocity Truncation Pack

Velocity Truncation Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.05.01.01 0E.01.03.03.15.00.00.00	Velocity Truncation Pack	N/A	None	N/A	Variable	This is the Universal Key for the Velocity Truncation Pack type
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2B.00.00.00	X_Component	Float	Meters/Second	Uint16	F2	Velocity East to West Map 0..(2^{16}-1) to -900.0..900.0 meters/second. Resolution: 0.0275. Range: -900.0-900.0 meters/second.

Velocity Truncation Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2B.01.00.00	Y_Component	Float	Meters/Second	Uint16	F2	Velocity South to North Map 0..(2^{16}-1) to -900.0..900.0 meters/second. Resolution: 0.0275. Range: -900.0-900.0 meters/second.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2B.02.00.00	Z_Component	Float	Meters/Second	Uint16	F2	Velocity upward Map 0..(2^{16}-1) to -900.0..900.0 meters/second. Resolution: 0.0275. Range: -900.0-900.0 meters/second.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2D.03.00.00	Sigma_X	Float	Meters/Second	Uint16	F2	Standard deviation along X axis. Map 0..(2^{16}-1) to 0.0..650.0 meters. Resolution: ~0.01 meters/second. Range: 0.0..650.0 meters/second.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2D.04.00.00	Sigma_Y	Float	Meters/Second	Uint16	F2	Standard deviation along Y axis. Map 0..(2^{16}-1) to 0.0..650.0 meters. Resolution: ~0.01 meters/second. Range: 0.0..650.0 meters/second.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2D.05.00.00	Sigma_Z	Float	Meters/Second	Uint16	F2	Standard deviation along Z axis. Map 0..(2^{16}-1) to 0.0..650.0 meters. Resolution: ~0.01 meters/second. Range: 0.0..650.0 meters/second.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2E.03.00.00	Rho_X_Y	Float	N/A	Uint16	F2	Correlation coefficient between X and Y. Map 0..(2^{16}-1) to -1.0..+1.0. Resolution: ~3 x 10^{-5}. Range: -1.0..+1.0.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2E.04.00.00	Rho_X_Z	Float	N/A	Uint16	F2	Correlation coefficient between X and Z. Map 0..(2^{16}-1) to -1.0..+1.0. Resolution: ~3 x 10^{-5}. Range: -1.0..+1.0.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2E.05.00.00	Rho_Y_Z	Float	N/A	Uint16	F2	Correlation coefficient between Y and Z Map 0..(2^{16}-1) to -1.0..+1.0. Resolution: ~3 x 10^{-5}. Range: -1.0..+1.0.

7.9.3 ***Acceleration***

Table 11: Acceleration Truncation Pack

Acceleration Truncation Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.05.01.01 0E.01.03.03.16.00.00.00	Acceleration Truncation Pack	N/A	None	N/A	Variable	This is the Universal Key for the Acceleration Truncation Pack type
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2C.00.00.00	X_Component	Float	Meters/Second^2	Uint16	F2	Acceleration West to East Map 0..(2^{16}-1) to -900.0..900.0 meters/second^2. Resolution: 0.0275. Range: -900.0-900.0 meters/second.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2C.01.00.00	Y_Component	Float	Meters/Second^2	Uint16	F2	Acceleration South to North Map 0..(2^{16}-1) to -900.0..900.0 meters/second^2. Resolution: 0.10275. Range: -900.0-900.0 meters/second.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2C.02.00.00	Z_Component	Float	Meters/Second^2	Uint16	F2	Acceleration upward Map 0..(2^{16}-1) to -900.0..900.0 meters/second^2. Resolution: 0.0275. Range: -900.0-900.0 meters/second.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2D.06.00.00	Sigma_X	Float	Meters/Second^2	Uint16	F2	Standard deviation along X axis. Map 0..(2^{16}-1) to 0.0..650.0 meters/second^2. Resolution: ~0.01 meters/second^2. Range: 0.0..650.0 meters/second^2.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2D.07.00.00	Sigma_Y	Float	Meters/Second^2	Uint16	F2	Standard deviation along Y axis. Map 0..(2^{16}-1) to 0.0..650.0 meters/second^2. Resolution: ~0.01 meters/second^2. Range: 0.0..650.0 meters/second^2.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2D.08.00.00	Sigma_Z	Float	Meters/Second^2	Uint16	F2	Standard deviation along Z axis. Map 0..(2^{16}-1) to 0.0..650.0 meters/second^2. Resolution: ~0.01 meters/second^2. Range: 0.0..650.0 meters/second^2.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2E.06.00.00	Rho_X_Y	Float	N/A	Uint16	F2	Correlation coefficient between X and Y. Map 0..(2^{16}-1) to -1.0..+1.0. Resolution: ~3 x 10^{-5}. Range: -1.0..+1.0.

Acceleration Truncation Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2E.07.00.00	Rho_X_Z	Float	N/A	Uint16	F2	Correlation coefficient between X and Z. Map 0..(2^{16}-1) to -1.0..+1.0. Resolution: ~3 x 10^{-5}. Range: -1.0..+1.0.
N/A	06.0E.2B.34.01.01.01.01 0E.01.01.03.2E.08.00.00	Rho_Y_Z	Float	N/A	Uint16	F2	Correlation coefficient between Y and Z Map 0..(2^{16}-1) to -1.0..+1.0. Resolution: ~3 x 10^{-5}. Range: -1.0..+1.0.

7.9.4 *Boundary*

Table 12: Boundary Series

Boundary Series							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.04.01.01 0E.01.03.03.17.00.00.00	Boundary Series	N/A	None	N/A	Variable	This is the Universal Key for the Bounding Box Truncation Pack type
N/A	06.0E.2B.34.02.05.01.01 0E.01.03.03.14.00.00.00	Location Pack	An Array of Location Elements	NA	A Series of Location Elements	V	A Series of Location data elements, one for each vertex of a bounding area or volume. Generally used for (planar) bounding boxes. However, it can support the specification of multifaceted (triangulated) volumes, as well, and it is used in this manner to describe the bounding volume of a track in VTracker.

7.10 VMTI Core Data

7.10.1 *Tag 1: Checksum*

Data Element Definition			
Element Name	Checksum	Type	Units
		Unsigned Integer	None
Valid Values	All integer values from 0 to 0xFFFF (65,535) Recomputed for each VMTI LDS instance		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.04.01.01.01 / 0E.01.02.03.01.00.00.00		LDS Tag
			1
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	2 Bytes	Binary
Notes			
- Checksum used to detect errors within a Local Data Set packet. - Lower 16-bits of summation. - Performed on entire LDS packet, including 16-byte UDS key and 1-byte checksum length.			
Conversion			
NA			
Example Value		Example Encoded LDS Value	
0x00 00		[K] [L] [V] = [0x01] [0x02] [0x00 00]	

It is permitted to include a Checksum in a VMTI LDS packet. If the VMTI LDS is not embedded with a *0601* LDS (which is allowed in exceptional cases only), the Checksum is mandatory. If it appears, it must come last in the packet. However, for bandwidth efficiency, it is recommended that the VMTI LDS Checksum be removed in deference to the companion *0601* checksum.

A checksum is not required (or desired) for each of the subordinate sets within the VMTI LDS. Bandwidth efficiency is the prime reason for this. It is allowable and conceivable that hundreds of targets will be detected per frame. Frame rates of 60 fps and higher will be commonplace. If included in the subordinate sets (VTarget Pack) a two-byte checksum will add approximately 100 kbps to the bandwidth overhead for every 100 targets in the frame. However, if no checksum is included with each VTarget Pack, hundreds of targets will be declared invalid, if even one of them has a transmission error, because the only other checksum is the VMTI LDS checksum. Nevertheless, those same targets will probably be transmitted with the next set of detections, nominally, one 60th of a second later. If the data link is so poor as to have errors in every frame, then this is an issue related to the data link that should be dealt with in context, rather than condemn all users to unnecessary overheads.

Refer to *0601* for example 16-bit checksum code.

7.10.2 ***Tag 2: UNIX Time Stamp***

Data Element Definition			
Element Name	UNIX Time Stamp	Type	Units
		Unsigned Integer	Microseconds
Valid Values	All integer values from 0 to (2 ^ 64) – 1 Unique for each VMTI LDS instance		Precision
			1 microsecond
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.03 / 07.02.01.01.01.05.00.00		LDS Tag
			2
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	8 Bytes	Binary
Notes			
- Microseconds elapsed since midnight Coordinated Universal Time (UTC) of January 1, 1970. - Derived from the POSIX IEEE 1003.1 standard.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
April 19 2001, 04:25:21.000000 GMT	[K] [L] [V] = [0x02] [0x08] [0x00 03 82 44 30 F6 CE 40]		

UNIX time, or POSIX time, is a system used to describe points in time. This system is widely used within systems of differing underlying architectures. UNIX time is an encoding of Coordinated Universal Time (UTC).

It is permitted to include the Time Stamp in a VMTI LDS packet. If it appears, it shall come as the first KLV element in the packet. For bandwidth efficiency, the VMTI LDS Time Stamp may be omitted in deference to the companion *0601* time stamp. However, if both time stamps are present, the VMTI LDS Time Stamp takes precedence.

Some VMTI systems may not have access to a GPS time signal. In such circumstances designers are recommended to use the VMTI CPU clock or embedded video time code, if available. If no time information is available at all, VMTI LDS packets can be sent without a time stamp. These packets can derive time stamps related to time of arrival at downstream processors / systems and use Tag 7: Video Frame Number (if present) as a frame sequence check.

7.10.3 *Tag 3: VMTI System Name or Description*

Data Element Definition			
Element Name	VMTI System Name / Description	Type	Units
		String	NA
Valid Values	A string of any 7-bit ISO Characters up to 32 characters long		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.02.02.7C.00.00.00		LDS Tag
			3
Volatility	Encoded Type	Length	Format
Static	String	Variable up to 32 Bytes	7-bit ISO
Notes			
- Name or description of the VMTI system. - Value field is free text. - Null termination is not required			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
DSTO_ADSS_VMTI	[K] [L] [V] = [0x03] [0x0E] [0x44 53 54 4F 5F 41 44 53 53 5F 56 4D 54 49]		

The VMTI System Name or Description field is used to identify the system that produced the VMTI targets. It should be as descriptive as possible.

7.10.4 *Tag 4: VMTI LDS Version Number*

Data Element Definition			
Element Name	VMTI LDS Version Number	Type	Units
		Unsigned Integer	NA
Valid Values	All integer values from 0 to 0xFFFF (65535)		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.02.05.04.00.00.00		LDS Tag
			4
Volatility	Encoded Type	Length	Format
Static	Unsigned Integer	Variable up to 2 Bytes	Binary
Notes			
- Version number of the VMTI LDS document used to generate the VMTI metadata. - Values of 1 through 65535 correspond to document revisions 1 through 65535. - Version 2 (0903.2) would be encoded as shown in the example below.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
2	[K] [L] [V] = [0x04] [0x01] [0x02]		

The VMTI LDS Version Number serves to notify downstream clients of the version of the VMTI LDS used to encode the VMTI metadata.

7.10.5 *Tag 5: Total Number of Targets Detected in the Frame*

Data Element Definition			
Element Name	Total Number of Targets Detected in the Frame	Type	Units
		Unsigned Integer	NA
Valid Values	All integer values from 0 to 0xFFFFFF (16,777,215)		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.02.03.36.00.00.00		LDS Tag
			5
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	Variable up to 3 Bytes	Binary
Notes			
- Total number of moving targets detected in the frame. - A value of 0 represents no targets detected			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
28 (0x1C)	[K] [L] [V] = [0x05] [0x01] [0x1C]		

This element represents the total number of moving objects detected by the VMTI process in the current frame on which the VMTI process has been run. This is particularly relevant when the number of targets reported (Tag 6) is less than the total number detected in the frame. This circumstance may arise when there are a large number of detected targets and a culling process has reduced the number of targets included in the VMTI LDS packet.

7.10.6 ***Tag 6: Number of Reported Targets***

Data Element Definition			
Element Name	Total Number of Reported Targets	Type	Units
		Unsigned Integer	NA
Valid Values	All integer values from 0 to 0xFFFFFF (16,777,215)		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.02.03.37.00.00.00		LDS Tag
			6
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	Variable up to 3 Bytes	Binary
Notes			
- Number of moving targets reported from the frame. - Number of Reported Targets = Total Number of Targets (Tag 5) – Number of Culled Targets - The culling process is usually linked to priority value or confidence level.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
14 (0x0E)	[K] [L] [V] = [0x06] [0x01] [0x0E]		

The Number of Reported Targets in the frame is required when one or more detected targets are not included in the VMTI LDS. It may be necessary (for bandwidth efficiency) to report only a subset of the detected targets. A culling process may be performed at the destination (for example, by display software) or at the source by the VMTI system itself. In all cases, it is expected that the culling will be undertaken according to priority values (VTarget Pack Tag 4) and/or confidence levels (VTarget Pack Tag 5).

7.10.7 *Tag 7: Video Frame Number*

Data Element Definition			
Element Name	Video Frame Number	Type	Units
		Unsigned Integer	NA
Valid Values	All integer values from 0 to 0xFFFFFF (16,777,215)		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.1F.00.00.00		LDS Tag
			7
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	Variable up to 3 Bytes	Binary
Notes			
- The video frame number corresponding to the frame in which the targets were detected. - Use of the time stamp (Tag 2) is preferred but frame number can be used where a time stamp is not available. - Range 0 to 2^{24} -1 which equates to approximately 155 hours at 30fps.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
78,000 (0x0130B0)	[K] [L] [V] = [0x07] [0x03] [0x01 30 B0]		

The Video Frame Number is required only when a time stamp is not available and, even then, is useful only if the video source includes a frame number. It is expected that most VMTI systems will be provided video directly from the sensor rather than from a video encoder. This is for two reasons; firstly, the VMTI process is likely to produce better results on uncompressed video and secondly, latencies can be minimized by running the VMTI and encoding processes in parallel rather than in sequence.

In the absence of a frame number in the raw video source, frame capture devices typically number frames from commencement of the capture and begin with frame 1. If the processes are parallel, the VMTI process will require its own capture process – independent of the encoder capture process – and the frame numbers of each will not be synchronous. Nevertheless, there will be cases where the VMTI process is run on encoded video. Under these circumstances and in the absence of a time stamp, the VMTI system should use the source frame number to populate the Video Frame Number element.

7.10.8 ***Tag 8: Frame Width***

Data Element Definition			
Element Name	Frame Width	Type	Units
		Unsigned Integer	Pixels
Valid Values	All integer values from 1 to 0xFFFFFF (16,777,215)		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.02.07.00.00.00		LDS Tag
			8
Volatility	Encoded Type	Length	Format
Periodic	Unsigned Integer	Variable up to 3 Bytes	Binary
Notes			
- Width of the video frame in pixels. - Pixels are assumed to appear in row-major order. Frame Width corresponds to the number of pixels in a row of the image. - Value of zero is meaningless and should not be used			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
1920 (0x0780)	[K] [L] [V] = [0x08] [0x02] [0x07 80]		

Frame Width has been included for two reasons. First, efficiency can be realized by the use of pixel number rather than X and Y coordinates to locate the target within the video frame. The pixel number calculation relies upon frame width. Second, two cameras of different frame size may reside within the one turret (*e.g.*, High Definition spotter scope with Standard Definition wide angle). The underlying video in the transport stream includes frame size information but when the underlying video is different from the video on which the VMTI process was run (or a metadata only stream is transmitted), then this Tag can be used to include the frame width.

7.10.9 *Tag 9: Frame Height*

Data Element Definition			
Element Name	Frame Height	Type	Units
		Unsigned Integer	Pixels
Valid Values	All integer values from 1 to 0xFFFFFF (16,777,215)		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.02.08.00.00.00		LDS Tag
			9
Volatility	Encoded Type	Length	Format
Periodic	Unsigned Integer	Variable up to 3 Bytes	Binary
Notes			
- Height of the video frame in pixels. - Pixels are assumed to appear in row-major order. Frame Height corresponds to the number of rows of pixels in the image. - Value of zero is meaningless and should not be used			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
1080 (0x0438)	[K] [L] [V] = [0x09] [0x02] [0x04 38]		

See description of Frame Width under Tag 8.

7.10.10 *Tag 10: VMTI Source Sensor*

Data Element Definition			
Element Name	VMTI Source Sensor	Type	Units
		String	NA
Valid Values	A string of any 7-bit ISO Characters up to 127 characters long		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 04.20.01.02.01.01.00.00		LDS Tag
			10
Volatility	Encoded Type	Length	Format
Periodic	String	Variable up to 127 Bytes	7-bit ISO
Notes			
- Free text identifier of the image source sensor *e.g.*, 'EO Nose', 'EO Zoom (DLTV)', 'EO Spotter', 'IR Mitsubishi PtSi Model 500', 'IR InSb Amber Model TBT', 'LYNX SAR Imagery', 'TESAR Imagery', etc. - Can be used to identify the VMTI process if more than one is run simultaneously. - Can be used to identify the imagery source on which the VMTI process was run for systems that have multiple bore-sighted sensors. - Null termination of the string is not required			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
"EO Nose"	[K] [L] [V] = [0x0A][0x07][0x45 4F 20 4E 6F 73 65]		

The VMTI Source Sensor should identity as uniquely as possible the sensor that was the source of the video used by the moving object detection process. This element must be updated at the first opportunity following a change.

7.10.11 *Tag 11: VMTI Sensor Horizontal Field of View*

Data Element Definition			
Element Name	VMTI Sensor Horizontal Field of View	Type	Units
		Float	Degrees
Valid Values	The set of real numbers from 0 to 180 inclusive		Precision
			~2.75 millidegrees
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.02 / 04.20.02.01.01.08.00.00		LDS Tag
			11
Volatility	Encoded Type	Length	Format
Periodic	Unsigned Integer	Variable up to 2 Bytes	Binary

Notes

- Horizontal field of view of imaging sensor input to VMTI process. Required only if VMTI process is run on a different imaging sensor from that described by the parent *0601* packet.
- Can be used with HFOV (Tag 16) from *0601* to scale VMTI X,Y coordinates.

Conversion

Given:
The maximum measured value is 180 degrees
The minimum measured value is 0 degrees

If Mv is 12.5 degrees, then

$$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(12.5 - 0)\,(2^{16} - 1)}{(180 - 0)}\right) = 4551$$

If Ev is 4551, then

$$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(0 + \frac{4551\,(180 - 0)}{(2^{16} - 1)}\right) = 12.49989 \text{ degrees}$$

Example Value	Example Encoded LDS Value
12.5 Degrees	[K] [L] [V] = [0x0B][0x02][0x11 C7]

This value should only be populated if the video input to the VMTI process is different from that streamed with the *0601* data. Under these circumstances the ratio (k_x) between the HFOV value in the VMTI LDS and that in the *0601* stream can be used to scale the VMTI X,Y coordinates for display in the streamed video display. The scaling is given by:

$$x_2 = k_x\left(x_1 - \left(\frac{Frame\_Width}{2}\right)\right) + \frac{Frame\_Width}{2}$$

Where: x_1 is the original X coordinate of the target extracted from the target pixel number.
x_2 is the scaled X coordinate of the target in the video.
k_x is the scaling factor calculated according to the following equation:

$$k_x = \frac{\tan\left(\frac{1}{2}\theta_{H1}\right)}{\tan\left(\frac{1}{2}\theta_{H2}\right)}$$

Where: θ_{H1} is the HFOV of the original (VMTI) sensor.
θ_{H2} is the HFOV of the sensor to which the targets are being scaled.

It is expected that in most cases k_x will be sufficient to perform scaling in both the X and Y directions (Y scaling uses frame height rather than frame width – see Tag 12). The VFOV element in VMTI LDS 12 is provided for those cases where the aspect ratio of the two sensors is different (for example 4:3 and 16:9).

These equations are valid if the frame width of the two sensors is the same. If not, the equation becomes more complex.

When the streamed video is from a narrow field of view sensor and the VMTI process is run on video from a bore-sighted wide field of view sensor, it can be expected that moving targets are outside the frame of the streamed video. Under these circumstances, the video display device could add a blank border around the active video and present the moving targets to the operator, albeit without the underlying video. Alternatively, the video display device could present highlights around the perimeter of the frame to indicate movers outside the frame and their radial direction from the bore-sight.

7.10.12 *Tag 12: VMTI Sensor Vertical Field of View*

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">VMTI Sensor Vertical Field of View</td><td>Type</td><td>Units</td></tr>
<tr><td>Float</td><td>Degrees</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">The set of real numbers from 0 to 180 inclusive</td><td>Precision</td></tr>
<tr><td>~2.75 milli-degrees</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">06.0E.2B.34. 01.01.01.07 / 04.20.02.01.01.0A.01.00</td><td>LDS Tag</td></tr>
<tr><td>12</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Periodic</td><td>Unsigned Integer</td><td>Variable up to 2 Bytes</td><td>Binary</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- Vertical field of view of imaging sensor input to VMTI process. Required only if VMTI process is run on a different imaging sensor from that described by the parent 0601 packet.
- Can be used with VFOV (Tag 17) from 0601 to scale VMTI X,Y coordinates
- Typically required only to account for aspect ratio variation.</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">Given:
The maximum measured value is 180 degrees
The minimum measured value is 0 degrees

If Mv is 10 degrees, then

$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(10 - 0)\,(2^{16} - 1)}{(180 - 0)}\right) = 3641$

If Ev is 3641, then

$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(0 + \frac{3641\,(180 - 0)}{(2^{16} - 1)}\right) = 10.00046\ degrees$</td></tr>
<tr><td colspan="2">Example Value</td><td colspan="2">Example Encoded LDS Value</td></tr>
<tr><td colspan="2">10.0 Degrees</td><td colspan="2">[K] [L] [V] = [0x0C][0x02][0x0E 39]</td></tr>
</table>

This value should only be populated if the video input to the VMTI process is different from that streamed with the *0601* data. Under these circumstances the ratio (k_y) between the VFOV value in the VMTI LDS and that in the *0601* stream can be used to scale the VMTI Y coordinate for display in the streamed video display. The scaling is given by:

$$y_2 = k_y\left(y_1 - \left(\frac{Frame\_Height}{2}\right)\right) + \frac{Frame\_Height}{2}$$

Where: y_1 is the original Y coordinate of the target extracted from the target pixel number.
y_2 is the scaled Y coordinate of the target in the video.

k_y is the scaling factor calculated according to the following equation:

$$k_y = \frac{\tan\left(\frac{1}{2}\theta_{V1}\right)}{\tan\left(\frac{1}{2}\theta_{V2}\right)}$$

Where: θ_{V1} is the VFOV of the original (VMTI) sensor.

θ_{V2} is the VFOV of the sensor to which the targets are being scaled.

It is expected that in most cases k_x (see Tag 11) will be sufficient to perform scaling in both the X and Y directions. The VFOV element in VMTI LDS 12 is provided for those cases where the aspect ratio of the two sensors is different (for example 4:3 and 16:9).

These equations are valid if the frame height of the two sensors is the same. If not, the equation becomes more complex.

When the streamed video is from a narrow field of view sensor and the VMTI process is run on video from a bore-sighted wide field of view sensor, it can be expected that moving targets are outside the frame of the streamed video. Under these circumstances, the video display device could add a blank border around the active video and present the moving targets to the operator albeit without the underlying video. Alternatively, the video display device could present highlights around the perimeter of the frame to indicate movers outside the frame and their radial direction from the bore-sight.

7.10.13 *Tag 101: VTargetSeries*

Data Element Definition			
Element Name	VTargetSeries	Type	Units
		Array	VTarget Pack
Valid Values	All valid VTarget Pack structures		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.04.01.01 / 0E.01.03.03.1B.00.00.00		LDS Tag
			101
Volatility	Encoded Type	Length	Format
Dynamic	Series	Variable	VTarget Pack
Notes			
- Series of target metadata. - Use the Target Local Set Tags. - The length field for VTargetSeries is the sum of the lengths of all the contained target metadata. - The length field for each VTarget Pack is the size of all elements within that pack, including the TargetID. - The mandatory Target ID Number allows discrimination between instances.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
Three VTarget Packs with Target IDs of 1, 2 and 3	[K] [L] [V] = [0x65] [0xL0] [0xL1 0x01 … 0xL2 0x02 … 0xL3 0x03 …]		

The VTargetSeries is a Series (SMPTE 336M Variable-Length Pack) of VTarget Packs. Each VTarget Pack contains the metadata pertaining to one individual target. Multiple instances of VTarget Pack can reside in one VTargetSeries. Each VTarget Pack is preceded by the short or long BER encoding of its length. The mandatory Target ID Number (first element of each VTarget Pack) provides discrimination between VTarget Packs.

7.11 VTarget Pack

Data Element Definition			
Element Name	VTarget Pack	Type	Units
		Structure	NA
Valid Values	This is a data structure composed of a Target ID Number and a sequence of Tag-Length-Value triplets		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.05.01.01 / 0E.01.03.03.07.00.00.00		LDS Tag
			NA
Volatility	Encoded Type	Length	Format
Dynamic	Defined Length Pack	Variable	Binary
Notes			
- A VTarget Pack has a structure that is identical to a Local Data Set with the exception that the mandatory Target ID Number element has neither tag nor length.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
Target ID = 27 (0x1B) Centroid Pixel = 409,600 (0x06 40 00)	[V {T_i L_i V_i}] = [0x1B {0x01 0x03 [0x06 40 00]}]		

The VTarget Pack contains metadata elements that describe one single target.

7.11.1 ***Target ID Number***

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Target ID Number</td><td>Type</td><td>Units</td></tr>
<tr><td>Unsigned Integer</td><td>NA</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">All integer values from 1 to 2,097,151</td><td>Precision</td></tr>
<tr><td>NA</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">NA</td><td>Pack Tag</td></tr>
<tr><td>NA</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>Unsigned Integer</td><td>Variable up to 3 Bytes</td><td>BER-OID</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- This element is mandatory and it must come first in the VTarget Pack.
- It is BER-OID encoded to convey the length but has no Tag field.
- A value of zero is not to be used.
- BER-OID encoding allocates the upper bit of each Byte to indicate “last Byte” (0) or not (1). 3 Bytes provide up to 21 bits for the ID value, for a maximum value of 2,097,151. [With BER-OID encoding, there really is no compelling reason to specify a limit at all.]</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">NA</td></tr>
<tr><td>Example Value</td><td colspan="3">Example Encoded LDS Value</td></tr>
<tr><td>27 (0x1B)</td><td colspan="3">[V] = [0x1B]</td></tr>
</table>

The Target ID Number uniquely identifies a target across multiple frames until the identification number is reset by the New Detection Flag (Tag 6 within the VTarget Pack). Sophisticated VMTI systems may use the same Target ID Number to identify a common target detected by different sensors, and therefore allow correlation between targets detected simultaneously by different sensor systems.

This element is mandatory, as it has no Tag or Length field. Thus, its absence will result in incorrect parsing of the subsequent data. It shall be included as the first element of the VTarget Pack in the form of a BER-OID encoded single value.

7.11.2 *Tag 1: Target Centroid Pixel Number*

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Target Centroid Pixel Number</td><td>Type</td><td>Units</td></tr>
<tr><td>Unsigned Integer</td><td>Pixels</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">All integer values from 1 to 0xFFFFFFFFFFFF (281,474,976,710,655)</td><td>Precision</td></tr>
<tr><td>NA</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">06.0E.2B.34.01.01.01.01 / 0E.01.02.03.38.00.00.00</td><td>Pack Tag</td></tr>
<tr><td>1</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>Unsigned Integer</td><td>Variable up to 6 Bytes</td><td>Binary</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- Defines the position of the centroid of the target within the video frame in pixels.
- Numbering commences with 1, at the top left pixel, and proceeds from left to right, top to bottom.</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">NA</td></tr>
<tr><td>Example Value</td><td colspan="3">Example Encoded LDS Value</td></tr>
<tr><td>409,600 (0x06 40 00)</td><td colspan="3">[K] [L] [V] = [0x01][0x03][0x06 40 00]</td></tr>
</table>

The Target Centroid Pixel Number specifies the position of the target centroid[10] within the frame. The calculation of the pixel number uses the equation: X + ((Y-1) x Frame Width)). The top left pixel of the frame equates to (X,Y) = (1,1) and a pixel number of 1. The Frame Width would normally be provided by the underlying video, but in the absence of video or if the VMTI process is run on a different video input from that in the stream, then VMTI LDS Tag 8 can be used.

The range has been set at 5 bytes to allow frame sizes of up to 1 Terapixel. It is important for bit efficiency to rely on variable length payloads for this value. For example, if the pixel number is < 65536 then only two bytes should be used.

10 The coordinates of the centroid (geometric center) (x_c, y_c) of a non-intersecting closed polygon with n vertices can be calculated using:

$$x_c = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i), \quad y_c = \frac{1}{6A}\sum_{i=0}^{n-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

Where, $$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i)$$

As a practical matter, calculation of the exact centroid is probably not necessary, although it is not precluded. The centroid of a simple bounding box might be adequate.

7.11.3 ***Tag 2: Target Bounding Box Top Left Pixel Number***

Data Element Definition			
Element Name	Target Bounding Box Top Left Pixel Number	Type	Units
		Unsigned Integer	Pixels
Valid Values	All integer values from 1 to 0xFFFFFFFFFFFF (281,474,976,710,655)		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.02.03.39.00.00.00		Pack Tag
			2
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	Variable up to 6 Bytes	Binary
Notes			
- Defines the position of the top left corner of the target bounding box within the video frame in pixels. - Numbering commences with 1, at the top left pixel, and proceeds from left to right, top to bottom			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
409,600 (0x06 40 00)	[K] [L] [V] = [0x02][0x03][0x06 40 00]		

The position of the top left corner of the target bounding box within the frame is specified as a pixel number. The calculation of the pixel number uses the equation: X + ((Y-1) x Frame Width)). The top left pixel of the frame equates to (X,Y) = (1,1) and a pixel number of 1. The Frame Width would normally be provided by the underlying video, but in the absence of video or if the VMTI process is run on a different video input from that in the stream, then VMTI LDS Tag 8 can be used.

The range has been set at 6 bytes to allow frame sizes over 1 Terapixel. It is important for bit efficiency to rely on variable length payloads for this value. For example, if the pixel number is < 65536 then only two bytes should be used.

7.11.4 *Tag 3: Target Bounding Box Bottom Right Pixel Number*

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Target Bounding Box Bottom Right Pixel Number</td><td>Type</td><td>Units</td></tr>
<tr><td>Unsigned Integer</td><td>Pixels</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">All integer values from 1 to 0xFFFFFFFFFFFF (281,474,976,710,655)</td><td>Precision</td></tr>
<tr><td>NA</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">06.0E.2B.34.01.01.01.01 / 0E.01.02.03.3A.00.00.00</td><td>Pack Tag</td></tr>
<tr><td>3</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>Unsigned Integer</td><td>Variable up to 6 Bytes</td><td>Binary</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- Defines the position of the bottom right corner of the target bounding box within the video frame in pixels.
- Numbering commences with 1, at the top left pixel, and proceeds from left to right, top to bottom.
- Maximum size of frame is 1 Terapixel.</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">NA</td></tr>
<tr><td>Example Value</td><td colspan="3">Example Encoded LDS Value</td></tr>
<tr><td>409,600 (0x06 40 00)</td><td colspan="3">[K] [L] [V] = [0x03][0x03][0x06 40 00]</td></tr>
</table>

The position of the bottom right corner of the target bounding box within the frame is specified as a pixel number. The calculation of the pixel number uses the equation: X + ((Y-1) x Frame Width)). The top left pixel of the frame equates to (X,Y) = (1,1) and a pixel number of 1. The Frame Width would normally be provided by the underlying video, but in the absence of video or if the VMTI process is run on a different video input from that in the stream, then VMTI LDS Tag 8 can be used.

The range has been set at 5 bytes to allow frame sizes of up to 1 Terapixel. It is important for bit efficiency to rely on variable length payloads for this value. For example, if the pixel number is < 65536 then only two bytes should be used.

7.11.5 ***Tag 4: Target Priority***

Data Element Definition			
Element Name	Target Priority	Type	Units
		Unsigned Integer	NA
Valid Values	All integer values from 1 to 0xFF (255)		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.02.03.3B.00.00.00		Pack Tag
			4
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	1 Byte	Binary
Notes			
- Priority or validity of target based on criteria within the VMTI system. The target(s) with the highest priority may not have the highest confidence level. - Potential for use in limited bandwidth scenarios to only send highest priority targets. - Multiple targets may have the same priority. - Range is 1 to 255, where 1 is the highest priority.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
27 (0x1B)	[K] [L] [V] = [0x04][0x01][0x1B]		

Target Priority is designed to provide systems that are downstream from VMTI processors a means for intelligently culling targets for a given frame. For example, VMTI processors may generate thousands of hits. These may be used meaningfully in trackers (where bandwidth may not be an issue). However, other clients may have more restrictive bandwidth limitations or not wish to overload systems with thousands of hits. For example, from a human perspective, it may be undesirable to clutter a situational awareness display with thousands of VMTI targets.

7.11.6 *Tag 5: Target Confidence Level*

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Target Confidence Level</td><td>Type</td><td>Units</td></tr>
<tr><td>Unsigned Integer</td><td>NA</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">The set of integer values from 0 to 100 inclusive</td><td>Precision</td></tr>
<tr><td>NA</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">06.0E.2B.34.01.01.01.01 / 0E.01.02.03.3C.00.00.00</td><td>Pack Tag</td></tr>
<tr><td>5</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>Unsigned Integer</td><td>1 Byte</td><td>Binary</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- Confidence level, expressed as a percentage, based on criteria within the VMTI system. Target(s) with the highest confidence may not have the highest priority value.
- Potential for use in limited bandwidth scenarios to only send highest confidence targets.
- Multiple targets may have the same confidence level.
- Range 0 to 100, where 100 percent is the highest confidence.
- Although a confidence level of 0 percent is permitted, it indicates no confidence that a detection is a potential target.</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">NA</td></tr>
<tr><td>Example Value</td><td colspan="3">Example Encoded LDS Value</td></tr>
<tr><td>80 (0x50)</td><td colspan="3">[K] [L] [V] = [0x05][0x01][0x50]</td></tr>
</table>

Target Priority and Target Confidence Level are designed to provide systems that are downstream from VMTI processors a means for intelligently assessing and culling targets for a given frame. A target may be detected with a high confidence but be a low priority target.

7.11.7 *Tag 6: New Detection Flag / Target History*

Data Element Definition			
Element Name	New Detection Flag / Target History	Type	Units
		Unsigned Integer	NA
Valid Values	All integer values from 0 to 0xFFFF (65535)		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.02.03.3D.00.00.00		Pack Tag
			6
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	Variable up to 2 Bytes	Binary
Notes			
- The number of previous times the same target has been detected. - Range 0 to 65535 frames, where a value of 0 denotes the target as a new detection. - Detections are not required to be in consecutive frames.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
2765 (0x0A CD)	[K] [L] [V] = [0x06][0x02][0x0A CD]		

This element is designed primarily to indicate detection of a new target or reuse of a previous VTarget Pack Target ID Number, but it also provides the ability to indicate target persistence and may provide useful context when a target reappears after not being detected for a significant time.

Refer to the description of the VTarget Pack Target ID Number for caveats.

7.11.8 *Tag 7: Percentage of Target Pixels*

Data Element Definition			
Element Name	Percentage of Target Pixels	Type	Units
		Unsigned Integer	Percent
Valid Values	The set of integer numbers from 1 to 100 inclusive		Precision
			1 %
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.02.03.3E.00.00.00		Pack Tag
			7
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	1 Byte	Binary
Notes			
- The percentage of pixels within the bounding box that are detected to be target pixels rather than background pixels. - Range 1 to 100, where 100 signifies that the target completely fills the bounding box. - A value of 0 is invalid. (If a detection has occurred, the bounding box should be sized such that a non-zero percentage of pixels overlaps the target.) - Values above 100 are invalid.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
50% (0x32)	[K] [L] [V] = [0x07][0x01][0x32]		

This element can be used to provide information about how many of the pixels within the bounding box are target pixels. The use of the VMask, VChip, VObject, or VFeature Local Data Sets is recommended for cases where more detail about the target is required.

7.11.9 ***Tag 8: Target Color***

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Target Color</td><td>Type</td><td>Units</td></tr>
<tr><td>Unsigned Integer</td><td>NA</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">All integer values from 0 to 0xFFFFFF (16,777,215) divided into three 8-bit fields</td><td>Precision</td></tr>
<tr><td>NA</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">06.0E.2B.34.01.01.01.01 / 0E.01.02.03.3F.00.00.00</td><td>Pack Tag</td></tr>
<tr><td>8</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>Unsigned Integer</td><td>3 Bytes</td><td>Binary</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- Dominant color of the target, expressed using RGB color values.
- For use when metadata is transmitted in the absence of the underlying video.
- VFeature LDS can be used for more comprehensive color information.
- Represents the RGB color value with:
    - bits 1 – 8 (first byte) = Red
    - bits 9 – 16 (second byte) = Green
    - bits 17 – 24 (third byte) = Blue</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">Let:
Red = 0x55
Green = 0x88
Blue = 0x33

Then:
Target Color = [0x55 00 00] | [0x88 00] | [0x33] = [0x55 88 33]</td></tr>
<tr><td>Example Value</td><td colspan="3">Example Encoded LDS Value</td></tr>
<tr><td>[0x55 88 33]</td><td colspan="3">[K] [L] [V] = [0x08][0x03][0x55 88 33]</td></tr>
</table>

This element may be used for general mapping of any multispectral dataset to an RGB value; however, it is designed primarily to provide color information about a target in the absence of the video. For example, the target is a red car. Mapping sensor systems from frequencies outside of the visible spectrum to an RGB color value is supported; however this may be a source of confusion. Even assigning a monochrome (8-bit) infrared value to RGB as a grayscale value may be misleading to an end user who will interpret the signal as a color rather than an indication of relative intensity within a scene. Monochrome intensity values can be sent (up to 24 bits) using Tag 9: Target Intensity.

7.11.10 *Tag 9: Target Intensity*

Data Element Definition			
Element Name	Target Intensity	Type	Units
		Unsigned Integer	NA
Valid Values	All integer values from 0 to 0xFFFFFF (16,777,215)		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.02.03.40.00.00.00		Pack Tag
			9
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	Variable up to 3 Bytes	Binary
Notes			
- Dominant intensity of the target. - For use when metadata is transmitted in the absence of the underlying video. - VFeature LDS can be used for more comprehensive spectral information. - Value 0 is black.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
13140 [0x33 54]	[K] [L] [V] = [0x09][0x02][0x33 54]		

This element can represent the indicative or dominant intensity of the target at a dynamic range of up to 24 bits. Any sensor with a broad dynamic range may use this tag; however, it is primarily designed for use with Infrared systems that may detect targets at higher than 8-bits per pixel dynamic range and transmit video at lower dynamic ranges (or not include video in the transport stream). Other details providing context or relating to accuracy and precision would be contained in the VFeature LDS.

7.11.11 *Tag 10: Target Location Latitude Offset*

Data Element Definition			
Element Name	Target Location Latitude Offset	Type	Units
		Float	Degrees
Valid Values	The set of real numbers from -19.2 to 19.2 inclusive		Precision
			~2.3 micro-degrees
KLV Encoding			
Universal Label	06.0E.2B.34. 01.01.01.07 / 0E.01.02.03.41.00.00.00		Pack Tag
			10
Volatility	Encoded Type	Length	Format
Dynamic	Signed Integer	Variable up to 3 Bytes	Binary

Notes

- Latitude offset for target from Frame Center Latitude (*0601*), based on WGS84 ellipsoid.
- Use with *0601* Frame Center Latitude.
- Map $-(2^{23}-1)..(2^{23}-1)$ to -/+19.2 degrees.
- Use $-(2^{23})$ as an "error" indicator.
- Resolution: ~2.3 micro-deg, ~0.25 meters at equator.
- Range: +/- 2136 km.

Conversion

Given:
The maximum measured value is +19.2 degrees
The minimum measured value is -19.2 degrees

If Mv is 10.00 degrees, then

$$Ev = (int)round\left(Emin + \frac{(Mv - Mmin)(Emax - Emin)}{(Mmax - Mmin)}\right)$$
$$= (int)round\left(-2^{23} + \frac{(10 - (-19.2))((2^{23} - 1) - (-2^{23}))}{(19.2 - (-19.2))}\right) = 4369066$$

If Ev is 4369066, then

$$Mv = \left(Mmin + \frac{(Ev - Emin)(Mmax - Mmin)}{(Emax - Emin)}\right) = \left(0 + \frac{(4369066 - (-2^{23}))(19.2 - (-19.2))}{((2^{23} - 1) - (-2^{23}))}\right)$$
$$= 10.00000021 \text{ degrees}$$

Example Value	Example Encoded LDS Value
10.00 Degrees	[K] [L] [V] = [0x0A][0x03][0x42 AA AA]

The target location has a real earth coordinate represented by a latitude-longitude pair. Target locations that lie above the horizon do not correspond to a point on the earth. Also, target locations may lie outside of the mapped range. Both cases should either not be reported, or be reported as an “error”.

The Target Latitude Offset is added to the Frame Center Latitude metadata item from the parent *0601* packet to determine the Latitude of the target. Both KLV data items must be converted to decimal prior to addition to determine the actual measured or calculated motion imagery target location.

7.11.12 *Tag 11: Target Location Longitude Offset*

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Target Location Longitude Offset</td><td>Type</td><td>Units</td></tr>
<tr><td>Float</td><td>Degrees</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">The set of real numbers from -19.2 to 19.2 inclusive</td><td>Precision</td></tr>
<tr><td>~2.3 micro degrees</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">06.0E.2B.34. 01.01.01.07 / 0E.01.02.03.42.00.00.00</td><td>Pack Tag</td></tr>
<tr><td>11</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>Signed Integer</td><td>Variable up to 3 Bytes</td><td>Binary</td></tr>
</table>

Notes

- Longitude offset for target from Frame Center Longitude (*0601*), based on WGS84 ellipsoid.
- Use with Frame Center Longitude.
- Map $-(2^{23}-1)..(2^{23}-1)$ to -/+ 19.2 degrees.
- Use $-(2^{23})$ as an "error" indicator.
- Resolution: ~2.3micro deg, ~0.25 meters at equator.
- Range: +/- 2136 km.

Conversion

Given:
The maximum measured value is +19.2 degrees
The minimum measured value is -19.2 degrees

If Mv is 10.00 degrees, then

$$Ev = (int)round\left(Emin + \frac{(Mv - Mmin)(Emax - Emin)}{(Mmax - Mmin)}\right)$$

$$= (int)round\left(-2^{23} + \frac{(10 - (-19.2))((2^{23} - 1) - (-2^{23}))}{(19.2 - (-19.2))}\right) = 4369066$$

If Ev is 4369066, then

$$Mv = \left(Mmin + \frac{(Ev - Emin)(Mmax - Mmin)}{(Emax - Emin)}\right) = \left(0 + \frac{(4369066 - (-2^{23}))(19.2 - (-19.2))}{((2^{23} - 1) - (-2^{23}))}\right)$$

$$= 10.00000021 \text{ degrees}$$

Example Value	Example Encoded LDS Value
10.00 Degrees	[K] [L] [V] = [0x0B][0x03][0x42 AA AA]

The target location has a real earth coordinate represented by a latitude-longitude pair. Target locations that lie above the horizon do not correspond to a point on the earth. Also, target locations may lie outside of the mapped range. Both cases should either not be reported, or be reported as an "error".

The Target Longitude Offset is added to the Frame Center Longitude metadata item from the parent *0601* packet to determine the Longitude of the target. Both KLV data items must be converted to decimal prior to addition to determine the actual measured or calculated motion imagery target location.

7.11.13 *Tag 12: Target Height*

Data Element Definition			
Element Name	Target Height	Type	Units
		Float	Meters
Valid Values	The set of real numbers from -900 to 19,000 inclusive		Precision
			~0.3 meters
KLV Encoding			
Universal Label	06.0E.2B.34. 01.01.01.07 / 0E.01.02.03.43.00.00.00		Pack Tag
			12
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	Variable up to 2 Bytes	Binary

Notes

- Height of the target, expressed as height in meters above the WGS84 ellipsoid.
- Map 0..(2^{16}-1) to -900..19000 meters.
- Resolution: ~0.3 meters.

Conversion

Given:
The maximum measured value is +19,000 meters
The minimum measured value is -900 meters

If Mv is 10,000 meters, then

$$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(10000 - 0)\,(2^{16} - 1)}{(19000 - (-900))}\right) = 35896$$

If Ev is 35896, then

$$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(-900 + \frac{35896\,(19000 - (-900))}{(2^{16} - 1)}\right) = 9{,}999.98 \text{ meters}$$

Example Value	Example Encoded LDS Value
10,000 Meters	[K] [L] [V] = [0x0C][0x02][0x8C 38]

Target Height is the Height Above the WGS84 Ellipsoid (HAE) of the target in meters.

7.11.14 ***Tag 13: Bounding Box Top Left Latitude Offset***

Data Element Definition			
Element Name	Bounding Box Top Left Latitude Offset	Type	Units
		Float	Degrees
Valid Values	The set of real numbers from -19.2 to 19.2 inclusive		Precision
			~2.3 micro degrees
KLV Encoding			
Universal Label	06.0E.2B.34. 01.01.01.07 / 0E.01.02.03.44.00.00.00		Pack Tag
			13
Volatility	Encoded Type	Length	Format
Dynamic	Signed Integer	Variable up to 3 Bytes	Binary

Notes

- Latitude offset for top left corner of target bounding box from Frame Center Latitude (*0601*), based on WGS84 ellipsoid.
- Use with Frame Center Latitude.
- Map $-(2^{23}-1)..(2^{23}-1)$ to -/+ 19.2 degrees.
- Use $-(2^{23})$ as an "error" indicator.
- Resolution: ~2.3micro deg, ~0.25 meters at equator.
- Range: +/- 2136 km.

Conversion

Given:
The maximum measured value is +19.2 degrees
The minimum measured value is -19.2 degrees

If Mv is 10.00 degrees, then

$$Ev = (int)round\left(Emin + \frac{(Mv - Mmin)\,(Emax - Emin)}{(Mmax - Mmin)}\right)$$
$$= (int)round\left(-2^{23} + \frac{(10 - (-19.2))\,((2^{23} - 1) - (-2^{23}))}{(19.2 - (-19.2))}\right) = 4369066$$

If Ev is 4369066, then

$$Mv = \left(Mmin + \frac{(Ev - Emin)\,(Mmax - Mmin)}{(Emax - Emin)}\right) = \left(0 + \frac{(4369066 - (-2^{23}))\,(19.2 - (-19.2))}{((2^{23} - 1) - (-2^{23}))}\right)$$
$$= 10.00000021 \text{ degrees}$$

Example Value	Example Encoded LDS Value
10.00 Degrees	[K] [L] [V] = [0x0D][0x03][0x42 AA AA]

The bounding box corners have real earth coordinates represented by a latitude-longitude pairs. Bounding box corners that lie above the horizon do not correspond to points on the earth. Also, bounding box corners may lie outside of the mapped range. Both cases should either not be reported, or be reported as an “error”.

The Bounding Box Top Left Latitude Offset is added to the Frame Center Latitude metadata item from the parent *0601* packet to determine the Latitude of the top left corner of the target bounding box. Both KLV data items must be converted to decimal prior to addition to determine the actual measured or calculated motion imagery bounding box corner location.

7.11.15 *Tag 14: Bounding Box Top Left Longitude Offset*

Data Element Definition			
Element Name	Bounding Box Top Left Longitude Offset	Type	Units
		Float	Degrees
Valid Values	The set of real numbers from -19.2 to 19.2 inclusive	Precision	
		~2.3 micro degrees	
KLV Encoding			
Universal Label	06.0E.2B.34. 01.01.01.07 / 0E.01.02.03.45.00.00.00	Pack Tag	
		14	
Volatility	Encoded Type	Length	Format
Dynamic	Signed Integer	Variable up to 3 Bytes	Binary

Notes

- Longitude offset for top left corner of target bounding box from Frame Center Latitude (*0601*), based on WGS84 ellipsoid.
- Use with Frame Center Longitude.
- Map -(2^{23}-1)..(2^{23}-1) to -/+19.2 degrees.
- Use -(2^{23}) as an "error" indicator.
- Resolution: ~2.3micro deg, ~0.25 meters at equator.
- Range: +/- 2136 km.

Conversion

Given:
The maximum measured value is +19.2 degrees
The minimum measured value is -19.2 degrees

If Mv is 10.00 degrees, then

$$Ev = (int)round\left(Emin + \frac{(Mv - Mmin)\,(Emax - Emin)}{(Mmax - Mmin)}\right)$$

$$= (int)round\left(-2^{23} + \frac{(10 - (-19.2))\,((2^{23} - 1) - (-2^{23}))}{(19.2 - (-19.2))}\right) = 4369066$$

If Ev is 4369066, then

$$Mv = \left(Mmin + \frac{(Ev - Emin)\,(Mmax - Mmin)}{(Emax - Emin)}\right) = \left(0 + \frac{(4369066 - (-2^{23}))\,(19.2 - (-19.2))}{((2^{23} - 1) - (-2^{23}))}\right) = 10.00000021 \text{ degrees}$$

Example Value	Example Encoded LDS Value
10.00 Degrees	[K] [L] [V] = [0x0E][0x03][0x42 AA AA]

The bounding box corners have real earth coordinates represented by a latitude-longitude pairs. Bounding box corners that lie above the horizon do not correspond to points on the earth. Also, bounding box corners may lie outside of the mapped range. Both cases should either not be reported, or be reported as an "error".

The Bounding Box Top Left Longitude Offset is added to the Frame Center Latitude metadata item from the parent *0601* packet to determine the Longitude of the top left corner of the target bounding box. Both KLV data items must be converted to decimal prior to addition to determine the actual measured or calculated motion imagery bounding box corner location.

7.11.16 *Tag 15: Bounding Box Bottom Right Latitude Offset*

Data Element Definition			
Element Name	Bounding Box Bottom Right Latitude Offset	Type	Units
		Float	Degrees
Valid Values	The set of real numbers from -19.2 to 19.2 inclusive		Precision
			~2.3 micro degrees
KLV Encoding			
Universal Label	06.0E.2B.34. 01.01.01.07 / 0E.01.02.03.46.00.00.00		Pack Tag
			15
Volatility	Encoded Type	Length	Format
Dynamic	Signed Integer	Variable up to 3 Bytes	Binary

Notes

- Latitude offset for bottom right corner of target bounding box from Frame Center Latitude (*0601*), based on WGS84 ellipsoid.
- Use with Frame Center Latitude.
- Map $-(2^{23}-1)..(2^{23}-1)$ to -/+ 19.2 degrees.
- Use $-(2^{23})$ as an "error" indicator.
- Resolution: ~2.3micro deg, ~0.25 meters at equator.
- Range: +/- 2136 km.

Conversion

Given:
The maximum measured value is +19.2 degrees
The minimum measured value is -19.2 degrees

If Mv is 10.00 degrees, then

$$Ev = (int)round\left(Emin + \frac{(Mv - Mmin)\,(Emax - Emin)}{(Mmax - Mmin)}\right)$$

$$= (int)round\left(-2^{23} + \frac{(10 - (-19.2))\,((2^{23} - 1) - (-2^{23}))}{(19.2 - (-19.2))}\right) = 4369066$$

If Ev is 4369066, then

$$Mv = \left(Mmin + \frac{(Ev - Emin)\,(Mmax - Mmin)}{(Emax - Emin)}\right) = \left(0 + \frac{(4369066 - (-2^{23}))\,(19.2 - (-19.2))}{((2^{23} - 1) - (-2^{23}))}\right) = 10.00000021 \text{ degrees}$$

Example Value	Example Encoded LDS Value
10.00 Degrees	[K] [L] [V] = [0x0F][0x03][0x42 AA AA]

The bounding box corners have real earth coordinates represented by a latitude-longitude pairs. Bounding box corners that lie above the horizon do not correspond to points on the earth. Also, bounding box corners may lie outside of the mapped range. Both cases should either not be reported, or be reported as an "error".

The Bounding Box Bottom Right Latitude Offset is added to the Frame Center Latitude metadata item from the parent *0601* packet to determine the Latitude of the bottom right corner of the target bounding box. Both KLV data items must be converted to decimal prior to addition to determine the actual measured or calculated motion imagery bounding box corner location.

7.11.17 ***Tag 16: Bounding Box Bottom Right Longitude Offset***

Data Element Definition			
Element Name	Bounding Box Bottom Right Longitude Offset	Type	Units
		Float	Degrees
Valid Values	The set of real numbers from -19.2 to 19.2 inclusive		Precision
			~2.3 micro degrees
KLV Encoding			
Universal Label	06.0E.2B.34. 01.01.01.07 / 0E.01.02.03.47.00.00.00		Pack Tag
			16
Volatility	Encoded Type	Length	Format
Dynamic	Signed Integer	Variable up to 3 Bytes	Binary

Notes

- Longitude offset for bottom right corner of target bounding box from Frame Center Latitude (*0601*), based on WGS84 ellipsoid.
- Use with Frame Center Longitude.
- Map -(2^{23}-1)..(2^{23}-1) to -/+ 19.2 degrees.
- Use -(2^{23}) as an "error" indicator..
- Resolution: ~2.3micro deg, ~0.25 meters at equator.
- Range: +/- 2136 km.

Conversion

Given:
The maximum measured value is +19.2 degrees
The minimum measured value is -19.2 degrees

If Mv is 10.00 degrees, then

$$Ev = (int)round\left(Emin + \frac{(Mv - Mmin)(Emax - Emin)}{(Mmax - Mmin)}\right)$$
$$= (int)round\left(-2^{23} + \frac{(10 - (-19.2))((2^{23} - 1) - (-2^{23}))}{(19.2 - (-19.2))}\right) = 4369066$$

If Ev is 4369066, then

$$Mv = \left(Mmin + \frac{(Ev - Emin)(Mmax - Mmin)}{(Emax - Emin)}\right) = \left(0 + \frac{(4369066 - (-2^{23}))(19.2 - (-19.2))}{((2^{23} - 1) - (-2^{23}))}\right)$$
$$= 10.00000021 \text{ degrees}$$

Example Value	Example Encoded LDS Value
10.00 Degrees	[K] [L] [V] = [0x10][0x03][0x42 AA AA]

The bounding box corners have real earth coordinates represented by a latitude-longitude pairs. Bounding box corners that lie above the horizon do not correspond to points on the earth. Also, bounding box corners may lie outside of the mapped range. Both cases should either not be reported, or be reported as an “error”.

The Bounding Box Bottom Right Longitude Offset is added to the Frame Center Latitude metadata item from the parent *0601* packet to determine the Longitude of the bottom right corner of the target bounding box. Both KLV data items must be converted to decimal prior to addition to determine the actual measured or calculated motion imagery bounding box corner location.

7.11.18 *Tag 17: Target Location*

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Target Location</td><td>Type</td><td>Units</td></tr>
<tr><td>Location Structure</td><td>NA</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">See Location Type definition</td><td>Precision</td></tr>
<tr><td>NA</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">06.0E.2B.34.02.05.01.01 / 0E.01.03.03.14.00.00.00</td><td>Pack Tag</td></tr>
<tr><td>17</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>Location</td><td>Variable</td><td>Binary</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- See the Location Type definition for a full description of this element.</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">NA</td></tr>
<tr><td colspan="2">Example Value</td><td colspan="2">Example Encoded LDS Value</td></tr>
<tr><td colspan="2">NA</td><td colspan="2">NA</td></tr>
</table>

Target Location provides detailed geopositioning information for the target, including the standard deviation and correlation coefficients. This element is of type Location and can be treated as a Defined Length Truncation Pack. A full description of the Location type can be found in paragraph 1.

7.11.19 *Tag 18: Target Boundary*

Data Element Definition			
Element Name	Target Boundary	Type	Units
		Boundary Structure	NA
Valid Values	See Boundary Type definition		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.04.01.01 / 0E.01.03.03.17.00.00.00		Pack Tag
			18
Volatility	Encoded Type	Length	Format
Dynamic	Boundary	Variable	Binary
Notes			
- See the Boundary Type definition for a full description of this element.			
Conversion			
NA			
Example Value		Example Encoded LDS Value	
NA		NA	

Target Boundary provides detailed geopositioning information for the boundary around an area or volume of interest. This boundary is defined by an arbitrary number of vertices. Each vertex is an element of type Location. Typical boundaries are the bounding boxes defined by two or four vertices.

The Location type captures geopositioning data about a specific location on or in close proximity to the surface of the Earth. The elements of these packs fall into three groups, namely, geospatial location (Latitude, Longitude, and Height), standard deviations for these values, and correlation coefficients among them. Location elements are structured as Defined-Length Truncation Packs, allowing unknown or less important elements to be omitted from the end of the Pack.

In general, use of Target Boundary is preferred over Target Bounding Box (Tags 13 through 16) if accuracy and correlation information is available. Such information is valuable for fusion with other moving object indicators, such as, radar-based GMTI, to support track identification and tracking.

A full description of the Boundary Type can be found in paragraph 1.

7.11.20 ***Tag 101: VMask LDS***

Data Element Definition			
Element Name	VMask LDS	Type	Units
		VMask Structure	NA
Valid Values	NA		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.03.01.01 / 0E.01.03.03.08.00.00.00		Pack Tag
			101
Volatility	Encoded Type	Length	Format
Dynamic	VMask LDS	Variable	Binary
Notes			
- None			
Conversion			
NA			
Example Value		Example Encoded LDS Value	
NA		NA	

The VMask LDS is intended to allow the inclusion of metadata elements that define the outline of the detected target within a video frame. This information can be used to redraw the outline in downstream clients or to “chip” the target from the video.

The VMask shape can be specified by either (1) a Series of three or more points that represent the vertices of a polygon within the video frame or (2) a bit mask that identifies the pixels of the video frame subsumed by the target.

7.11.20.1 *Tag 1: Polygon*

Data Element Definition			
Element Name	Polygon	Type	Units
		Array	NA
Valid Values	All integer values from 1 to 0xFFFFFFFFFFFF (281,474,976,710,655)		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.04.01.01 / 0E.01.03.03.18.00.00.00		LDS Tag
			1
Volatility	Encoded Type	Length	Format
Dynamic	Series of Unsigned Integers	Variable	Binary
Notes			
- A Series of three or more points that represent the vertices of a polygon within the video frame. - Each point is a pixel number with numbering commencing with 1, at the top left pixel, proceeding from left to right, top to bottom, then encoded using the Length-Value construct of a Variable-Length Pack.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
14762, 14783, 15115	[T][L] { [L][V] [L][V] [L][V] } = [0x01][0x09] [0x02][0x39 AA] [0x02][0x39 BF] [0x02][0x3B 0B]		

This element specifies a Series of three or more points that represent the vertices of a polygon within the video frame. The first point specified should also be treated as the last point, to close the polygon. The calculation of the pixel number uses the equation: X + ((Y-1) x Frame Width)). The top left pixel of the frame equates to (X,Y) = (1,1) and a pixel number of 1. Each pixel number shall be encoded using the Length-Value construct of a Variable-Length Pack.

7.11.20.2 *Tag 2: Bit Mask*

Data Element Definition			
Element Name	Bit Mask	Type	Units
		Array	NA
Valid Values	All integer values from 1 to 0xFFFFFFFFFFFF (281,474,976,710,655) for pixel number. All positive integers up to the maximum pixel number for run length.	Precision	
		NA	
KLV Encoding			
Universal Label	06.0E.2B.34.02.04.01.01 / 0E.01.03.03.19.00.00.00	LDS Tag	
		2	
Volatility	Encoded Type	Length	Format
Dynamic	Series of Unsigned Integers	Variable	Binary
Notes			
- A run-length encoding of a bit mask describing the pixels that subtend the target within the video frame. - A Series of pixel-number-plus-run-length pairs, each describing the starting pixel number and the number of pixels in a run. - Pixel numbering commences with 1, at the top left pixel, proceeding from left to right, top to bottom. - Pixel numbers are encoded using the Length-Value construct of a Variable-Length Pack. - The length of each run is encoded using BER Length encoding.			
Conversion			
NA			
Example Value		Example Encoded LDS Value	
NA		NA	

This element describes the area of the frame occupied by a target, using a bit mask, with 1 to indicate that a pixel subtends a part of the target and 0 to indicate otherwise. The criterion used to decide whether or not a particular pixel "covers" a part of the target is somewhat arbitrary and is left to the implementer. The implementer is free to decide whether overlap with all, a majority, or just a fraction of the pixel constitutes "covering" the target.

In general, it is expected a target will occupy only a relatively small portion of an image frame. To minimize the number of bytes required to encode the bit mask, (a form of) run-length encoding shall be used. Beginning with the upper left pixel of the frame, proceeding from left to right, row by row, each run of pixels that subtends a part of the target shall be encoded by specifying the number of the pixel at the start of the run and the number of pixels in the run.

The calculation of the pixel number uses the equation: X + ((Y-1) x Frame Width)). The top left pixel of the frame equates to (X,Y) = (1,1) and a pixel number of 1. The pixel number shall be encoded using the Length-Value construct of a Variable-Length Pack. The length of the run shall be encoded using BER Length encoding.

For example, in the 16 x 9 table below (Figure 6), the pairs of pixel numbers and run lengths (pixel, run) would be:

(74, 2) = [0x01 4A] [0x02]

(89, 4) = [0x01 59] [0x04]

(106, 2) = [0x01 6A] [0x02]

For this example, each run length is encoded in a single byte (as shown), using the short form of BER Length encoding. The long form of BER Length encoding would, of course, be used for run lengths exceeding 127 pixels. It is expected that run lengths will usually (but not necessarily) be “small”, leveraging the bit efficiency of the short form.

1	2	3	4	.	.	.									
17															
33															
49															
65									1	1					
81								1	1	1	1				
97									1	1					
113															
129															144

Figure 6. Sample Bit Mask

7.11.21 *Tag 102: VChip LDS*

Data Element Definition			
Element Name	VChip LDS	Type	Units
		VChip Structure	NA
Valid Values	NA		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.03.01.01 / 0E.01.03.03.13.00.00.00		Pack Tag
			102
Volatility	Encoded Type	Length	Format
Dynamic	VChip LDS	Variable	Binary
Notes			
- None			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
NA	NA		

The VChip LDS is intended to allow the inclusion of an image "chip" of the target. It is expected that this LDS will be used most commonly in bandwidth challenged environments, where the operator does not have access to the underlying video stream.

In general, the image chip will simply be "embedded" with the VMTI metadata. However, this specification permits reference to an image using a Uniform Resource Identifier/Locator (URI / URL) to support linking to a previously stored image, obviating the need to include the image data itself in the stream. This mechanism would most likely be used "downstream" of the collector. See VChip Tag 2: Image URI.

7.11.21.1 *Tag 1: Image Type*

Data Element Definition			
Element Name	Image Type	Type	Units
		String	NA
Valid Values	A string of 7-bit ISO characters that correspond to a MIME image type entered into the IANA Registry		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.30.00.00.00		LDS Tag
			1
Volatility	Encoded Type	Length	Format
Dynamic	String	Variable	ISO-7
Notes			
- Only a subset of MIME image types is permitted: jpeg and png.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
jpeg	[K] [L] [V] = [0x01][0x04][0x6A 70 65 67]		

This element is a Multi-Purpose Internet Extension (MIME) type specifying the VChip image type (*e.g.*, “jpeg”). A MIME registry is maintained by the Internet Assigned Numbers Authority (IANA) at http://www.iana.org/assignments/media-types/index.html. Only a subset of MIME image types is approved for use in *0903*, namely, jpeg and png.

The jpeg and png image types are chosen because they are common formats for compressed imagery. The most significant advantage of jpeg is that it is ubiquitous. A disadvantage is that it is lossy, although the degree of compression (and, hence, quality) is adjustable. The primary advantage of png is that it is lossless. Moreover, it provides RGB bit depths up to 48 bits per pixel (16 bits per component) and therefore provides a format that can preserve “raw” pixel values.

Note that MIME types in general encompass more than just image media types, including audio, video, message, and several other media types. At some future time, it might be reasonable to extend VChip to include video clips in addition to image chips.

7.11.21.2 *Tag 2: Image URI*

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Image_URI</td><td>Type</td><td>Units</td></tr>
<tr><td>String</td><td>NA</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">A string of 7-bit ISO characters that comply with the rules for building a valid URI</td><td>Precision</td></tr>
<tr><td>NA</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">06.0E.2B.34.01.01.01.01 / 0E.01.01.03.31.00.00.00</td><td>LDS Tag</td></tr>
<tr><td>2</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>ISO-7</td><td>Variable</td><td>ISO-7</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- None</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">NA</td></tr>
<tr><td colspan="2">Example Value</td><td colspan="2">Example Encoded LDS Value</td></tr>
<tr><td colspan="2">NA</td><td colspan="2">NA</td></tr>
</table>

The Image URI is a Uniform Resource Identifier (usually, a Uniform Resource Locator) that refers to an image of the type specified by VChip Tag1: Image Type, stored on a network or a file system. http://host.com/path/prettypicture.jpg is an example.

In some situations, probably downstream from the collection source, such a reference could be used in lieu of embedding the image chip in the stream.

7.11.21.3 *Tag 3: Embedded Image*

Data Element Definition			
Element Name	Embedded Image	Type	Units
		Wrapper	NA
Valid Values	Any image implemented in compliance with the MIME type specified in Tag 1.		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.32.00.00.00		LDS Tag
			3
Volatility	Encoded Type	Length	Format
Dynamic	Wrapper	Variable	Binary
Notes			
- None			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
NA	NA		

The Embedded Image is an image “chip”, of the type specified by VChip Tag1: Image Type, embedded in the VMTI stream.

7.11.22 ***Tag 103: VObject LDS***

Data Element Definition			
Element Name	VObject LDS	Type	Units
		VObject Structure	NA
Valid Values	NA		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.03.01.01 / 0E.01.03.03.09.00.00.00		Pack Tag
			103
Volatility	Encoded Type	Length	Format
Dynamic	VObject LDS	Variable	Binary
Notes			
- None			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
NA	NA		

The VObject LDS is intended to allow the inclusion of data that describes the class or type of a target (aircraft, watercraft, car, truck, train, dismount, etc), to an arbitrary level of detail. For example, it might be useful to expand the notion of a “dismount” to include combatant, noncombatant, male, female, etc.

However, any attempt to specify a “complete” set of target types or classes within *0903* is destined to fail. Some users may find a predetermined enumeration satisfactory, but others may find it to be lacking. For example, suppose dismounts are characterized as combatant, noncombatant, male, and female. What if a user would like to classify dismounts further as adult and child?

To address this problem, *0903* employs the concept of an ontology, a “vocabulary” to model entities and concepts, along with their properties and relations. Using this approach, a community of users can define a new ontology or, more likely, to extend an existing ontology to meet their specific needs.

0903 mandates the use of the Web Ontology Language (OWL), developed by the World Wide Web Consortium (W3C), to define the VObject ontology.

The OWL Web Ontology Language Reference can be found at http://www.w3.org/TR/owl-ref.

7.11.22.1 *Tag 1: Ontology*

Data Element Definition			
Element Name	Ontology	Type	Units
		String	NA
Valid Values	A URI constructed from 7-bit ISO characters, which refers to a valid OWL ontology		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.31.00.00.00		LDS Tag
			1
Volatility	Encoded Type	Length	Format
Dynamic	ISO-7	Variable	Character
Notes			
- None			
Conversion			
NA			
Example Value		Example Encoded LDS Value	
NA		NA	

This element is a Uniform Resource Identifier (URI) which refers to a VObject ontology. The ontology shall be expressed using the Web Ontology Language OWL.

The Jet Propulsion Laboratory Semantic Web for Earth and Environmental Terminology (SWEET) (http://sweet.jpl.nasa.gov/ontology/) provides a collection of ontologies, written in the OWL ontology language that can serve as examples and starting points for the development of additional domain-specific extended ontologies.

7.11.22.2 *Tag 2: Class*

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Class</td><td>Type</td><td>Units</td></tr>
<tr><td>String</td><td>NA</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">A string of 7-bit ISO characters that comprise the name of the class, as defined in the VObject Ontology.</td><td>Precision</td></tr>
<tr><td>NA</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">06.0E.2B.34.01.01.01.01 / 0E.01.01.03.33.00.00.00</td><td>LDS Tag</td></tr>
<tr><td>2</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>String</td><td>Variable</td><td>Character</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- None</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">NA</td></tr>
<tr><td colspan="2">Example Value</td><td colspan="2">Example Encoded LDS Value</td></tr>
<tr><td colspan="2">NA</td><td colspan="2">NA</td></tr>
</table>

The VObject Class is the name of the target class or type, as defined in the VObject Ontology.

7.11.23 *Tag 104: VFeature LDS*

Data Element Definition			
Element Name	VFeature LDS	Type	Units
		VFeature Structure	NA
Valid Values	NA		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.03.01.01 / 0E.01.03.03.0A.00.00.00		Pack Tag
			104
Volatility	Encoded Type	Length	Format
Dynamic	VFeature LDS	Variable	Binary
Notes			
- None			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
NA	NA		

The VFeature LDS is intended to allow the inclusion of data that describes the target or features of the target (dents, wheels, number of passengers etc). Descriptive information can range from simple text for a label, to complex data sets containing spectral or radiometric data.

The definition of a set of elements to describe target features can be a complex undertaking. Rather than create a unique specification for the VFeature LDS, 0903 has adopted the Open Geospatial Consortium (OGC) Geography Markup Language (GML), as specified in ISO Standard 19136 [4], and has based the VFeature LDS upon the OGC Observations and Measurements (O&M) Encoding Standard and related schemas.

The O&M standard is described in two documents: (1) *Observations and Measurements - Part 1 - Observation schema*, and (2) *Observations and Measurements - Part 2 - Sampling Features*.

ISO Technical Committee 211 is developing ISO Standard 19156 based upon the O&M standard.

7.11.23.1 *Tag 1: Schema*

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Schema</td><td>Type</td><td>Units</td></tr>
<tr><td>String</td><td>NA</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">A URI constructed from 7-bit ISO characters which refers to the OGC Observation schema (http://schemas.opengis.net/om/1.0.0/) or a related schema.</td><td>Precision</td></tr>
<tr><td>NA</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">06.0E.2B.34.01.01.01.01 / 0E.01.01.03.31.00.00.00</td><td>LDS Tag</td></tr>
<tr><td>1</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>ISO-7</td><td>Variable</td><td>Character</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- None</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">NA</td></tr>
<tr><td colspan="2">Example Value</td><td colspan="2">Example Encoded LDS Value</td></tr>
<tr><td colspan="2">NA</td><td colspan="2">NA</td></tr>
</table>

This element is a Uniform Resource Identifier (URI) which points to the OGC Observation schema (http://schemas.opengis.net/om/1.0.0/) or a related schema.

OGC defines an Observation as "an action with a result which has a value describing some phenomenon". The Observation is defined as a Feature within the context of the OGC General Feature Model [ISO 19101, ISO 19109]. An Observation may involve use of a sensor or observer, analytical procedure, simulation, or other mathematical process. An Observation yields an estimate of the value of a property of the feature of interest, and can account for error that may be present in the estimate.

Observation values may have many data types, from primitive to complex, including category, measure, and geometry. Combinations of data types can be used to encode properties having multiple components. The notion of "coverage" can be used for properties that vary over the feature of interest.

O&M Sampling addresses the sampling sub-elements of a feature that are used to represent the whole.

7.11.23.2 *Tag 2: Feature*

Data Element Definition			
Element Name	Feature	Type	Units
		Wrapper	NA
Valid Values	Any OGC GML document that validates against the schema specified in Tag 1: Schema		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.34.00.00.00		LDS Tag
			2
Volatility	Encoded Type	Length	Format
Dynamic	Wrapper	Variable	Character
Notes			
- None			
Conversion			
NA			
Example Value		Example Encoded LDS Value	
NA		NA	

The VFeature Feature is an OGC GML document structured according to the schema specified by VFeature Tag 1: Schema. It contains one or more values observed for a feature of interest.

7.11.24 *Tag 105: VTracker LDS*

Data Element Definition			
Element Name	VTracker LDS	Type	Units
		VTracker Structure	NA
Valid Values	NA		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.03.01.01 / 0E.01.03.03.0B.00.00.00		Pack Tag
			17
Volatility	Encoded Type	Length	Format
Dynamic	VTracker LDS	Variable	Binary
Notes			
- None			
Conversion			
NA			
Example Value		Example Encoded LDS Value	
NA		NA	

The VTarget LDS is intended to contain spatial and temporal information ancillary to VChip, VObject, and VFeature, to assist in tracking the target. Such information will allow motion imagery tracking algorithms to create better tracks from the VMTI target information.

Note that use of the VTrack (no “er”) LDS defined in §8 is generally preferred over the VTracker LDS for representation of target tracks.

Tag 1: Target ID

Data Element Definition			
Element Name	Target ID	Type	Units
		Unsigned Integer	NA
Valid Values	The set of all integers from 0 to $2^{64} – 1$ [0xFFFFFFFFFFFFFFFF] inclusive		Precision
			1
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.35.00.00.00		LDS Tag
			1
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	Variable up to 8 Bytes	Binary
Notes			
- None			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
4,294,965,678 [0xFFFFF9AE]	[K] [L] [V] = [0x01][0x04][0xFF FF F9 AE]		

The Target ID is intended to be a unique identifier for all VMTI detections known or believed to be the same entity. Ideally, it would be unique in the Universe for all time, but specification of a method to achieve such uniqueness is beyond the scope of this document.

7.11.24.1 *Tag 2: Detection Status*

Data Element Definition			
Element Name	Detection Status	Type	Units
		Unsigned Integer	NA
Valid Values	Detection Status is an enumerated data element with values assigned from Table 13 below.		Precision
			1
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.36.00.00.00		LDS Tag
			2
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	1 Byte	Binary
Notes			
- None			
Conversion			
NA			
Example Value		Example Encoded LDS Value	
1 [0x01]		[K] [L] [V] = [0x02][0x01][0x01]	

Detection Status indicates the state of VMTI detections for a given entity. The values in Table 13 should be defined consistently with those of *NATO STANAG 4676 Tracking Standard for ISR Systems* (draft).

Value	Status
0	Inactive
1	Active
2	Dropped
3	Stopped

Table 13: Track Status Values

- Active – Detections for the entity have been established or updated on the basis of an associated VMTI report or a prediction. An entity can resume this state by transition from Stopped or Dropped to "moving", when a VMTI detection (or a prediction) with a new position has become associated with it.
- Dropped – The entity could not be correlated with any VMTI detection for an interval of time that exceeds a particular threshold. An entity can remain in a Dropped, or "lost", condition for an indeterminate period of time, if there is some likelihood that it might be Resumed. Eventually, it may become Inactive.
- Stopped – The entity has either become stationary or was always in a fixed location.
- Inactive – The VMTI detections for the entity have ended. The entity may have merged with one or more other entities, to have split into two or more new entities, or to have ceased to exist because no VMTI detection can be correlated with it.

7.11.24.2 *Tag 3: Start Time Stamp*

Data Element Definition			
Element Name	Start Time Stamp	Type	Units
		Unsigned Long Integer	Microseconds
Valid Values	All integer values from 0 to (2 ^ 64) – 1		Precision
			1 microsecond
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.03 / 07.02.01.01.01.05.00.00		LDS Tag
			3
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	Variable up to 8 Bytes	Binary
Notes			
- Microseconds elapsed since midnight (00:00:00), January 1, 1970. - Derived from the POSIX IEEE 1003.1 standard.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
April 19 2001, 04:25:21 GMT	[K] [L] [V] = [0x03] [0x08] [0x00 03 82 44 30 F6 CE 40]		

The Start Time Stamp captures the date and time of the first observation of the entity.

7.11.24.3 *Tag 4: End Time Stamp*

Data Element Definition			
Element Name	End Time Stamp	Type	Units
		Unsigned Long Integer	Microseconds
Valid Values	All integer values from 0 to (2 ^ 64) – 1		Precision
			1 microsecond
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.03 / 07.02.01.01.01.05.00.00		LDS Tag
			4
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	Variable up to 8 Bytes	Binary
Notes			
- Microseconds elapsed since midnight (00:00:00), January 1, 1970. - Derived from the POSIX IEEE 1003.1 standard.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
April 19 2001, 04:25:21 GMT	[K] [L] [V] = [0x04] [0x08] [0x00 03 82 44 30 F6 CE 40]		

The End Time Stamp captures the date and time of the most recent observation of the entity.

7.11.24.4 *Tag 5: Bounding Box*

Data Element Definition			
Element Name	Bounding Box	Type	Units
		Boundary Structure	NA
Valid Values	See VMTI Boundary Type Description		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.04.01.01 / 0E.01.03.03.17.00.00.00		LDS Tag
			5
Volatility	Encoded Type	Length	Format
Dynamic	Boundary	Variable	Binary
Notes			
- None			
Conversion			
NA			
Example Value		Example Encoded LDS Value	
NA		NA	

The Bounding Box comprises a set of Boundary vertices that specify a minimum bounding area or volume, which encloses the full extent of VMTI detections for the entity. For a simple, planar bounding box, the area will generally lie on the surface of the Earth (although not necessarily, depending upon the Height values provided), in essence defining the "footprint" of the track. By specifying additional vertices, complex, multifaceted volumes can be described. In this case, the Boundary type requires vertices to be ordered so that looking toward Earth center, they spiral in a clockwise direction, from lowest elevation to highest.

7.11.24.5 *Tag 6: Algorithm*

Data Element Definition			
Element Name	Algorithm	Type	Units
		String	NA
Valid Values	The set of all 7-bit ISO characters		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.02.03.49.00.00.00		LDS Tag
			6
Volatility	Encoded Type	Length	Format
Dynamic	ISO-7	Variable	Character
Notes			
- None			
Conversion			
NA			
Example Value		Example Encoded LDS Value	
"test" [74 65 73 74]		[K] [L] [V] = [0x06][0x04][74 65 73 74]	

The Algorithm is the name or description of the algorithm or method used to create or maintain object movement reports or intervening predictions of such movement. The intent of this element is to identify uniquely the VMTI algorithm or method used.

7.11.24.6 *Tag 7: Confidence*

Data Element Definition			
Element Name	Confidence	Type	Units
		Unsigned Short	NA
Valid Values	The set of all integers from 0 to 100 inclusive		Precision
			1
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.37.00.00.00		LDS Tag
			7
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	1 Byte	Binary
Notes			
- Confidence level, expressed as a percentage. - Value 0 percent indicates no confidence; value 100 percent indicates absolute certainty.			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
50 [0x32]	[K] [L] [V] = [0x07][0x01][0x32]		

Confidence is an estimation of the certainty or correctness of the VMTI movement detections. Larger values indicate greater confidence. Zero indicates no confidence. A high degree of confidence might be indicated for detections derived from a large number of unambiguous target reports, such as, for a single vehicle on a road in a desert environment. A lower degree of confidence might be appropriate when target reports could be associated with several overlapping or nearby tracks in a partially obscured environment, such as, for dismounts (people) in an urban setting.

7.11.24.7 *Tag 8: Number of Points*

Data Element Definition			
Element Name	Number of Points	Type	Units
		Unsigned Short	NA
Valid Values	The set of all integers from 1 to 65,536 inclusive		Precision
			1
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.38.00.00.00		LDS Tag
			8
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	Variable up to 2 Bytes	Binary
Notes			
- None			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
27 [0x1B]	[K] [L] [V] = [0x08][0x01][0x1B]		

Number of Points is the number of coordinates of type Location that describe the locus of entity VMTI detections. The Number of Points must be at least 1, else no locus would exist.

7.11.24.8 *Tag 9: Locus*

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Locus</td><td>Type</td><td>Units</td></tr>
<tr><td>Array</td><td>NA</td></tr>
<tr><td rowspan="2">Valid Values</td><td rowspan="2" colspan="2">See Series Description</td><td>Precision</td></tr>
<tr><td>NA</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td rowspan="2" colspan="2">06.0E.2B.34.02.04.01.01 / 0E.01.03.03.1A.00.00.00</td><td>LDS Tag</td></tr>
<tr><td>9</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>Series of Location data elements</td><td>Variable</td><td>Binary</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- None</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">NA</td></tr>
<tr><td colspan="2">Example Value</td><td colspan="2">Example Encoded LDS Value</td></tr>
<tr><td colspan="2">NA</td><td colspan="2">NA</td></tr>
</table>

The Locus is a Series (see §6.5) of points that represent the locations of entity VMTI detections. Each point is an element of type Location. The points should be ordered chronologically from start to end of the VMTI detections.

7.11.24.9 *Tag 10: Velocity*

Data Element Definition			
Element Name	Velocity	Type	Units
		Velocity Structure	NA
Valid Values	See Velocity Type Description		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.05.01.01 / 0E.01.03.03.15.00.00.00		LDS Tag
			10
Volatility	Encoded Type	Length	Format
Dynamic	Velocity	Variable	Binary
Notes			
- None			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
NA	NA		

Velocity is the velocity of the entity at the time of last observation.

7.11.24.10 *Tag 11: Acceleration*

Data Element Definition			
Element Name	Acceleration	Type	Units
		Acceleration Structure	NA
Valid Values	See Acceleration Type Description		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.05.01.01 / 0E.01.03.03.16.00.00.00		LDS Tag
			11
Volatility	Encoded Type	Length	Format
Dynamic	Acceleration	Variable	Binary
Notes			
- None			
Conversion			
NA			
Example Value	Example Encoded LDS Value		
NA	NA		

Acceleration is the acceleration of the entity at the time of last observation

8 Track Metadata

This section defines the track metadata LDS, which is named VTrack. With few exceptions, the data elements of the VTrack LDS are the same as those defined for the VMTI LDS. Whereas the VMTI LDS describes detections of object movement associated with a single frame, the VTrack LDS describes a set of detections that may span many frames and that are presumed to be associated with a single, identified object. The VTrack LDS adopts a "track-centric" view of the detection data, whereas the VMTI LDS has a "frame-centric" view. The data elements used to describe a detection are the same, regardless of which view is taken, but they are organized differently. Appendix C portrays the UML[11] data model of the VMTI LDS, and Appendix D depicts the data model of the VTrack LDS. Examination and comparison of the two models should make the differences in organization readily apparent.

The VMTI LDS contains an element VTargetSeries, which is comprised of a set of VTarget detections for a given frame. Similarly, the VTrack LDS contains an element VTrackItemSeries, which is comprised of a set of VTrackItem detections associated with a track.

The VTrack LDS and its constituent VTrackItem Packs provide a mechanism for reporting complete tracks, from the first detection of an object movement to the latest. However, VTrack can also be used to report track updates, including new detections that extend a previously reported track, or updates to previously reported track information.

Tables summarizing the VTrack LDS and the VTrackItem Pack are defined in this section, with descriptions of each element unique to the VTrack LDS (those not found in the VMTI LDS).

Note that the VMTI LDS contains an element, VTracker[12], which could be used as an alternative to VTrack to specify track metadata. VTracker contains spatial and temporal information for a given detection within a frame, including preceding detections of the same (presumed) object. Use of VTracker is discouraged (although not forbidden). Use of VTrack is recommended, because it maps much more directly to NATO STANAG 4676, the NATO ISR Tracking Standard.[13] Appendix E describes the correspondence between VTrack data elements and STANAG 4676 data elements.

8.1 VTrack Local Data Set

The VTrack LDS, shown in Table 14, consists primarily of metadata elements that appear in the VMTI LDS (and its VTracker Pack), plus two additional data elements, Track ID and VTrackItemSeries.

---

[11] Unified Modeling Language (UML) is a specification developed by Object Management Group, a not-for-profit computer industry consortium, defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems.

[12] Note the distinction between VTrack<u>er</u> and VTrack (no "er").

[13] As of this date, STANAG 4676 is not yet a ratified NATO standard (still classified as a Study). Moreover, STANAG 4676 currently lacks the extensive set of motion imagery feature information provided by VTrack to support object disambiguation and identification. Also, STANAG 4676 currently recognizes only XML as a data encoding method, which (because of its "verbosity") limits its use in bandwidth-limited communications environments. Use of a more efficient data encoding method (such as, KLV) might mitigate this deficiency. As STANAG 4676 matures to address these deficiencies, it may become the preferred representation for motion imagery derived tracks.

Track ID is a value that uniquely identifies a track, using a 128-bit (16-byte) Universal Unique Identification (UUID) as standardized by the Open Software Foundation in ISO/IEC 9834-8:2005 [16].

VTrackItemSeries contains a time-sequenced set of VTrackItem Packs that describe points (e.g., VMTI detections) along the track of a moving object. Such track points are often connected using straight line segments to describe (an approximation of) the path taken by the moving object. However, this Standard does not restrict interpretation of the points solely to this manifestation.

Table 14: VTrack LDS

VTrack LDS							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.03.01.01 0E.01.03.03.1E.00.00.00	VTrack KLV Dictionary	N/A	None	N/A	Variable	This is the Universal Key for the VTrack LDS
01	06.0E.2B.34.04.01.01.01 0E.01.02.03.01.00.00.00	Checksum					See Table 1: VMTI LDS.
02	06.0E.2B.34.01.01.01.03 07.02.01.01.01.05.00.00	Track Time Stamp					See Table 1: VMTI LDS.
03	06.0E.2B.34.01.01.01.01 0E.01.02. 03.5C.00. 00.00	Track ID	Unsigned Integer	NA	Uint128	F16	A unique identifier (UUID) for the track.
04	06.0E.2B.34.01.01.01.01 0E.01.01. 03.36.00. 00.00	Track Status					See Table 7: VTracker LDS.
05	06.0E.2B.34.01.01.01.03 07.02.01. 01.01.05. 00.00	Track Start Time					See Table 7: VTracker LDS.
06	06.0E.2B.34.01.01.01.03 07.02.01. 01.01.05. 00.00	Track End Time					See Table 7: VTracker LDS.
07	06.0E.2B.34.02.04.01.01 0E.01.03. 03.17.00. 00.00	Track Bounding Box					See Table 7: VTracker LDS.
08	06.0E.2B.34.01.01.01.01 0E.01.02. 03.49.00. 00.00	Tracker Algorithm					See Table 7: VTracker LDS.
09	06.0E.2B.34.01.01.01.01 0E.01.01. 03.37.00. 00.00	Track Confidence					See Table 7: VTracker LDS.

VTrack LDS							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
10	06.0E.2B.34.01.01.01.01 0E.01.02. 02.7C.00. 00.00	VMTI System Name / Description					See Table 1: VMTI LDS.
11	06.0E.2B.34.01.01.01.01 0E.01.02.05.04.00.00.00	VMTI LDS Version Number					See Table 1: VMTI LDS.
12	06.0E.2B.34.01.01.01.01 04.20.01.02.01.01.00.00	VMTI Source Sensor					See Table 1: VMTI LDS.
13	06.0E.2B.34.01.01.01.01 0E.01.01.03.38.00.00.00	Number of Track Items					See Table 7: VTracker LDS.
101	06.0E.2B.34.02.03.01.01 0E.01.03.03.1F.00.00.00	VTrackItemSeries	Array	NA	NA	V	Series of track item metadata elements, each of which is a VTrackItem Pack as defined in Table 15: VTrackItem.

8.2 VTrackItem Pack

The VTrackItem Pack, shown in Table 15, contains data elements from the VMTI LDS and the VMTI VTarget Pack to describe not only the geographic location of a track point, but also a rich characterization of the moving object, including velocity, acceleration, appearance (features), and type. Furthermore, VTrackItem supports identification and reference (“linking”) to the motion imagery essence in which the moving object appeared and from which detections of its movement were made.

Reference to the motion imagery essence is by means of a Uniform Resource Identifier (URI), which uniquely identifies a motion imagery stream or file. The Video Essence URI may be a Uniform Resource Name (URN), compliant with MISB RP 1204 Motion Imagery Identification System. Alternatively, the Video Essence URI may be a Uniform Resource Locator (URL) that not only uniquely identifies the Video Essence but also provides a means for locating the essence by describing its access mechanism (e.g., its network “location”). Internet Standard RFC 3986 [13] defines URI syntax and a process for resolving URI references.

Table 15: VTrackItem Pack

VTrackItem Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
	06.0E.2B.34.02.03.01.01 0E.01.03.03.20.00.00.00	VTrackItem Pack	N/A	None	N/A	Variable	This is the Universal Key for the VTrackItem Pack. This is a Truncation Pack
N/A	N/A	Target ID Number					See Table 7: VTracker LDS.
01	06.0E.2B.34.01.01.01.03 07.02.01.01.01.05.00.00	Target Time Stamp					See Table 1: VMTI LDS.

VTrackItem Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
02	06.0E.2B.34.01.01.01.01 0E.01.02.03.38.00.00.00	Target Centroid Pixel Number					See Table 2: VTarget Pack.
03	06.0E.2B.34.01.01.01.01 0E.01.02.03.58.00.00.00	Target Centroid Pixel Row					See Table 2: VTarget Pack.
04	06.0E.2B.34.01.01.01.01 0E.01.02.03.59.00.00.00	Target Centroid Pixel Column					See Table 2: VTarget Pack.
05	06.0E.2B.34.01.01.01.01 0E.01.02.03.39.00.00.00	Bounding Box Top Left Pixel Number					See Table 2: VTarget Pack.
06	06.0E.2B.34.01.01.01.01 0E.01.02.03.3A.00.00.00	Bounding Box Bottom Right Pixel Number					See Table 2: VTarget Pack.
07	06.0E.2B.34.01.01.01.01 0E.01.02.03.3B.00.00.00	Target Priority					See Table 2: VTarget Pack.
08	06.0E.2B.34.01.01.01.01 0E.01.02.03.3C.00.00.00	Target Confidence Level					See Table 2: VTarget Pack.
09	06.0E.2B.34.01.01.01.01 0E.01.02.03.3D.00.00.00	New Detection Flag / Target History					See Table 2: VTarget Pack.
10	06.0E.2B.34.01.01.01.01 0E.01.02.03.3E.00.00.00	Percentage of Target Pixels					See Table 2: VTarget Pack.
11	06.0E.2B.34.01.01.01.01 0E.01.02.03.3F.00.00.00	Target Color					See Table 2: VTarget Pack.
12	06.0E.2B.34.01.01.01.01 0E.01.02.03.40.00.00.00	Target Intensity					See Table 2: VTarget Pack.
13	06.0E.2B.34.02.05.01.01 0E.01.03.03.14.00.00.00	Target Location					See Table 2: VTarget Pack.
14	06.0E.2B.34.02.04.01.01 0E.01.03.03.17.00.00.00	Target Boundary					See Table 2: VTarget Pack.
15	06.0E.2B.34.02.05.01.01 0E.01.03.03.15.00.00.00	Velocity					See Table 7: VTracker LDS.
16	06.0E.2B.34.02.05.01.01 0E.01.03.03.16.00.00.00	Acceleration					See Table 7: VTracker LDS.
17	06.0E.2B.34.02.03.01.01 0E.01.03.03.1D.00.00.00	WAMI FPA Index					See Table 2: VTarget Pack.

VTrackItem Pack							
Tag ID	Key Value (hex)	Key Name	Data Type	Units	KLV Format	Length in Bytes	Notes
18	06.0E.2B.34.01.01.01.01 0E.01.01.03.1F.00.00.00	Video Frame Number					See Table 1: VMTI LDS.
19	06.0E.2B.34.01.01.01.01 0E.01.02.03.5D.00.00.00	Video Essence	String	NA	ISO-7	V	A Uniform Resource Identifier (URI), which uniquely identifies a motion imagery stream or file. The Video Essence URI may be a Uniform Resource Name (URN), compliant with MISB RP 1204 Motion Imagery Identification System. Alternatively, the Video Essence URI may be a Uniform Resource Locator (URL) that not only uniquely identifies the Video Essence but also provides a means for locating the essence by describing its access mechanism (e.g., its network "location"). See Internet Standard RFC 3986 [13].
20	06.0E.2B.34.01.01.01.01 0E.01.01.02.07.00.00.00	Frame Width					See Table 1: VMTI LDS.
21	06.0E.2B.34.01.01.01.01 0E.01.01.02.08.00.00.00	Frame Height					See Table 1: VMTI LDS.
22	06.0E.2B.34.01.01.01.02 04.20.02.01.01.08.00.00	Sensor Horizontal Field of View					See Table 1: VMTI LDS.
23	06.0E.2B.34. 01.01.01.07 04.20.02.01.01.0A.01.00	Sensor Vertical Field of View					See Table 1: VMTI LDS.
101	06.0E.2B.34.02.03.01.01 0E.01.03.03.08.00.00.00	VMask LDS					See Table 3: VMask LDS.
102	06.0E.2B.34.02.03.01.01 0E.01.03.03.09.00.00.00	VObject					See Table 5: VObject LDS.
103	06.0E.2B.34.02.03.01.01 0E.01.03.03.0A.00.00.00	VFeature LDS					See Table 6: VFeature LDS.
105	06.0E.2B.34.02.03.01.01 0E.01.03.03.13.00.00.00	VChip LDS					See Table 4: VChip LDS.

8.3 Required VTrack and VTrackItem Data Elements

Both the VTrack LDS and the VTrackItem Pack contain data elements that might not be required every time VTrack and its constituent VTrackItems are reported, either because the tracking system is incapable of producing such data or because the data changes infrequently.

Especially when sending metadata in a bandwidth-constrained environment, it is prudent and justified to devote most of the available bandwidth to information that changes rapidly, sending less frequently changing information less often. For example, the location of the moving object should be provided at every opportunity, whereas an image chip (VChip) of the object might be provided just “once in a while”.

Data elements that must be provided every time a track is reported or updated are listed in Table 16.

Table 16: Required Metadata Elements[14]

Required Metadata Elements			
VTrack	VTrackItem	Tag ID	Name
X		01	Checksum
X		02	Track Time Stamp (UNIX Time Stamp)
X		03	Track ID
	X	N/A	Target ID Number
		01	Target Time Stamp
		12	Target Location

[14] These required metadata elements are consistent with those required by STANAG 4676.

9 Glossary of Acronyms

BER	Basic Encoding Rules
DMA	Defense Mapping Agency (now National Geospatial-Intelligence Agency)
EG	Engineering Guideline
EO	Electro-Optic
EON	Electro-Optic Narrow
EOW	Electro-Optic Wide
FOV	Field of View
FPA	Focal Plane Array
GML	Geography Markup Language
HFOV	Horizontal Field of View
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
KLV	Key-Length-Value
ISS	Image Source Sensor
LDS	Local Data Set
MISB	Motion Imagery Standards Board
MSB	Most Significant Byte
msb	Most Significant Bit
OGC	Open Geospatial Consortium
SMPTE	Society of Motion Picture and Television Engineers
TLV	Tag, Length, Value
TRM	Technical Reference Manual / Material
TS	Transport Stream
UAS	Unmanned Aerial / Airborne System
UAV	Unmanned Aerial / Airborne Vehicle
UINT	Unsigned Integer
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Coordinate Universal Time
VFOV	Vertical Field of View
VMTI	Video Moving Target Indicator
VSS	VMTI Source Sensor
WAMI	Wide Area Motion Imagery

Appendix A. KLV Users Guide [Normative]

For a tutorial on KLV, refer to MISB *TRM 1006 Key-Length-Value (KLV) Users Guide* [20].

A.1 Key to Tag Mappings

Within the context of a KLV data structure (such as a Local Data Set), the 16-byte Key may be replaced by a short one- or two-byte “Tag”, resulting in a Tag-Length-Value (TLV) element. This mapping saves bandwidth by substituting a compact Tag for the full 16-byte Key.

Note: It is possible to associate multiple Tags with a single Key, because each instantiation within the LDS carries a unique Tag and is therefore an unambiguous reference.

A.2 Tag and Length Formats

VMTI LDS Tags are represented in the KLV stream using BER-OID encoded unsigned integers. Within this specification, they are expressed using decimal values. (See *TRM 1006* for a discussion of BER-OID encoding.)

Length fields in the VMTI LDS are unsigned integers encoded using ASN.1 BER short or long form. This is consistent with the encoding indicated by bytes 5 and 6 in the 16-byte Universal Label.

A.3 Bit and Byte Order

The bit and byte order in *0903* is big-endian (see [12]).

A.4 Variable Length Encoding

The *0903* VMTI LDS pioneers the use of variable length encoding for integer values. When a floating point value is mapped to an integer value, the integer value shall be encoded using a variable length encoding (unless otherwise specified).

The specification of each *0903* variable-length-encoded metadata element shall define the maximum number of bytes required in order to represent the possible range of integer values. The number of bytes used to encode the integer value shall be less than or equal to the specified maximum length. A single byte shall be used to encode the value zero.

Although 0903 specifies the maximum number of bytes for each variable-length element, there is from a KLV perspective no compelling reason to do so, since the KLV length field must be present for such elements anyway. Well-constructed encoders and decoders should be capable of dealing with any reasonable length, even if it exceeds the specified maximum. However, this information is useful – in fact, necessary – when a floating point value is to be encoded as an integer, as described in this Appendix. In this case, the maximum number of bytes connotes the precision of the result. Nonetheless, if a reasonable person would find it proper to exceed the stated maximum, then the relevant encoder and decoder shall be capable of correctly processing the encoded value.

For unsigned integers, values in the range $[0, 2^8 - 1]$ can be represented by a single byte, values in $[0, 2^{16} - 1]$ by two bytes, $[0, 2^{24} - 1]$ by three bytes, etc. The leading zeros of an unsigned integer are insignificant. For example, the value 255 can be represented by the four byte value 0x000000FF and can also be represented by the single byte 0xFF.

To demonstrate the variable length encoding, consider VTarget Pack Tag 0x01 (Target Centroid Pixel Number). This element is defined to be an unsigned integer value with variable length up to 6 bytes. The following is the Tag-Length-Value (TLV) encoding for a centroid pixel number value of 200: {[0x01] [0x01] [0xC8]}. The following is the TLV encoding for a centroid pixel number value of 123456: {[0x01] [0x03] [0x01E240]}.

For signed integers, values in the range $[-2^7, 2^7 - 1]$ can be represented by a single byte, values in $[-2^{15}, 2^{15} - 1]$ by two bytes, $[-2^{23}, 2^{23} - 1]$ by three bytes, etc. Signed integers are encoded using the two's complement system. When changing the number of bytes of a signed integer, care must be taken to preserve both the magnitude and sign of the value. Truncating leading 0x00 or 0xFF bytes from a signed integer representation will preserve the integer's value if and only if the most significant bit remains unchanged. For example, the four-byte value 0x000000FF (binary <u>0</u>000 0000 0000 0000 0000 0000 1111 1111) represents the signed integer value of 255, the two-byte value 0x00FF (binary <u>0</u>000 0000 1111 1111) also represents the signed integer value of 255, whereas the single-byte value 0xFF (binary <u>1</u>111 1111) represents the signed integer value of −1. Note that the most significant bit is clear for both four- and two-byte representations, whereas the most significant bit is set for the one-byte representation. Likewise, the three leading 0xFF can be safely truncated from the four-byte value 0xFFFFFFFF (binary <u>1</u>111 1111 1111 1111 1111 1111 1111 1111) to give the single-byte value 0xFF (binary <u>1</u>111 1111). Both values represent the signed integer value of −1.

Sign-extension of the most significant bit is required when increasing the number of bits used to represent a signed integer. For example, the signed integer value of −128 is represented as the single byte 0x80, which in binary is <u>1</u>000 0000. Note that the leading bit is set. To represent −128 using two bytes, sign extend the most significant bit (1) to give <u>1111 1111 1</u>000 0000 (0xFF80). As another example, the signed integer value of 127 is represented as the single byte 0x7F, which in binary is <u>0</u>111 1111. Note the leading bit is clear. To represent 127 using two bytes, sign extend the most significant bit (0) to give <u>0000 0000 0</u>111 1111 (0x007F).

The variable length encoding is mandated for the VMTI LDS and all subordinate structures in order to minimize bandwidth overheads associated with the transmission of the data. Variable length values may be expanded by *0903* decoders into more convenient internal coding structures as required, for example UINT24 to UINT32.

A.5 Floating Point to Fixed Point Encoding and Decoding

As stated in §6.1, variable length encoding is mandated to minimize bandwidth overheads. While this precept can be applied straightforwardly to integer values, floating point values are not directly compatible with such a strategy. IEEE Standard 754 specifies encodings of 32, 64, and 128 bits, which precludes encoding "small" values (*e.g.*, zero) in, say, a single byte.

To mitigate this limitation, *0903* mandates transformation of floating point values into integer values, using appropriate scale and offset. The method to be used for encoding and decoding floating point values as integers, signed or unsigned, is defined here.[15]

[15] A number of MISB Standards prescribe methods for encoding and decoding floating point values as integers. While these are largely consistent, they differ in some details (e.g., the manner in which out-of-range values are indicated). To eliminate inconsistencies in future Standards, the MISB has created a standardized method, described in RP 1201 *Floating Point to Integer Mapping* [15]. However, to maintain backward compatibility with existing implementations, the method specified here shall be used for *0903*.

Let:

Mv = Measured (floating point) value
Ev = Encoded (integer) value

Mmin = Minimum measured value
Mmax = Maximum measured value

Lmax = Maximum length of the encoded value in bits, derived directly from the specified maximum number of bytes

Emin = Minimum encoded value
0, for unsigned integer values
-2^{Lmax-1}, for signed integer values

Emax = Maximum encoded value
$2^{Lmax} - 1$, for unsigned integer values
$2^{Lmax-1} - 1$, for signed integer values

Then:

To encode a floating point value as a signed integer[16]

$$Ev = (int)round\left(Emin + \frac{(Mv - Mmin)(Emax - Emin)}{(Mmax - Mmin)}\right)$$

If unsigned integers are used, then Emin = 0, simplifying the expression to

$$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right)$$

The precision of the encoded value is therefore

$$P = \frac{(Mmax - Mmin)}{(Emax - Emin)}$$

To decode an integer-encoded floating point value

$$Mv = \left(Mmin + \frac{(Ev - Emin)(Mmax - Mmin)}{(Emax - Emin)}\right)$$

Examples:

Signed integer –

[16] round() is presumed to be a built-in rounding function that correctly handles negative values to avoid a "rounding up" bias. For example, -0.6 should be rounded to -1.0, not 0. int() is presumed to be a signed integer "casting" operator. uint() is the analog for unsigned integer casting.

Mmin = -90 degrees
Mmax = 90 degrees

Lmax = 16 bits (2 bytes)

Emin = $-2^{Lmax-1} = -2^{15} = -32768$
Emax = $2^{Lmax-1} - 1 = 2^{15} - 1 = 32767$

If Mv = 0.001 degrees, then

$$Ev = (int)round\left((-32768) + \frac{(0.001 - (-90))\,(32767 - (-32768))}{(90 - (-90))}\right)$$

$$Ev = (int)round\left((-32768) + \frac{(90.001)\,(65535)}{(180)}\right)$$

$$Ev = (int)round(-0.1359) = 0$$

$$P = \frac{180}{65535} = 0.002747 \text{ degrees} = \sim 2.75 \text{ millidegrees}$$

If Ev = 0, then

$$Mv = \left((-90) + \frac{(0 - (-32768))\,(90 - (-90))}{(32767 - (-32768))}\right)$$

$$Mv = \left((-90) + \frac{(32768)\,(180)}{(65535)}\right)$$

$$Mv = 0.00137 \text{ degrees}$$

Unsigned integer –

Mmin = -90 degrees
Mmax = 90 degrees

Lmax = 16 bits (2 bytes)

Emin = 0
Emax = $2^{Lmax} - 1 = 2^{16} - 1 = 65535$

If Mv = 0.001 degrees, then

$$Ev = (int)round\left(\frac{(0.001 - (-90))\,65535}{(90 - (-90))}\right)$$

$$Ev = (int)round(32767.86) = 32768$$

$$P = \frac{180}{65535} = 0.002747 \text{ degrees} = \sim 2.75 \text{ millidegrees}$$

If Ev = 32768, then

$$Mv = \left(-90 + \frac{(32768 - 0)\,(90 - (-90))}{(65535 - 0)}\right)$$

$$Mv = 0.00137 \text{ degrees}$$

The choice of signed versus unsigned integer encoding should be governed by which is more likely to yield shorter encoded values (fewer required bytes). If values tend to be clustered about the middle of the measured range, then signed integer encoding will generally allow encoding in fewer bytes. If values tend to be clustered about the low end of the measured range, then unsigned integer encoding would be a better choice. As demonstrated in the two preceding examples, given a range from -90 to 90 degrees and a measured value of 0.001 degrees, signed integer encoding yields an encoded value of 0, whereas unsigned integer encoding yields an encoded value of 32768. The former can be represented by a single byte, but the latter requires two bytes.

Appendix B. KLV Complex Types [Normative]

KLV Complex Types used in *0903* are defined in this Appendix.

B.1 Series Type

The Series type is a one-dimensional array of data elements, all of the same type, as shown in Figure 6. The Series type is based on the SMPTE "Multiple" class, which is defined in draft *SMPTE 2003-200X Types Dictionary Structure*. In a Series, the element type is known, the size of each element can be determined prior to parsing[17], and the number of elements is determined as the data is parsed.

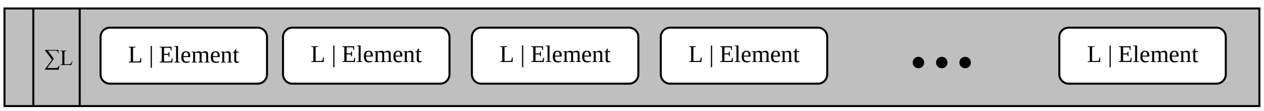


Figure 7: Series Type

B.2 Location Type

The Location type captures geopositioning data about a specific location on or in close proximity to the surface of the Earth. Location elements are structured as Defined-Length Truncation Packs. The elements of these packs fall into three groups, as shown in Figure 7. The first, and highest priority group, provides the geospatial location. This group includes the elements for Latitude, Longitude, and Height. The second, and medium priority group, provides standard deviations for the values in the first group. Standard deviation provides a measure of the variability of a coordinate value, an estimate of potential error. The third, and lowest priority group, provides correlation coefficients among the elements of the first group. The correlation coefficients provide a measure of systematic behavior, whether variation in the values of pairs of variables is "coupled" or random.[18]

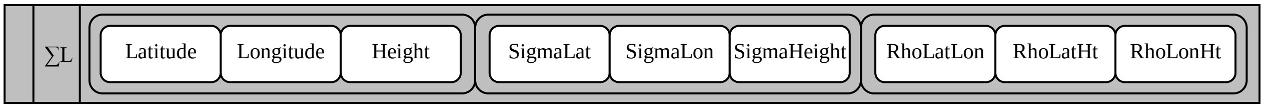
Standard deviations and correlation coefficients can be useful means to express confidence in the geocoordinates. For example, if standard deviations are small and correlation coefficients are near unity, then we can be "confident" the coordinate values are quite accurate. On the other hand, if standard deviations are large and correlation coefficients are near zero, we may expect potentially large, random errors.

Truncation shall be allowed only at a group boundary. No filler values shall be used for (unknown) higher priority elements.

---

[17] In general, the length of each element will be stated explicitly, using the Length-Value construct, as in a Variable-Length Pack. However, if the length of each element can be inferred, the Length field may be omitted, as in a Defined-Length Pack.

[18] Correlation is used because it is dimensionless and can be specified with a fixed range of values from -1.0 to +1.0, inclusive. Covariance, while it is a similar measure of "relatedness," is in units obtained by multiplying the units of two variables, and thus has values less well constrained.



Geospatial Coordinates Standard Deviations Correlation Coefficients

Figure 8: Location Type

B.2.1 *Latitude*

Data Element Definition			
Element Name	Latitude	Type	Units
		Float	Degrees
Valid Values	The set of real numbers from 90.0 to -90.0 inclusive		Precision
			~0.042 micro-degrees
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.28.00.00.00		Pack Tag
			NA
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	4 Bytes	Binary
Notes			
- Based on WGS84 ellipsoid.			
Conversion			

Given:
The maximum measured value is +90 degrees
The minimum measured value is -90 degrees.

If Mv is 43.0 degrees, then

$$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(43.0 - (-90))\,(2^{32} - 1)}{(90 - (-90))}\right) = 3173503612$$

If Ev is 3173503612, then

$$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(-90 + \frac{3173503612\,(90 - (-90))}{(2^{32} - 1)}\right) = 42.99999998 \text{ degrees}$$

Example Value	Example Encoded LDS Value
43.00 Degrees	[V] = [0xBD 27 D2 7C]

Latitude is the latitude of a point in degrees with respect to the WGS84 datum.

B.2.2 *Longitude*

<table>
<tr><th colspan="4">Data Element Definition</th></tr>
<tr><td rowspan="2">Element Name</td><td rowspan="2">Longitude</td><td>Type</td><td>Units</td></tr>
<tr><td>Float</td><td>Degrees</td></tr>
<tr><td rowspan="2">Valid Values</td><td colspan="2" rowspan="2">The set of real numbers from -180 to 180 inclusive</td><td>Precision</td></tr>
<tr><td>~0.083 micro-degrees</td></tr>
<tr><th colspan="4">KLV Encoding</th></tr>
<tr><td rowspan="2">Universal Label</td><td colspan="2" rowspan="2">06.0E.2B.34.01.01.01.01 / 0E.01.01.03.29.00.00.00</td><td>Pack Tag</td></tr>
<tr><td>NA</td></tr>
<tr><td>Volatility</td><td>Encoded Type</td><td>Length</td><td>Format</td></tr>
<tr><td>Dynamic</td><td>Unsigned Integer</td><td>4 Bytes</td><td>Binary</td></tr>
<tr><td colspan="4">Notes</td></tr>
<tr><td colspan="4">- Based on WGS84 ellipsoid.</td></tr>
<tr><td colspan="4">Conversion</td></tr>
<tr><td colspan="4">Given:
The maximum measured value is +180 degrees
The minimum measured value is -180 degrees.

If Mv is 110.0 degrees, then
$$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(110.0 - (-180))\,(2^{32} - 1)}{(180 - (-180))}\right) = 3459834765$$
If Ev is 3459834765, then
$$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(-180 + \frac{4551\,(180 - (-180))}{(2^{32} - 1)}\right) = 110.0 \text{ degrees}$$</td></tr>
<tr><td>Example Value</td><td colspan="3">Example Encoded LDS Value</td></tr>
<tr><td>110.00 Degrees</td><td colspan="3">[V] = [0xCE 38 E3 8D]</td></tr>
</table>

Longitude is the longitude of a point in degrees with respect to the WGS84 datum.

B.2.3 *Height*

Data Element Definition			
Element Name	Height	Type	Units
		Float	Meters
Valid Values	The set of real numbers from -900 to 19,000 inclusive		Precision
			~0.3 meters
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2A.00.00.00		Pack Tag
			NA
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	2 Bytes	Binary
Notes			
- Height with respect to the surface of the Earth. - Must be expressed as Height Above the Ellipsoid (HAE), using the WGS84 ellipsoid. - Map 0..(2^{16}-1) to -900..19000 meters. - Resolution: ~0.3 meters.			
Conversion			

Given:
The maximum measured value is +19,000 meters
The minimum measured value is -900 meters
 	
If Mv is 10,000 meters, then
$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(10000 - (-900))\,(2^{16} - 1)}{(19000 - (-900))}\right) = 35896$
 	
If Ev is 35896, then
$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(0 + \frac{35896\,(19000 - (-900))}{(2^{16} - 1)}\right) = 9999.98 \text{ meters}$			

Example Value	Example Encoded LDS Value		
10,000 meters	[K] [L] [V] = [0x0C][0x02][0x8C 38]		

Height is the height of a point in meters above the surface of the Earth, expressed as Height above the WGS84 Ellipsoid (HAE).

B.2.4 *Location Standard Deviation (Sigma)*

The Location Standard Deviation elements are:

- Sigma_Latitude = the standard deviation of a measure of Latitude
- Sigma_Longitude = the standard deviation of a measure of Longitude
- Sigma_Height = the standard deviation of a measure of Height

Each standard deviation is encoded the same way. The only difference between them is in the geospatial measurement referenced.

Data Element Definition			
Element Name	Sigma_Latitude Sigma_Longitude Sigma_Height	Type	Units
		Float	Meters
Valid Values	The set of real numbers ranging from 0.0 to 650.0 inclusive		Precision
			0.009918
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2D.00.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2D.01.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2D.02.00.00		LDS Tag
			NA
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	2 Bytes	Binary
Notes			
- Geolocation Standard Deviations must follow the following order in the element 1) Sigma_Latitude 2) Sigma_Longitude 3) Sigma_Height			
Conversion			

Given:
The maximum standard deviation is 650.0
The minimum standard deviation is 0.0

If Mv is 300 meters, then

$$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(300 - 0.0)\,(2^{16} - 1)}{(650.0 - 0.0)}\right) = 30247$$

If Ev is 30247, then

$$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(0.0 + \frac{30247\,(650.0 - 0.0)}{(2^{16} - 1)}\right) = 300.0\ meters$$

Example Value	Example Encoded LDS Value
Sigma_Latitude = 300 (0x76 27) Sigma_Longitude = 200 (0x4E C5) Sigma_Height = 100 (0x27 62)	[0x76 27 0x4E C5 0x27 67]

B.2.5 *Geolocation Correlation Coefficients*

The Correlation Coefficient elements are:

- Rho_Lat_Lon = the correlation coefficient between Latitude and Longitude.
- Rho_Lat_Ht = the correlation coefficient between Latitude and Height.
- Rho_Lon_Ht = the correlation coefficient between Longitude and Height.

Each correlation coefficient is encoded the same way. The only difference between them is in the geospatial measurement referenced.

Data Element Definition			
Element Name	Rho_Lat_Lon Rho_Lat_Ht Rho_Lon_Ht	Type: Float	Units: NA
Valid Values	The set of real numbers ranging from -1.0 to 1.0 inclusive		Precision: 3.0518×10^{-5}
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2E.00.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2E.01.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2E.02.00.00		LDS Tag: NA
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	2 Bytes	Binary
Notes			
- Geolocation correlation coefficients must follow the following order in the element 1) Rho_Lat_Lon 2) Rho_Lat_Ht 3) Rho_Lon_Ht			

Conversion

Given:
The maximum coefficient value is 1.0
The minimum coefficient value is –1.0

If Mv is 0.75, then

$$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(0.75 - (-1.0))\,(2^{16} - 1)}{(1.0 - (-1.0))}\right) = 57343$$

If Ev is 57343, then

$$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(-1.0 + \frac{57343\,(1.0 - (-1.0))}{(2^{16} - 1)}\right) = 0.749996$$

Example Value	Example Encoded LDS Value
Rho_Lat_Lon = 0.75 (0xDF FF) Rho_Lat_Ht = 0.50 (0xBF FF) Rho_Lon_Ht = 0.25 (0x9F FF)	[0xDF FF 0xBF FF 0x9F FF]

B.3 Velocity Type

The Velocity type captures data about the velocity of a moving object. Velocity elements are structured as Defined-Length Truncation packs. The elements of these packs fall into three groups, as shown in Figure 8. The first, and highest priority group, provides the measurements of velocity along three right-handed Cartesian coordinate axes, specifically, X (West to East), Y (South to North), and Z (vertical). The second, and medium priority group, provides standard deviations for the first group measurements. The third and lowest priority group provides three correlation coefficients for elements in the first group.

Truncation shall be allowed only at a group boundary. No filler values shall be used for (unknown) higher priority elements.

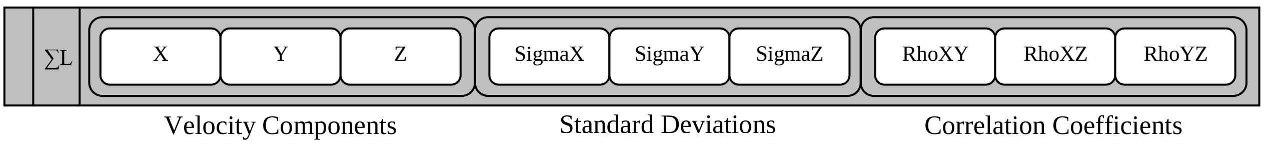


Figure 9: Velocity Type

B.3.1 *Velocity*

The Velocity elements are measured along three right-handed Cartesian coordinate axes, specifically, X (West to East), Y (South to North), and Z (vertical):

- X_Component = the value measured along the X axis
- Y_Component = the value measured along the Y axis
- Z_Component = the value measured along the Z axis

Each measured value is encoded the same way. The only difference between them is in the axis along which the measurement was made.

Data Element Definition			
Element Name	X_Component Y_Component Z_Component	Type Float	Units Meters/Second
Valid Values	The set of real numbers ranging from -900.0 to 900.0 inclusive		Precision 0.0275
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2B.00.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2B.01.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2B.02.00.00		LDS Tag NA
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	2 Bytes	Binary
Notes			
- Velocity Measured Values must appear in the following order in the group 1) X_Component 2) Y_Component 3) Z_Component *Editor: If objects traveling at less than ~7 meters/second will be frequently encountered, encoding these values as Signed, rather than Unsigned, Integers of Variable length would allow a single byte to be used for the "slow" movers. Dismounts would fall into this category.*			
Conversion			

Given:
The maximum measured value is 900.0
The minimum measured value is -900.0
 	
If Mv is 300 meters/second, then
 	
$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(300 - (-900.0))\,(2^{16} - 1)}{(900.0 - (-900.0))}\right) = 43690$
 	
If Ev is 43690, then			

$$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax} \right) = \left(-900.0 + \frac{43690\,(900.0 - (-900.0))}{(2^{16} - 1)} \right) = 300\ m/s$$	
Example Value	**Example Encoded LDS Value**
X_Component = 300 (0xAA AA) Y_Component = 200 (0x9C 71) Z_Component = 100 (0x8E 38)	[0xAA AA 0x9C 71 0x8E 38]

B.3.2 *Velocity Standard Deviation (Sigma)*

The Velocity Standard Deviation elements are:

- Sigma_X = the standard deviation of a value along the X axis
- Sigma_Y = the standard deviation of a value along the Y axis
- Sigma_Z = the standard deviation of a value along the Z axis

Each standard deviation is encoded the same way. The only difference between them is in the Velocity element referenced.

Data Element Definition			
Element Name	Sigma_X Sigma_Y Sigma_Z	Type: Float	Units: Meters/Second
Valid Values	The set of real numbers ranging from 0.0 to 650.0 inclusive		Precision: 0.009918
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2D.03.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2D.04.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2D.05.00.00		LDS Tag: NA
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	2 Bytes	Binary

Notes

- Velocity Standard Deviations must follow the following order in the element
  4) Sigma_X
  5) Sigma_Y
  6) Sigma_Z

Conversion

Given:
The maximum standard deviation is 650.0
The minimum standard deviation is 0.0

If Mv is 300 meters/second, then

$$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(300 - 0.0)\,(2^{16} - 1)}{(650.0 - 0.0)}\right) = 30247$$

If Ev is 30247, then

$$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(0.0 + \frac{30247\,(650.0 - 0.0)}{(2^{16} - 1)}\right) = 300.0\ m/s$$

Example Value	Example Encoded LDS Value
Sigma_X = 300 (0x76 27) Sigma_Y = 200 (0x4E C5) Sigma_Z = 100 (0x27 62)	[0x76 27 0x4E C5 0x27 67]

B.3.3 *Velocity Correlation Coefficients*

The Correlation Coefficients elements are:

- Rho_X_Y = the correlation coefficient between X Velocity and Y Velocity
- Rho_X_Z = the correlation coefficient between X Velocity and Z Velocity
- Rho_Y_Z = the correlation coefficient between Y Velocity Z Velocity

Each correlation coefficient is encoded the same way. The only difference between them is in Velocity element referenced.

Data Element Definition			
Element Name	Rho_X_Y Rho_X_Z Rho_Y_Z	Type	Units
		Float	NA
Valid Values	The set of real numbers ranging from -1.0 to 1.0 inclusive		Precision
			3.0518×10^{-5}
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2E.03.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2E.04.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2E.05.00.00		LDS Tag
			NA
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	2 Bytes	Binary
Notes			
- Velocity correlation coefficients must follow the following order in the element 4) Rho_X_Y 5) Rho_X_Z 6) Rho_Y_Z			
Conversion			

Given:
The maximum coefficient value is 1.0
The minimum coefficient value is –1.0
 	
If Mv is 0.75, then
$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(0.75 - (-1.0))\,(2^{16} - 1)}{(1.0 - (-1.0))}\right) = 57343$
 	
If Ev is 57343, then
$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(-1.0 + \frac{57343\,(1.0 - (-1.0))}{(2^{16} - 1)}\right) = 0.749996$			

Example Value		Example Encoded LDS Value	
Rho_X_Y = 0.75 (0xDF FF) Rho_X_Z = 0.50 (0xBF FF) Rho_Y_Z = 0.25 (0x9F FF)		[0xDF FF 0xBF FF 0x9F FF]	

B.4 Acceleration Type

The Acceleration type captures data about the acceleration of a moving object. Acceleration elements are structured as Defined-Length Truncation Packs. The elements of these packs fall into three groups, as shown in Figure 9. The first, and highest priority group, provide measurements of acceleration along three right-handed Cartesian coordinate axes. The second, and medium priority group, provides the standard deviations for the first group measurements. The third and lowest priority group provides the three correlation coefficients for the Acceleration elements.

Truncation shall be allowed only at a group boundary. No filler values shall be used for (unknown) higher priority elements.

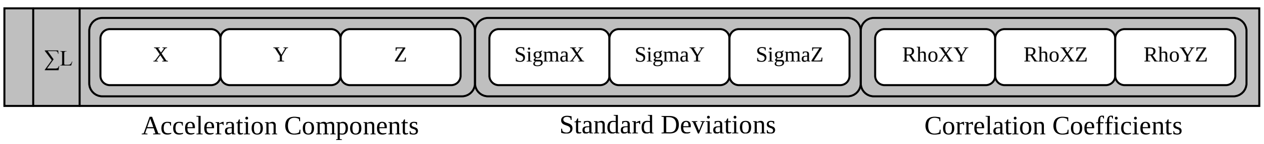


Figure 10: Acceleration Type

B.4.1 *Acceleration*

The Acceleration elements are measured along three right-handed Cartesian coordinate axes, specifically, X (West to East), Y (South to North), and Z (vertical):

- X_Component = the value measured along the X axis
- Y_Component = the value measured along the Y axis
- Z_Component = the value measured along the Z axis

Each measured value is encoded the same way. The only difference between them is the axis along which the measurement was made.

Data Element Definition			
Element Name	X_Component Y_Component Z_Component	Type: Float	Units: Meters/Second ^{2}
Valid Values	The set of real numbers ranging from -900.0 to 900.0 inclusive		Precision: 0.0275
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2C.00.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2C.01.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2C.02.00.00		LDS Tag: NA
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	2 Bytes	Binary

Notes

- Acceleration Measured Values must follow the following order in the element
  4) X_Component
  5) Y_Component
  6) Z_Component

Conversion

Given:
The maximum measured value is 900.0
The minimum measured value is -900.0

If Mv is 300 meters/second ^{2}, then

$$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(300 - (-900.0))\,(2^{16} - 1)}{(900.0 - (-900.0))}\right) = 43690$$

If Ev is 43690, then

$$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(-900.0 + \frac{43690\,(900.0 - (-900.0))}{(2^{16} - 1)}\right) = 300\ m/s^2$$

Example Value	Example Encoded LDS Value
X_Component = 300 (0xAA AA) Y_Component = 200 (0x9C 71)	[0xAA AA 0x9C 71 0x8E 38]

Z_Component = 100 (0x8E 38)	

B.4.2 *Acceleration Standard Deviation (Sigma)*

The Acceleration Standard Deviation elements are:

- Sigma_X = the standard deviation of a value along the X axis
- Sigma_Y = the standard deviation of a value along the Y axis
- Sigma_Z = the standard deviation of a value along the Z axis

Each standard deviation is encoded the same way. The only difference between them is in the Acceleration elements referenced.

Data Element Definition			
Element Name	Sigma_X Sigma_Y Sigma_Z	Type: Float	Units: Meters/Second^2
Valid Values	The set of real numbers ranging from 0.0 to 650.0 inclusive		Precision: 0.009918
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2D.06.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2D.07.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2D.08.00.00		LDS Tag: NA
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	2 Bytes	Binary

Notes

- Acceleration Standard Deviations must appear in the following order in the group
  7) Sigma_X
  8) Sigma_Y
  9) Sigma_Z

Conversion

Given:
The maximum standard deviation is 650.0
The minimum standard deviation is 0.0
If Mv is 300 meters/second ^{2}, then

$$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(300 - 0.0)\,(2^{16} - 1)}{(650.0 - 0.0)}\right) = 30247$$

If Ev is 30247, then

$$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(0.0 + \frac{30247\,(650.0 - 0.0)}{(2^{16} - 1)}\right) = 300.0\ m/s^2$$

Example Value	Example Encoded LDS Value
Sigma_X = 300 (0x76 27) Sigma_Y = 200 (0x4E C5)	[0x76 27 0x4E C5 0x27 67]

Sigma_Z = 100 (0x27 62)	

B.4.3 *Acceleration Correlation Coefficients*

The Acceleration Correlation Coefficients elements are:

- Rho_X_Y = the correlation coefficient between X Acceleration and Y Acceleration
- Rho_X_Z = the correlation coefficient between X Acceleration and Z Acceleration
- Rho_Y_Z = the correlation coefficient between Y Acceleration and Z Acceleration.

Each correlation coefficient is encoded the same way. The only difference between them is in the Acceleration elements referenced.

Data Element Definition			
Element Name	Rho_X_Y Rho_X_Z Rho_Y_Z	Type	Units
		Float	NA
Valid Values	The set of real numbers ranging from -1.0 to 1.0 inclusive		Precision
			3.0518×10^{-5}
KLV Encoding			
Universal Label	06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2E.06.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2E.07.00.00 06.0E.2B.34.01.01.01.01 / 0E.01.01.03.2E.08.00.00		LDS Tag
			NA
Volatility	Encoded Type	Length	Format
Dynamic	Unsigned Integer	2 Bytes	Binary
Notes			
- Acceleration correlation coefficients must follow the following order in the element 7) Rho_X_Y 8) Rho_X_Z 9) Rho_Y_Z			
Conversion			

Given:
The maximum coefficient value is 1.0
The minimum coefficient value is –1.0
 	
If Mv is 0.75, then
$Ev = (uint)round\left(\frac{(Mv - Mmin)\,Emax}{(Mmax - Mmin)}\right) = (uint)round\left(\frac{(0.75 - (-1.0))\,(2^{16} - 1)}{(1.0 - (-1.0))}\right) = 57343$
 	
If Ev is 57343, then
$Mv = \left(Mmin + \frac{Ev\,(Mmax - Mmin)}{Emax}\right) = \left(-1.0 + \frac{57343\,(1.0 - (-1.0))}{(2^{16} - 1)}\right) = 0.749996$			

Example Value	Example Encoded LDS Value
Rho_X_Y = 0.75 (0xDF FF)	[0xDF FF 0xBF FF 0x9F FF]

Rho_X_Z = 0.50 (0xBF FF) Rho_Y_Z = 0.25 (0x9F FF)	

B.5 Boundary Type

A Boundary is a Series of vertices which define an area or volume enclosing a region of interest. Each vertex is an element of type Location. There can be up to 65,635 vertices in a Boundary. The vertices should be ordered so that looking toward Earth center, they spiral in a clockwise direction from lowest elevation to highest. An example of a shape and expected ordering of vertices is depicted in .

Data Element Definition			
Element Name	Boundary	Type	Units
		Array	NA
Valid Values	Any valid structure of type Location		Precision
			NA
KLV Encoding			
Universal Label	06.0E.2B.34.02.04.01.01 / 0E.01.03.03.17.00.00.00		LDS Tag
			NA
Volatility	Encoded Type	Length	Format
Dynamic	Series of Location data elements	Variable	Binary
Notes			
- None			
Conversion			
NA			
Example Value		Example Encoded LDS Value	
NA		NA	

At least two vertices must be specified.

When only two vertices are provided, they should be interpreted as opposite corners of a simple, planar bounding box, aligned with the Latitude-Longitude grid.

Vertex Location elements are not required all to be the same length. Correlation coefficient and standard deviation groups may be omitted. (Integer encoding may result in varying lengths, as well.)

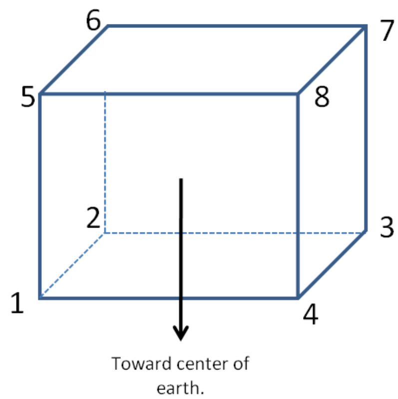
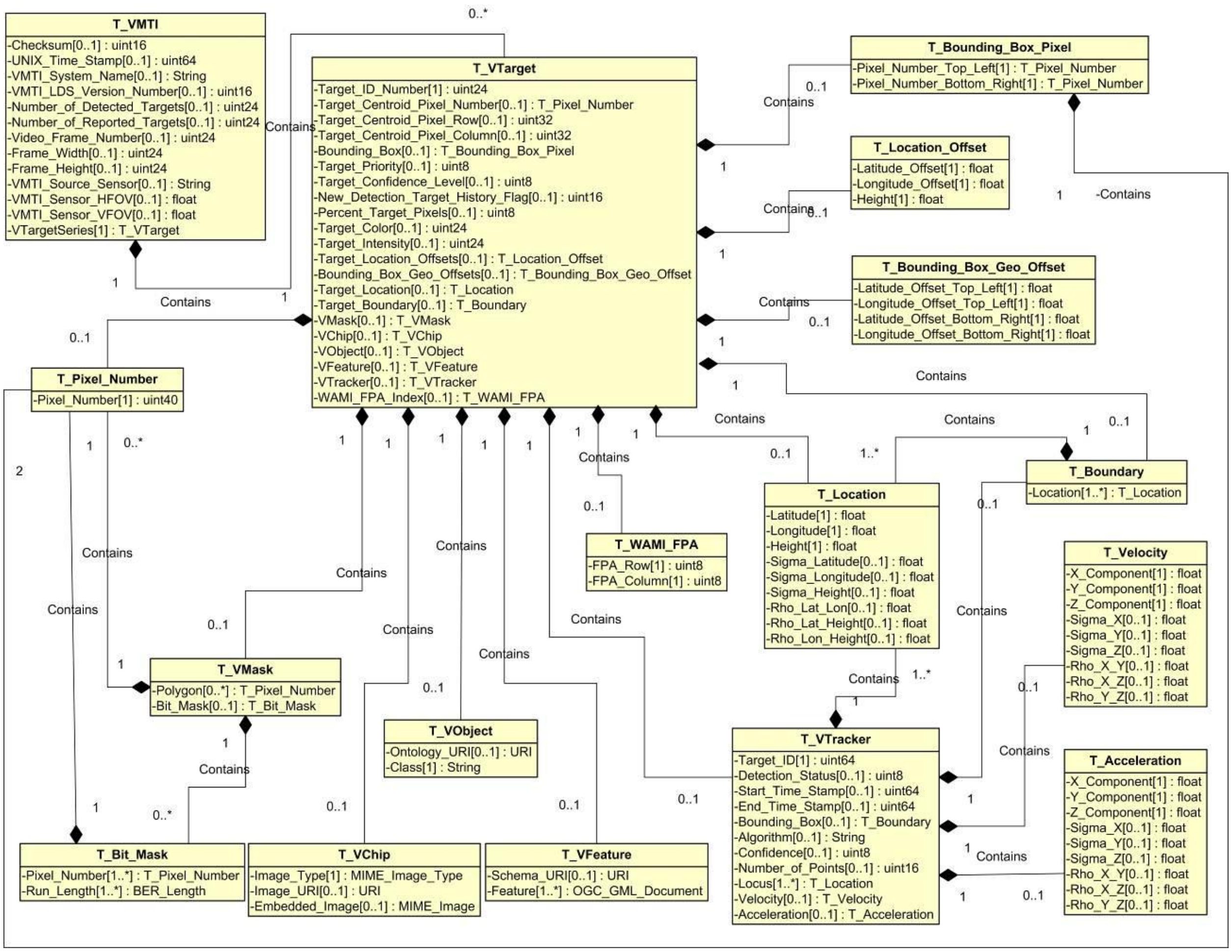
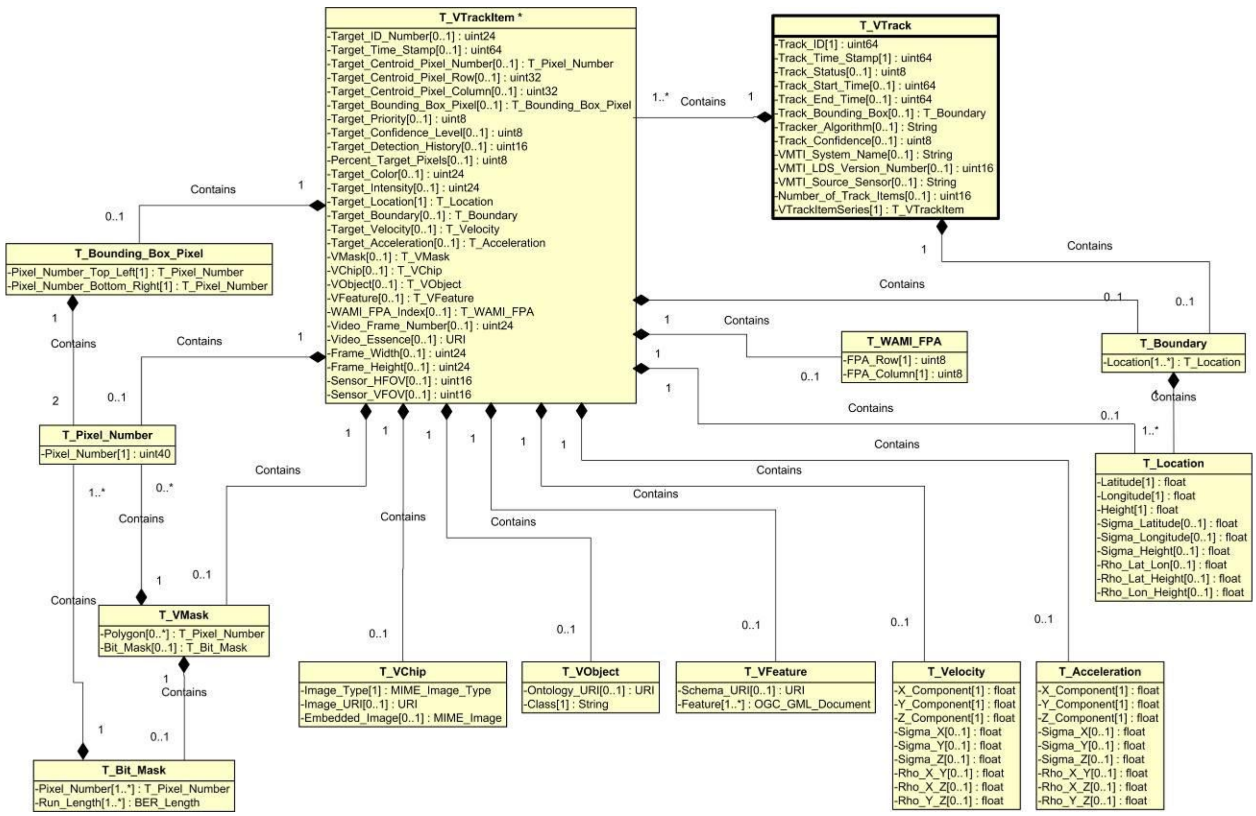


Figure 11. Example of a simple cubic boundary. Vertices 1-4 are coplanar, as are vertices 5-8. The vertices would be encoded in the following order: [1, 2, 3, 4, 5, 6, 7, 8].

Appendix C. VMTI LDS UML Data Model [Informative]



Appendix D. VTrack LDS UML Data Model [Informative]



Appendix E. Correspondence between VTrack and STANAG 4676 [Informative]

As of this date, STANAG 4676 Edition 1 (DRAFT) [21] is not a ratified NATO standard (still classified as a Study). However, ratification is anticipated in the near future.

Although STANAG 4676 Edition 1 does not yet address many of the object properties that can be derived from motion imagery, such as, shape, spectral characteristics, feature characteristics, and fine-grained motion behavior, extension of the STANAG to accommodate this information ought to be straightforward. The MISB is working with the STANAG 4676 technical support team to address necessary additions and to harmonize STANAG 4676 with MISB Standard 0903.

To illustrate anticipated compatibility of STANAG 4676 and MISB Standard 0903, this Appendix uses the UML data model of STANAG 4676, defined in the STANAG 4676 Implementation Guide AEDP-12 [22], to describe the correspondence between VTrack data elements and those defined in STANAG 4676 Edition 1. The mapping is shown in Figure 12. Classes of the STANAG 4676 data model appear in the color beige/tan. VTrack classes appear in green.

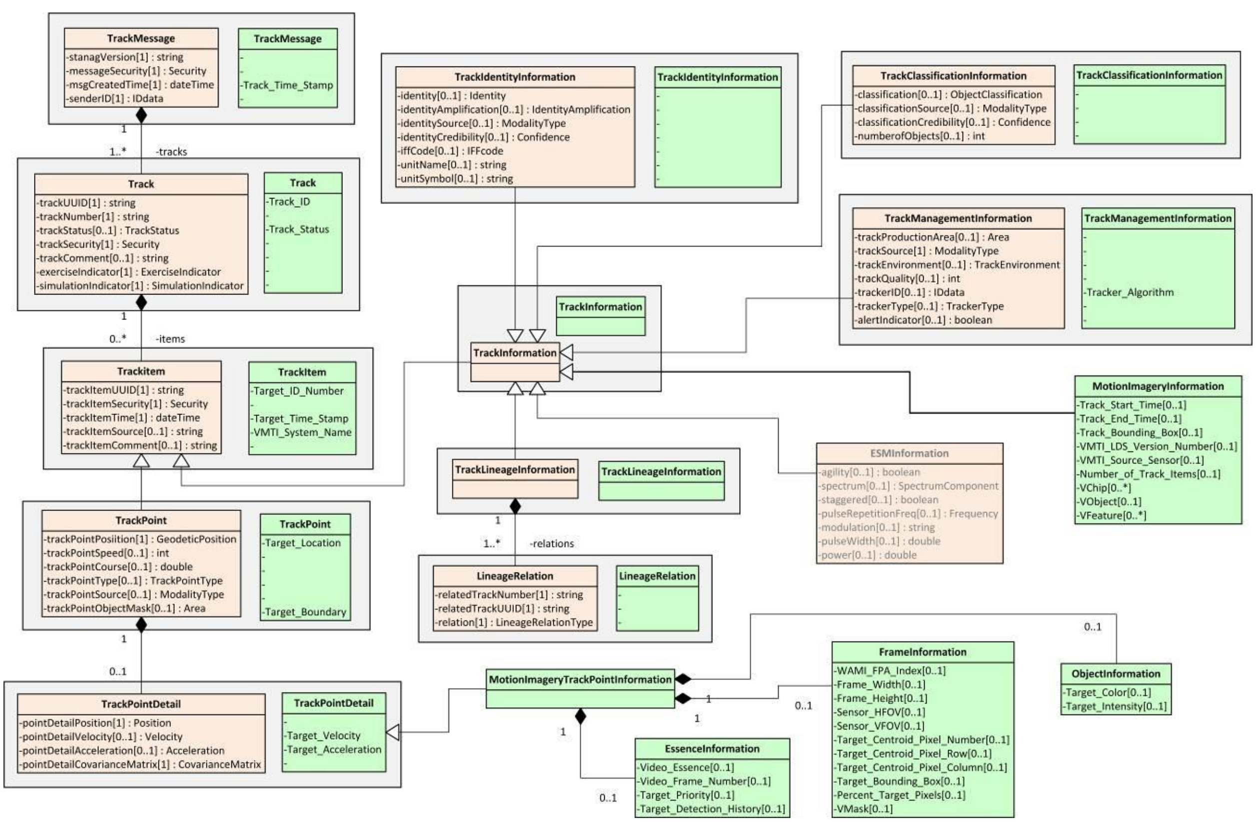


Figure 12. A Mapping between the STANAG 4676 Edition 1 UML Data Model and the VTrack Data Model

The fundamental class of the STANAG 4676 data model is a TrackMessage, which contains information describing one or more Tracks. Track information may include TrackItems, which contain either track point information (TrackPoint) or amplifying "non-kinematic" information (TrackInformation) pertinent to a track (TrackIdentityInformation, TrackClassificationInformation, TrackManagementInformation, or TrackLineageInformation).

In the UML diagram, each STANAG 4676 class is paired with a green class box, which indicates the VTrack data elements ("attributes"), if any, that correspond to those in the STANAG 4676 class. Consider the TrackMessage class, for which the VTrack attribute Track_Time_Stamp corresponds to the TrackMessage attribute msgCreatedTime. No other matches are indicated.

When no corresponding VTrack data element is shown, the STANAG 4676 attribute is generally assumed to come from a metadata class other than VTrack. For example, security metadata would be taken from MISB Standard 0102 *Security Metadata Universal and Local Sets for Digital Motion Imagery* (NATO STANAG 4609 STD N0102).

In a few cases, there is no MISB standard that contains a matching attribute. If the STANAG 4676 attribute is optional, denoted by multiplicity "[0..1]" (*e.g.*, TrackIdentityInformation::iffCode), it can simply be left unspecified. However, if the attribute is mandatory, denoted by multiplicity "[1]" (*e.g.*, TrackMessage::senderID), an appropriate MISB counterpart will have to be defined.[19]

Since VTrack metadata is encoded using KLV but STANAG 4676 currently recognizes only XML as a data encoding method, conversion from KLV to XML will be required except for character string values (e.g., VMTI_System_Name).[20]

VTrack contains many attributes that can be derived from motion imagery but not from most other sources of track information. The data model defined by AEDP-12 is augmented, as shown in Figure 12, to include these additional data elements. A specialization of TrackInformation includes MotionImageryInformation, which contains attributes VChip, VObject, and VFeature (and others).[21] The specialization MotionImageryTrackPointInformation of TrackPointDetail includes EssenceInformation, FrameInformation, and ObjectInformation as subclasses. For clarity, these additions are shown in the "steamlined" data model of Figure 13.

It seems reasonable to assume that a complete harmonization of STANAG 4676 and Standard 0903 can be achieved, at least at the data model level, even if closure on encoding details remains elusive.

[19] The MISB asserts that some mandatory STANAG 4676 attributes should instead be optional. Specific examples include Track::trackNumber and LineageRelation::relatedTrackNumber, which are alternate expressions of mandatory attributes Track::trackUUID and LineageRelation::relatedTrackUUID, respectively; Track::exerciseIndicator and Track::simulationIndicator, which should be present in the affirmative only when relevant; and TrackPointDetail::pointDetailPosition, which is an alternate expression of mandatory attribute TrackPoint::trackPointPosition.

[20] Recall that MISB standards mandate the use of KLV encoding to maximize bit efficiency for limited bandwidth environments. Because of its "verbosity", XML encoding is generally unsuited for such environments. Use of an alternate, more efficient data encoding method (such as, KLV) is being considered for STANAG 4676.

[21] The UML data model defined in AEDP-12 contains a "stub" for the MotionImageryInformation subclass.

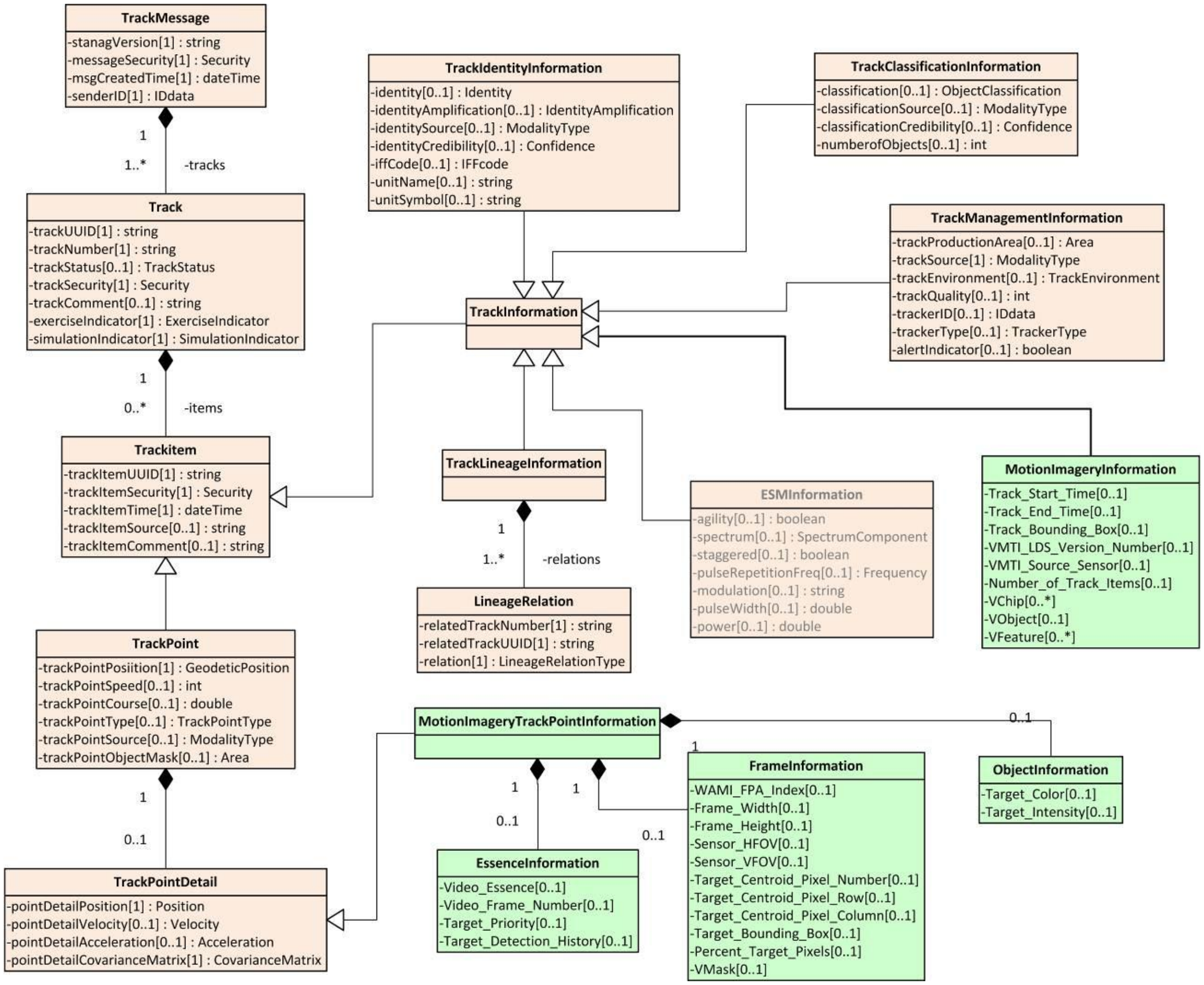


Figure 13. The STANAG 4676 Edition 1 UML Data Model with Motion Imagery VTrack Metadata Additions

Appendix F. Operational Considerations [Informative]

F.1 Bandwidth

Bandwidth management is an important issue affecting all systems. It is of particular importance for VMTI (and tracking) systems which produce a large amount of dynamic data at frame rates. Thus careful consideration has been given to efficient representation of the data.

In its simplest implementation, the VMTI LDS may send thousands of targets to downstream systems by providing a Target ID Number and the Target Centroid Pixel Number within the video frame for each target. At the other end of the scale, the VMTI LDS has scope to include multiple features about each target, image chips of the target, tracking information about the target, and numerous descriptive elements. The bandwidth overhead required to include all this information is very large – especially at 60 frames per second or higher.

Video sensors with frame rates of up to 60 frames per second (fps) are used operationally, and faster systems are expected in the future. It follows that VMTI rates will follow the same trend. In order to conserve bandwidth, a distinction is made between dynamic data, periodic data, and static data. Dynamic data changes continuously and is only valid at a specific instant in time. Periodic data changes periodically and is valid for a period of time. Static data rarely, if ever, changes within a single mission. The expected refresh rates for these three classes of data are listed in .

VMTI data does not have to be delivered at the frame rate of the motion imagery. Data that rarely changes should only be delivered often enough to assure that it is included in any clip extracted from the motion imagery stream. Dynamic data should be delivered at a rate that is appropriate to the granularity of the intelligence provided. For example, for a target moving at 3 meters per second, a rate of 60 updates a second provides very little value over a rate of 20 updates per second. In that case, while the motion imagery frame rate might be 60 fps, the VMTI update rate need only be 20 “fps”.

The VMTI LDS includes a large number of elements that may not be available from on-board VMTI processors. It is expected that downstream processes will contribute “value add” elements to a basic VMTI LDS stream. Calculations of bandwidth requirements should take into account where in the VMTI workflow the data is being added to the stream and the available bandwidth at that stage.

Table 17: Volatility Refresh Rates

Volatility	Refresh Rate
Dynamic	Every VMTI frame
Periodic	Every two (2) seconds or first available time after the value changes
Static	Every two (2) seconds or as available from an external source

Bandwidth implications and bandwidth management must be considered in the design of systems that generate VMTI LDS data. Reduction of metadata to a minimal effective configuration is recommended. Although the LDS provides scope for many elements, it is undesirable to populate elements just because Tags exist to support the data.

The VMTI LDS provides the option to include a time stamp. It is expected that the VMTI LDS time stamp will be used to synchronize multiplexing with the *0601* stream but not be explicitly included in the stream. However, if both time stamps are present, the VMTI time stamp takes precedence.

With respect to the error detection checksum, although the VMTI LDS mandates the use of a checksum, if the LDS is multiplexed with a *0601* stream, the VMTI LDS checksum should be omitted in deference to the new, recalculated *0601* checksum. This is in the interest of conserving bandwidth.

Every effort has been made not to repeat metadata elements that are found in other LDSs, particularly *0601*. However, some metadata elements (VMTI Source Sensor, VMTI HFOV and VFOV) in the VMTI LDS are comparable to those in *0601* LDS (Image Source Sensor, HFOV, and VFOV. They become relevant when the VMTI process is run on different video from that described by and/or included with the *0601* data. Consider, for example, two bore-sighted sensors, where *0601* metadata describes the video essence from one of the *0601* sensors, but *0903* metadata describes VMTI detections from the video essence of the other sensor.

Each VMTI process from a given sensor requires its own individual VMTI LDS. That is, a VMTI LDS shall not contain a mixture of moving targets detected by different sensors.

F.2 Co-located Bore-sighted Sensors

A system with multiple bore-sighted imagers within a single turret may send a video stream from one camera with a given field of view, synchronously with VMTI hits from other sensors with different fields of view. For example, a system containing a narrow field of view (FOV) electro-optic (EO) sensor (EON), a wide FOV EO sensor (EOW), and an infrared (IR) sensor may be transmitting the "video–in-control" stream from the EON sensor and simultaneously include VMTI hits from both the EOW sensor and the IR sensor. This situation is supported by providing a separate VMTI packet stream, consisting of distinct *0903* LDSs (within *0601* LDSs) for the IR and EOW cameras, multiplexed with the transport stream containing the EON video (with its own *0601* LDS).

VMTI packet streams shall be distinguished within the multiplexed transport stream by the VMTI Source Sensor field plus the VMTI Sensor Horizontal Field of View (HFOV) field. (See §7.10.10 and §7.10.11 for definitions.)

Sophisticated VMTI systems may use the same Target ID Number to identify a common target detected by different sensors and retain the use of same Target ID Number temporally (that is, from one detection to another). Also, downstream processes (*e.g.*, trackers and fusion systems) may reassign Target ID Numbers to identify a common target.

F.3 Independent Sensors

Each independent system requires a separate *0601* stream. The extra elements required in the VMTI LDS to support multiple non-bore-sighted sensors would disproportionally increase bandwidth requirements for inclusion within a single *0601* packet. Given that *0601* does not support such cases anyway, the most appropriate solution is to generate individual VMTI LDSs and *0601* streams for these sensors.

F.4 Wide Area Motion Imagery

Wide Area Motion Imagery (WAMI) systems present a problem not normally encountered with "traditional" airborne video sensors. WAMI systems can detect thousands of simultaneous moving objects over several square kilometers of area, but the volume of motion imagery collected is far beyond the downlink capabilities of the communications channel. A standalone stream of VMTI or Track information (transmitted independent of any video essence) could describe all of the detected moving objects and be used to cue analysts as to which objects to monitor actively and which to leave, say, to automated processes. This information can be used to task "spotlights" that specify regions of particular interest to the analyst, for which motion imagery is actually downloaded, reducing potential bandwidth requirements. These spotlights in turn can carry VMTI or Track data for the moving objects within the scope of the spotlight imagery.

VMTI and Track data for the spotlight streams will be similar to the independent sensors paradigm (§F.3) in their packaging and processing. However, VMTI and Track data used for cueing takes a different approach. In this case we cannot assume the existence of an associated motion imagery stream. Cueing VMTI or Track must therefore be self-contained, with no dependencies on other data streams.