

NOT MEASUREMENT  
SENSITIVE

MIL-STD-3014  
w/CHANGE 1  
5 May 2005  
SUPERSEDING  
MIL-STD-3014  
20 February 2004

**DEPARTMENT OF DEFENSE  
INTERFACE STANDARD  
FOR  
MISSION DATA EXCHANGE FORMAT**



AMSC: N/A

AREA: MCCR

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

**MIL-STD-3014****FOREWORD**

1. This standard is approved for use by all Departments and Agencies of the Department of Defense.
2. Prior to the introduction of this standard, mission data exchange files transferred between mission planning software and weapon system computers were typically developed independently for each system, and each mission file format was individually integrated on transport media. Transfer of these independent mission plan formats via multiple data transmission media and protocols was not typically performed, because sponsoring and funding integration with multiple media and protocols was too big a challenge for each individual weapon system to accept. This was especially true considering the length of time required for integration of weapon-unique data into international protocols such as TADIL-J, used by Link-16, compared to the product development cycle for weapon systems. As a result, these systems have typically developed only a single path by which to transfer their mission plans. In the past, this was accomplished using a hardware memory unit that is hand-carried by the pilot from the mission planning station to a launch platform. Historically, mission planning for weapon systems and their launch platforms has focused on deliberate, planned operations that include hours or days of planning time before weapon launch, and a dedicated platform mission in support of that launch.
3. An increased DoD focus on prosecution of time-sensitive targets (TSTs) and a DoD transition to network-centric operations (NCO) have demanded shorter and shorter times from discovery of a target by a sensor asset, through the targeting, command authorization, and weaponeering processes, to the assignment of an attack mission and mission details to a specific shooter and weapon. The timeline of the TST mission, combined with the NCO concept of operations (CONOPS) demands increasing flexibility, both in the selection of weapons to prosecute specific targets (based largely on minimizing time to weapon impact) and in delivering the mission details to the most available weapon system, even though it may already be in captive or free flight on another mission.
4. The functional goal of this standard is to establish a common mission data transfer format for weapon systems. Within this format, every mission data source node or workstation, weapon platform, digital communication link, and weapon system can build, modify, read, interpret, pass, or carry out a mission plan, using a common lexicon and a common interface. The mission-level goal is to improve the speed and flexibility of creating and delivering mission plans for weapon systems, in order to enhance their value of those weapon systems to commanders and warfighters. The economic goal of this standard is to reduce the cost, complexity, and risk of weapon integration into platforms and communications paths, by permitting a single file format for digital mission data to be used, regardless of source, content, or transmission path, by all weapon systems compliant with this standard.
5. This document is reviewed and maintained on a regular basis. The registry appendices are updated quarterly. The current registry, calendar for updates, and forms and address for submittal of requests for additions can be found online at <http://MIL-STD-3014.NAVY.MIL>.
6. Comments, suggestions, or questions on this document should be addressed to (ASC/ENOS, 2530 Loop Road West, Wright-Patterson AFB OH 45433-7101) or emailed to ([Engineering.Standards@wpafb.af.mil](mailto:Engineering.Standards@wpafb.af.mil)). Since contact information can change, you may want to verify the currency of this address information using the ASSIST Online database at <http://assist.daps.dla.mil/online/start/> or <http://assist.daps.dla.mil/quicksearch/>.

**MIL-STD-3014****CONTENTS**

<b>Paragraph</b>	<b>Page</b>
<b>1. SCOPE.....</b>	<b>1</b>
1.1 Scope.....	1
1.2 Purpose.....	1
1.3 Application.....	1
<b>2. APPLICABLE DOCUMENTS .....</b>	<b>2</b>
2.1 General.....	2
2.2 Government documents.....	2
2.2.1 Specifications, standards, and handbooks.....	2
2.3 Non-Government publications.....	2
2.4 Order of precedence.....	2
<b>3. DEFINITIONS.....</b>	<b>3</b>
3.1 Definitions.....	3
3.1.1 Assert/negate.....	3
3.1.2 Byte.....	3
3.1.3 Class code.....	3
3.1.4 Data element.....	3
3.1.5 Module editor.....	3
3.1.6 Module header.....	3
3.1.7 Organizational entity.....	4
3.1.8 Registry.....	4
3.1.9 Subordinate/superior module.....	4
3.1.10 Word.....	4
3.1.11 Syntax diagrams.....	4
3.2 Acronyms and abbreviations.....	5
<b>4. GENERAL REQUIREMENTS .....</b>	<b>6</b>
4.1 Data files.....	6
<b>5. DETAILED REQUIREMENTS.....</b>	<b>6</b>
5.1 Organization of data in files.....	6
5.2 Order of module data elements.....	7
5.2.1 Use of nonregistered values in header fields and data elements.....	8
5.2.2 Use of elements in program-unique modules.....	8
5.2.3 Use and interpretation of generic element types for specific purposes.....	8
5.2.4 Order of module headers.....	8
5.3 Definition of mandatory header fields.....	9
5.3.1 Module CLASS CODE field.....	10
5.3.2 MODULE SIZE field.....	10
5.3.3 HEADER CONTENT field.....	11

**MIL-STD-3014****CONTENTS**

<b>Paragraph</b>	<b>Page</b>
5.3.4 USER field.....	11
5.3.5 ELEMENT COUNT field.....	12
5.3.6 ELEMENT LIST field.....	13
5.4 Definition of optional header fields.....	13
5.4.1 Creator fields.....	16
5.4.2 CODE-related fields.....	17
5.4.3 Authorizer-related fields.....	18
5.4.4 Source fields.....	19
5.4.5 NOTE-related fields.....	20
5.5 Order of primitive data elements.....	20
5.6 Order of concatenated data elements.....	21
<b>6. NOTES .....</b>	<b>22</b>
6.1 Intended use.....	22
6.1.1 Implementation.....	22
6.1.2 Use of optional fields.....	23
6.2 Acquisition requirements.....	24
6.3 Tailoring guidance.....	24
6.4 Keyword list.....	24
6.5 Change notations.....	25

**TABLES**

TABLE I. Mandatory header field sequence.....	9
TABLE II. HEADER CONTENT field format.....	11
TABLE III. Optional header field sequence.....	14

**FIGURES**

FIGURE 1. Interpreting syntax diagrams.....	4
FIGURE 2. Data file organization.....	6
FIGURE 3. Order of bits in class codes.....	7
FIGURE 4. Order of a module's header and data.....	7
FIGURE 5. Order of mandatory and optional header fields.....	8
FIGURE 6. Order of mandatory header fields in syntax diagram format.....	9
FIGURE 7. Order of mandatory header fields in field size format.....	10
FIGURE 8. Structure of element list.....	12

**MIL-STD-3014****CONTENTS**

<b>Paragraph</b>	<b>Page</b>
FIGURE 9. Order of entries in element list.....	13
FIGURE 10. Order of optional header fields in syntax diagram format.....	15
FIGURE 11. Order of optional header fields (when fully populated) in field size format.....	15
FIGURE 12. Order of CREATE fields.....	16
FIGURE 13. Order of CODE fields.....	17
FIGURE 14. Order of AUTH fields.....	18
FIGURE 15. Order of SOURCE fields.....	19
FIGURE 16. Order of NOTE fields.....	20
FIGURE 17. Order of primitive data elements.....	20
FIGURE 18. Order of concatenated data elements.....	21

**MIL-STD-3014****1. SCOPE****1.1 Scope.**

This standard defines a format for digital data files used for mission-level programming of weapon systems. The term format applies to the organization of data information in the file and the definition of approved data types.

- a. This standard defines, first, a hierarchical or “nested” file architecture, which allows a broad and open-ended variety of data types to be combined and used for a variety of purposes.
- b. Second, this standard is supported by online registries that catalogue and define standard formats to be used for some common-use data elements, for which a common data format is of benefit.
- c. Third, this standard is supported by online registries that catalogue and identify unique data types required by specific acquisition programs. As a unique data type registry, this standard supports the use of data types whose definitions are proprietary, classified, or have very limited application. The registry serves to identify such data types without the need to define them, and prevents indeterminacy and confusion caused when different applications choose the same identifier for different types of data.
- d. Finally, this standard is supported by online registries that identify organizations and acquisition systems that produce, transfer, or act upon mission data files.

This standard does not establish requirements for content of mission data files, accuracy of data, or its applicability for any specific weapon system. Those characteristics are the responsibility of the organizational entity that generates the data and formats it for use in a weapon system.

**1.2 Purpose.**

The purpose of this standard is to provide a common lexicon with which all combat systems can exchange digital mission data for weapon systems, in order to deliver the desired effects to the enemy as rapidly, economically, effectively, and flexibly as possible.

**1.3 Application.**

This standard defines the file format for the communication of mission data for weapon systems. For this standard, weapon systems are defined as weapons that function autonomously for at least part of their mission, based on computers operating on received digital mission data. Such systems include those whose plan can be altered or updated in midflight, and those whose function includes operator interventions, such as those with man-in-the-loop terminal guidance.

This standard does not apply to direct weapon guidance commands supplied by an operator in a form other than digital data, such as reflected laser or radar energy, and infrared or radar frequency command-to-line-of-sight beam-steering signals.

**MIL-STD-3014****2. APPLICABLE DOCUMENTS****2.1 General.**

The documents listed in this section are specified in sections 3, 4, and 5 of this standard. This section does not include documents cited in other sections of this standard or recommended for additional information or as examples. While every effort has been made to ensure the completeness of this list, document users are cautioned that they must meet all specified requirements documents cited in sections 3, 4, and 5 of this standard, whether or not they are listed.

**2.2 Government documents.****2.2.1 Specifications, standards, and handbooks.**

The following specifications, standards, and handbooks form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those listed in the issue of the Department of Defense Index of Specifications and Standards (DoDISS) and supplement thereto, cited in the solicitation (see 6.2).

**STANDARDS**Department of Defense

MIL-STD-1760                    Department of Defense Interface Military Standard for Aircraft/Store Electrical Interconnection System

(Unless otherwise indicated, copies of the above specifications, standards, and handbooks are available from the Standardization Document Order Desk, 700 Robbins Avenue, Building 4D, Philadelphia, PA 19111-5094 or online via ASSIST at

<http://assist.daps.dla.mil/online/start/> or <http://assist.daps.dla.mil/quicksearch/>.)

**2.3 Non-Government publications.**

The following document forms a part of this standard to the extent specified herein. Unless otherwise specified, the issues of these documents are those cited in the solicitation or contract.

American National Standards Institute

ASCII                    American Standard Code for Information Interchange

(Copies of ASCII Codes are available at <http://www.asciitable.com/>.

**2.4 Order of precedence.**

In the event of a conflict between the text of this document and the references cited herein, the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

**MIL-STD-3014****3. DEFINITIONS****3.1 Definitions.**

Definitions applicable to this standard are as follows.

**3.1.1 Assert/negate.**

When describing digital states, the term “assert” is used to indicate the setting of a bit, flag, or other digital element to a bit state of logic “1”. The term “negate” is used to indicate the setting of a bit, flag, or other digital element to a bit state of logic “0”.

**3.1.2 Byte.**

An ordered sequence of eight bits, when used to describe binary information in this standard.

**3.1.3 Class code.**

The unique, two-byte identifier, assigned by the managing office, that uniquely identifies a registered data element.

**3.1.4 Data element.**

A digital instruction or piece of data, or set of instructions or data, provided to an autonomous system to preprogram subsequent autonomous behavior. A data element can be one of three types: primitive, concatenated, or module.

**3.1.4.1 Primitive data element.**

A primitive data element conveys a single piece of information (e.g., a latitude). It has a fixed size and format, and a defined class code.

**3.1.4.2 Concatenated data element.**

A concatenated data element is a specific, ordered sequence of primitive data elements (e.g., an ordered sequence of latitude followed by longitude) as defined in its registry entry. It has a fixed size and format and a defined class code.

**3.1.4.3 Modular data element.**

A modular data element contains one or more subordinate data elements, which can be primitive, concatenated, or other modules. Modules may optionally have a registered class code. Modules do not have a fixed size. They contain a module header that defines the length and content of the module. This standard places no limit to the length or the number of subordinate data elements in a module.

**3.1.5 Module editor.**

Software functionality that creates or edits modules of mission data files that comply with this standard.

**3.1.6 Module header.**

That portion of the information in a module that describes the content, nature, and intended use of that module distinct from the actual mission data contained in the data elements.

**MIL-STD-3014****3.1.7 Organizational entity.**

An organizational entity is a military unit at any level that is registered for identification purposes as creating or receiving files that comply with this standard.

**3.1.8 Registry.**

An official, controlled listing of approved values for a field or data type. Registries are found online at <HTTP://MIL-STD-3014.NAVY.MIL> and are frequently updated based on submittals of user requests via online forms linked to the registries.

**3.1.9 Subordinate/superior module.**

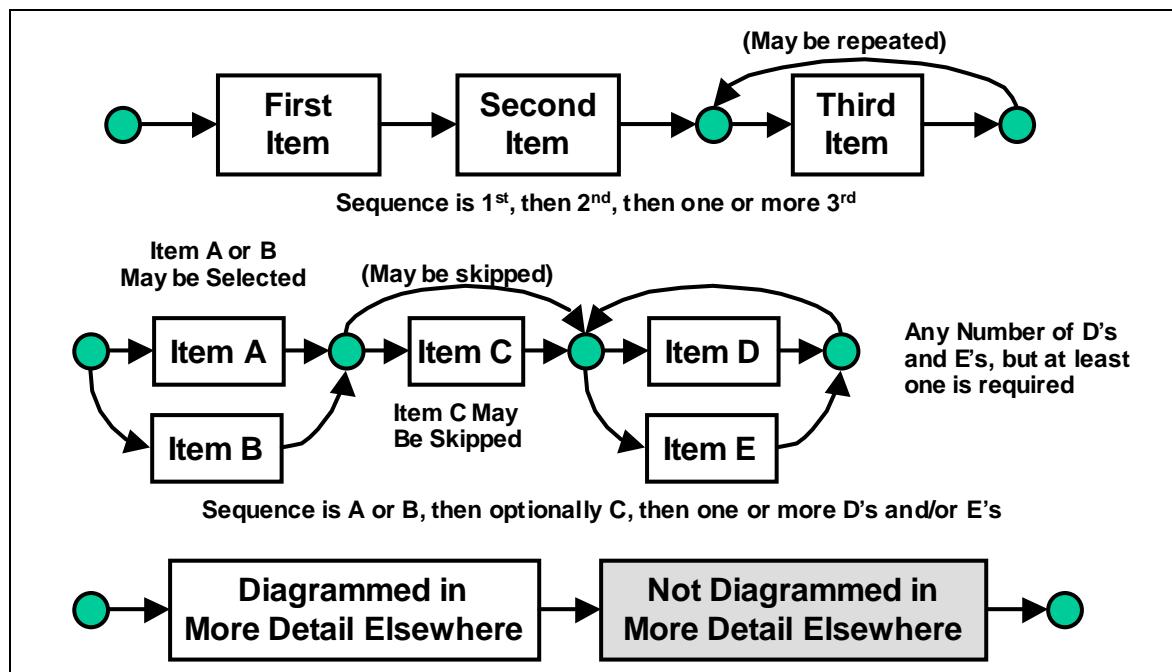
“Subordinate” and “superior” describe relationships between modules. A subordinate module is a module that is contained as a data element in another module. A superior module is a module that contains one or more subordinate modules as data elements.

**3.1.10 Word.**

An ordered sequence of sixteen bits, when used to describe binary information in this standard.

**3.1.11 Syntax diagrams.**

Figures in this standard that are used to show required and optional ordering or sequencing of fields and data elements. See figure 1 for examples.



**FIGURE 1. Interpreting syntax diagrams.**

## MIL-STD-3014

### 3.2 Acronyms and abbreviations.

The following acronyms and abbreviations are applicable to this standard:

ASCII	American Standard Code for Information Interchange
DoDISS	Department of Defense Index of Specifications and Standards
NATO	North Atlantic Treaty Organization
NCO	network-centric operations
PKI	public key interface
TST	time-sensitive target

#### **4. GENERAL REQUIREMENTS**

## 4.1 Data files.

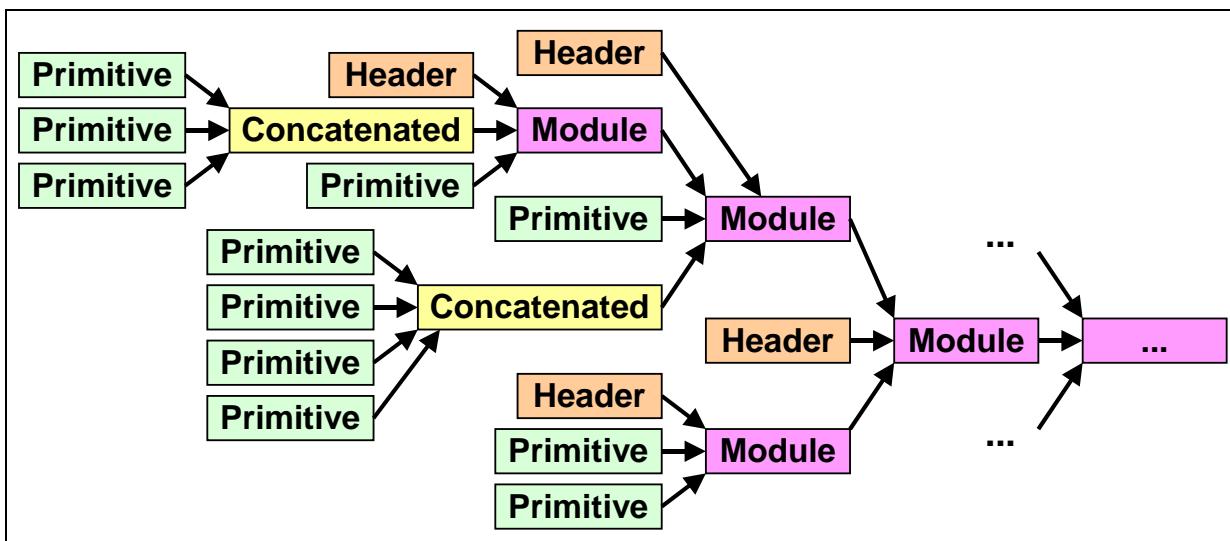
Mission data files for precision guided munitions complying with this standard shall comply with the requirements detailed in section 5.

## **5. DETAILED REQUIREMENTS**

The requirements in this section apply to and define the format and structure of data files complying with this standard, and the formatting of data elements used in the files.

## 5.1 Organization of data in files.

- a. Every file shall be formatted as a single module.
  - b. Data elements shall be ordered into modules containing a header followed by data content. The content may include any combination and quantity of subordinate primitive elements, concatenated elements, and/or other modules. Figure 2 is a characteristic example of the logical organization of a file showing how primitive, concatenated, and module data elements may be built into larger constructs.



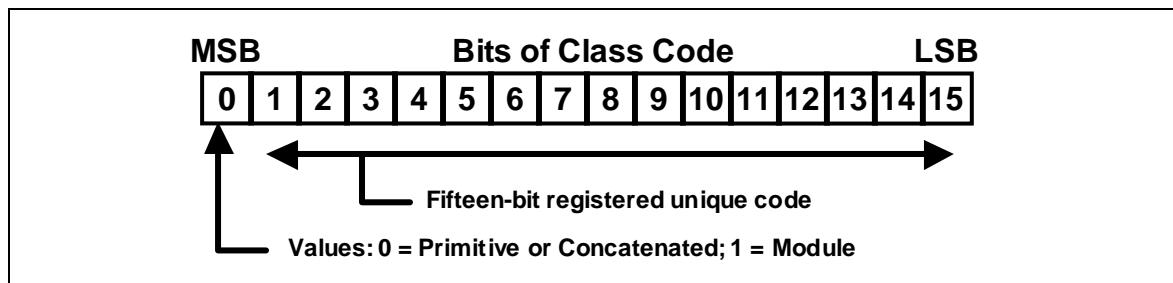
**FIGURE 2.** Data file organization.

- c. All fields shall be sequenced and ordered with the most significant part (bit, byte, word, etc.) first, and the first bit, when displayed, shall be on the left. Binary subparts of larger-order binary elements, such as bits in bytes or bytes in words or text fields, shall be ordered with the most significant subpart first in sequence and numbered "0".
  - d. Fields used as counts shall start counting at zero for the first element and count upwards.

**MIL-STD-3014**

e. Unless specifically defined in this standard, all ASCII text characters used in defined fields shall be upper case. General exceptions to this rule are fields whose format is defined as "ASCII text," such as the optional header fields "NOTE" and "SOURCE".

f. Every data element shall have a unique, registered, 16-bit class code assigned in the registry entry for that data element. Class codes are used in the Header's Element List to indicate the presence of data elements. Data elements are located in the data following the header. A data element's class code's 0th (most significant) bit shall be asserted or negated as shown on figure 3.

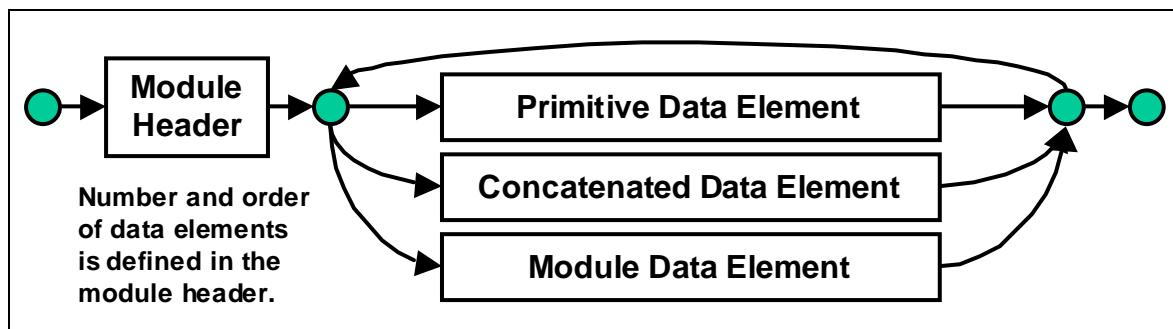


**FIGURE 3. Order of bits in class codes.**

g. Every data element shall have an associated 256-character, left justified, ASCII text, element DESCRIPTOR field that describes the overall data content and purpose of the element. The DESCRIPTOR field is for reference in the online registry only, and not for inclusion in files.

## 5.2 Order of module data elements.

Figure 4 shows the general layout and order of any module, as defined in this section.



**FIGURE 4. Order of a module's header and data.**

a. Every module data element shall be composed of a header followed by one or more subordinate data elements of any form: primitive, concatenated, or module. The header defines the purpose and content of the module in a manner that supports ready access to any one or more elements in the module. A registry of standard module designations, along with the general purpose and content of each, is found in the Module Data Element Registry located at <HTTP://MIL-STD-3014.NAVY.MIL>.

**MIL-STD-3014**

- b. Unless otherwise specified in the Module Data Element Registry for any specific module class code, the order of data elements within a module is explicitly not controlled by this standard; all software functions that read the content of modules in accordance with this standard shall be capable of accepting required and properly formatted data within a module, irrespective of its sort order within the module. As an example of a legitimate requirement for order of elements, flight path modules might require the module order of the waypoint data elements that define the flight path to match the order in which those waypoints are flown.
- c. The order of data elements within a module shall be the same as the order of their class codes in that module's header.

**5.2.1 Use of nonregistered values in header fields and data elements.**

- a. Nonregistered values shall not be used in any data element or header field that is controlled by a registry except as explicitly specified in the text of this standard or its appendices.
- b. All modules shall use a registered class code.

**5.2.2 Use of elements in program-unique modules.**

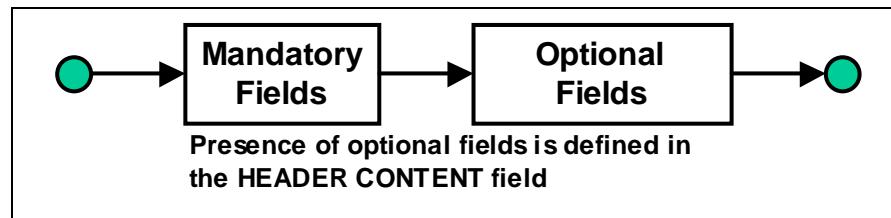
Primitive and concatenated data elements used in program-unique modules may be of standard or program-unique format. Program-unique modules at any level of module below the complete mission file (mission plan or mission update) level shall contain only primitive and/or concatenated data elements.

**5.2.3 Use and interpretation of generic element types for specific purposes.**

Because the module provides context information, the most generic possible form of a data type shall be used in preference to a more specific or limited data type. For example, when a NAV 3D ROUTE module is used for a single-warhead weapon, a target coordinate in that module shall be the generic GEO CORD 3D data type, in preference to the more specific WPT 3D CORD data type, because the last waypoint in the route is inherently the target.

**5.2.4 Order of module headers.**

Module headers shall consist of several mandatory and optional fields, ordered in a specific sequence, as specified in 5.3 and 5.4, and shown on figure 5. All header fields shall be designated as either fixed or variable length. Variable length fields shall be immediately preceded by a size field defining their size. The presence of optional fields shall be defined by the setting of bit flags in the HEADER CONTENTS field as shown on figure 5.



**FIGURE 5. Order of mandatory and optional header fields.**

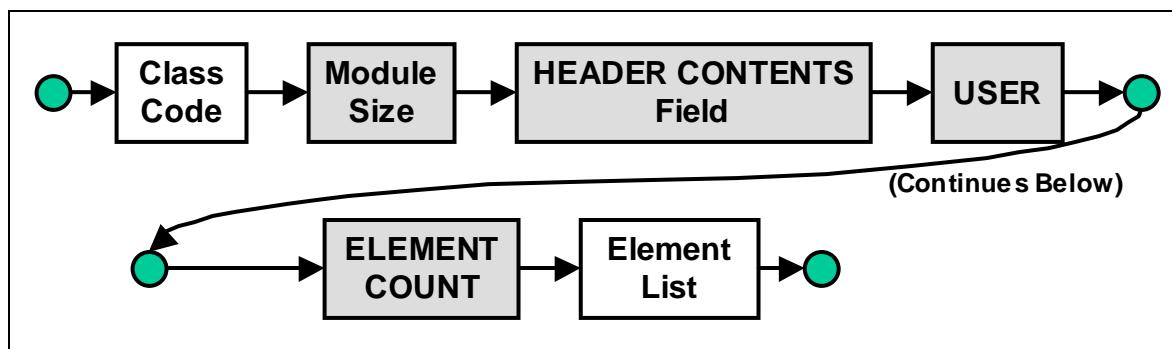
**MIL-STD-3014****5.3 Definition of mandatory header fields.**

The header of every module data element shall begin with the following mandatory fields as shown in table I and shall be in the order shown on figures 6 and 7.

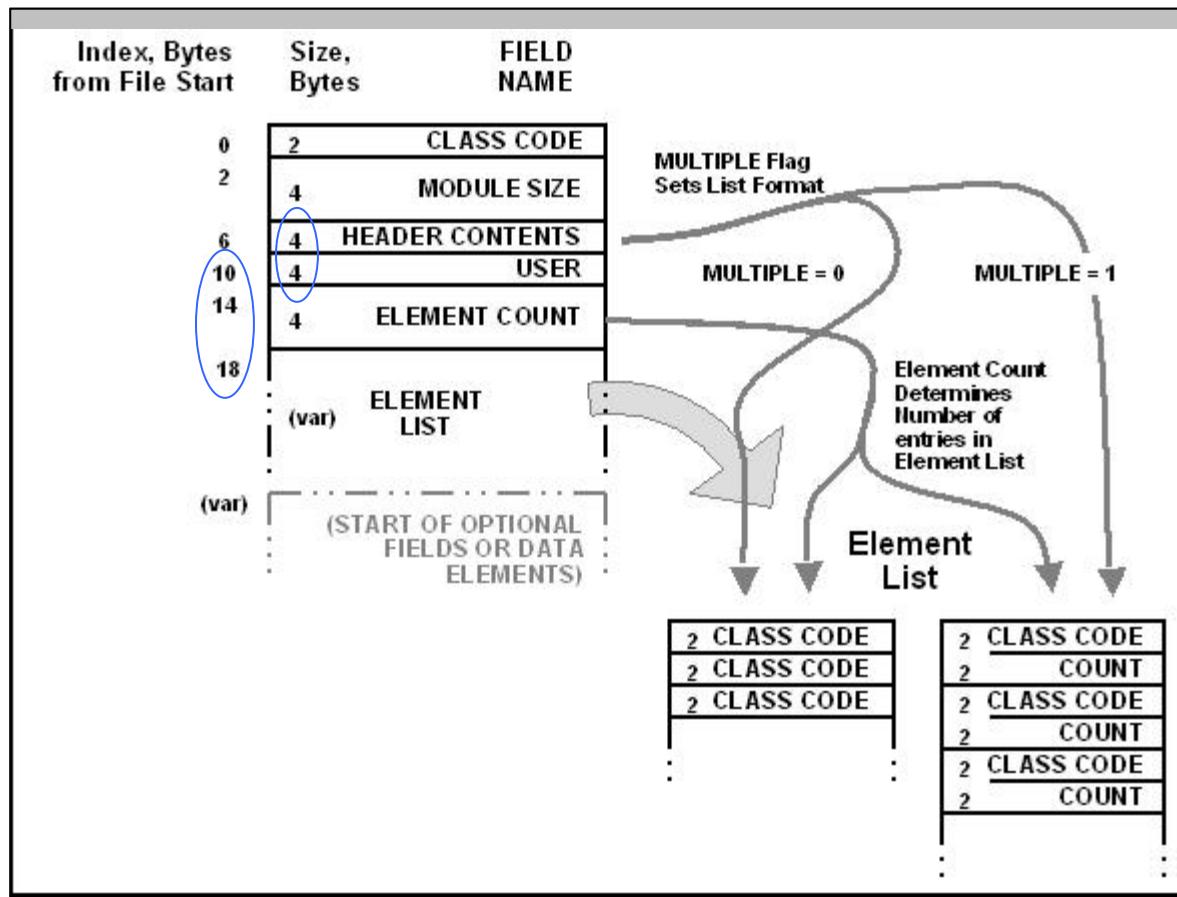
**TABLE I. Mandatory header field sequence.**

<b>Field Description</b>	<b>Ref. Para.</b>	<b>Interpretation</b>	<b>Size, bytes</b>	<b>Format</b>	<b>Registry</b>
CLASS CODE	5.3.1	Description of module's functional purpose	2	Serial registry number	Yes
MODULE SIZE	5.3.2	Total size (header & data elements) bytes	4	Unsigned integer	--
HEADER CONTENTS	5.3.3	Flags defining included optional field sets	4	Field of 32 bit-level flags	--
USER	5.3.4	Intended user for this module	4	See registry	Yes
ELEMENT COUNT	5.3.5	Number of primitives/congregated/sub-ordinate modules	4	Unsigned integer	--
ELEMENT LIST	5.3.6	List of class codes with (optional) associated quantities, one per element	n x 4 <sup>1</sup>	See ref. para.	--

<sup>1</sup> "n" = the value in the ELEMENT COUNT field, and represents the number of data elements contained in the module.

**FIGURE 6. Order of mandatory header fields in syntax diagram format.**

## MIL-STD-3014

**FIGURE 7. Order of mandatory header fields in field size format.****5.3.1 Module CLASS CODE field.**

- a. The module CLASS CODE shall be a two-byte hexadecimal field, whose registered values are defined in the Module Class/Type Registry, Appendix C. The class code is used in the header's element list, to identify the data element located in the data following the header. The 0<sup>th</sup> bit of the class code for a module shall be asserted.
- b. Every module data element shall have an associated 256-character, left-justified ASCII text element DESCRIPTOR field (for reference in the online registry only and not for inclusion in files) that describes the overall data content and purpose of the element.

**5.3.2 MODULE SIZE field.**

- a. The MODULE SIZE field shall define the total length of the module, in bytes, from the start of the header to the end of the last data element in the module.
- b. The MODULE SIZE field shall be a 32-bit unsigned integer.

**MIL-STD-3014**

c. All software functions that produce modules in accordance with this standard shall update the HEADER SIZE field of any module whose size is changed, and accordingly updating the size fields of any superior modules within which the edited module resides.

### **5.3.3 HEADER CONTENT field.**

a. The HEADER CONTENT field shall be a 32-bit field consisting of 32 one-bit flags indicating the presence of optional header information (see 5.4). Asserting a bit flag shall indicate the presence of the option; negating the bit flag shall indicate its absence. Sets of fields that are inherently correlated and grouped shall be represented by a single flag, as defined in table II and 5.4.

b. The definition of the header options whose presence is flagged by each bit in the header contents field is shown in table II and in the referenced paragraphs.

**TABLE II. HEADER CONTENT field format.**

<b>Bit No.</b>	<b>Flag Name</b>	<b>Invokes</b>	<b>Para Ref.</b>
0	MULTIPLE	Multiple count option in Element List	5.3.6
1	CREATE	Creator definition fields	5.4.1
2	CODE	Encoding/encryption definition fields	5.4.2
3	AUTH	Source verification/Digital Signature definition fields	5.4.3
4	SOURCE	“Built-from Module” definition fields	5.4.4
5	NOTE	Editor notes fields	5.4.5
6	(reserved)	(reserved)	(reserved)
7	(reserved)	(reserved)	(reserved)
8	(reserved)	(reserved)	(reserved)
9	(reserved)	(reserved)	(reserved)
10	(reserved)	(reserved)	(reserved)
11	(reserved)	(reserved)	(reserved)
12	(reserved)	(reserved)	(reserved)
13	(reserved)	(reserved)	(reserved)
14	(reserved)	(reserved)	(reserved)
15	(reserved)	(reserved)	(reserved)

### **5.3.4 USER field.**

a. The USER field shall define the end user of the module. The USER field shall be a numeric code, four bytes long, and shall be assigned by the organization that manages the User Registry (see 6.1).

**MIL-STD-3014**

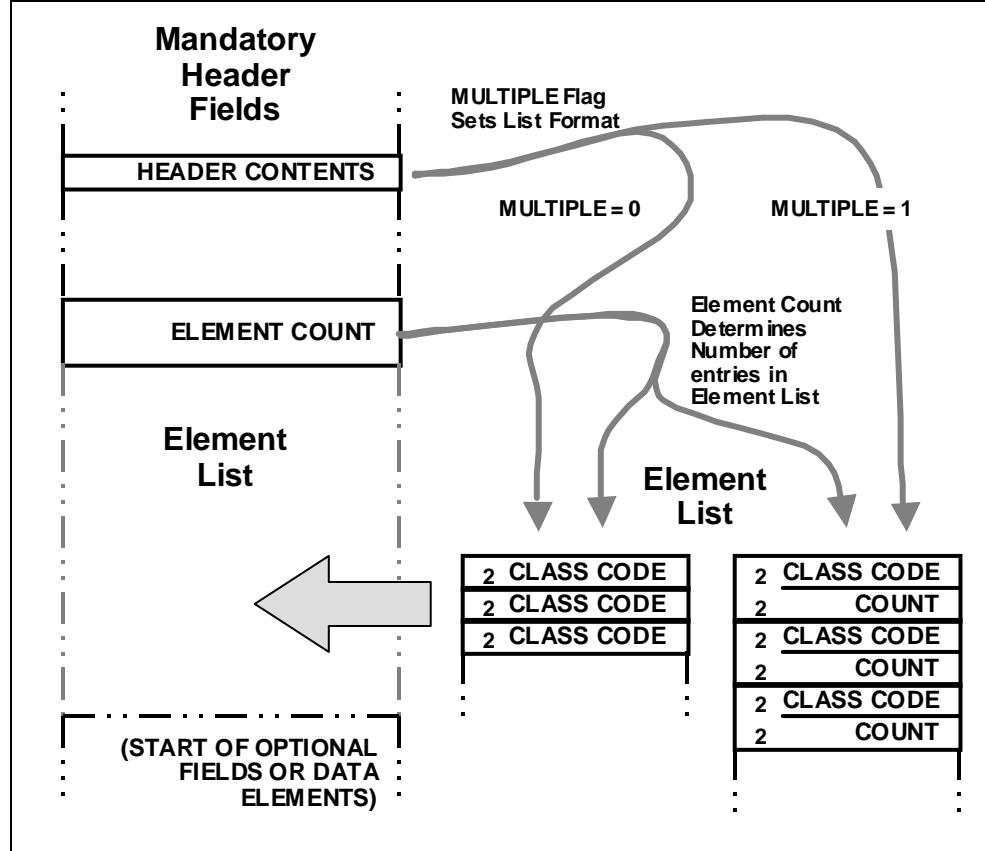
b. The User Registry shall be grouped into either end-user computer-based systems that act upon the mission data contained in the module or systems whose task is primarily to distribute or forward contained mission data to other systems. The first category, end users, is exemplified by precision-guided weapons, and the second category is exemplified by attack aircraft that launch the weapons.

c. If a module is placed within another module, and represents only a part of the mission data to be transmitted to a user system, it shall be assigned a user code of 00000000(hex.) If the module is designed for end use as mission data within the mission computer of the user, the user code shall be assigned in a sequence counting upwards from 00000001(hex.) If a module is intended for a system that will forward or distribute the enclosed data to other systems, the user code shall be assigned in a sequence counting down from FFFFFFFF(hex.).

**5.3.5 ELEMENT COUNT field.**

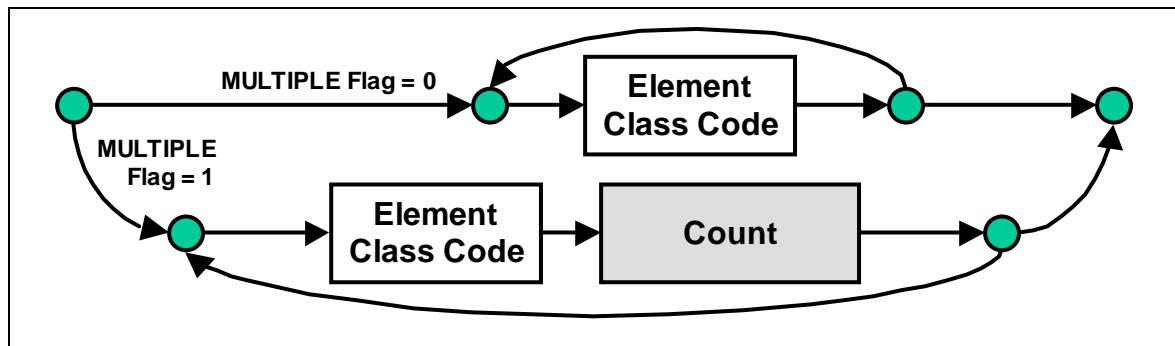
a. The ELEMENT COUNT field shall be an integer representing the number of immediately subordinate data elements in the module, as shown on figure 8. Note that this count does not include elements within subordinate modules (e.g., if a module contains two primitives plus a subordinate module that contains five elements, the total element count for the top module would be three: two primitives plus one module).

b. The ELEMENT COUNT field shall be formatted as a 4-byte, unsigned integer.

**FIGURE 8. Structure of element list.**

**MIL-STD-3014****5.3.6 ELEMENT LIST field.**

- a. The ELEMENT LIST field shall be a grouping of one or more ELEMENT fields, as shown on figure 9.



**FIGURE 9. Order of entries in element list.**

- b. Bit 0 of the HEADER CONTENT field (see 5.2.3) is the MULTIPLE flag. It shall determine which of two formats defines the element list.
- c. The MULTIPLE flag shall be negated to indicate that multiple element counts are disabled. When the MULTIPLE flag is negated, ELEMENT fields shall consist of a sequence of 2-byte class codes, one for each data element following the header in the module. The sequence of data elements following the header shall be the same as the sequence of ELEMENT fields in the element list. The length of the ELEMENT LIST shall be 2 bytes multiplied by the value of the ELEMENT COUNT field.
- d. The MULTIPLE flag shall be asserted to indicate that multiple element counts are enabled. When the MULTIPLE flag is asserted, ELEMENT fields shall be concatenations of field pairs consisting of 2-byte class codes, each followed by a 2-byte, positive GENERAL QUANTITY field describing the count of the immediately subordinate data elements in the module of the type defined by that class code. The length of this field shall be 4 (2 + 2) bytes multiplied by the value of the ELEMENT COUNT field.
- e. Whichever format is selected for the ELEMENT LIST field, the sequence and count of data elements following the header shall follow the sequence and count of ELEMENT descriptors in the ELEMENT LIST field.

NOTE: The ELEMENT LIST field is intended to aid in the parsing of file information by the using entity, and by editors. Indexing into the beginning of any subordinate module can be calculated by summing the known sizes of preceding primitive and concatenated data elements, and reading the size of preceding modules from their respective headers.

**5.4 Definition of optional header fields.**

Immediately following the mandatory fields in the header of every modular data element, the following optional fields shall be included as flagged in the mandatory HEADER CONTENTS field. The order of optional header fields, when included, shall be as shown on figures 10 and 11. Bits in the HEADER CONTENTS field that are not defined in table III shall be reserved for future use and shall be negated.

**MIL-STD-3014****TABLE III. Optional header field sequence.**

<b>Field Description</b>	<b>Flag in HDR CONTENT</b>	<b>Ref. Para.</b>	<b>Interpretation</b>	<b>Size, bytes</b>	<b>Format</b>	<b>Registry</b>
CREATE TIMESTAMP	Bit 1; CREATE	5.4.1.1	Date of last edit	8	TIME STAMP	--
CREATE ORG	Bit 1; CREATE	5.4.1.2	See Registry	8	8 ASCII Characters	Yes
CREATE INDIV	Bit 1; CREATE	5.4.1.3	Person generating module	8	8 ASCII Characters	--
CREATE SERNO	Bit 1; CREATE	5.4.1.4	Organization-controlled serial number for module	8	Unsigned integer	--
CODE TYPE	Bit 2; CODE	5.4.2.1	Type of post-encoding of module	4	4 ASCII characters	Yes
CODE SIZE	Bit 2; CODE	5.4.2.2	Size of decoding info	2	Unsigned integer	--
CODE KEY	Bit 2; CODE	5.4.2.3	Information for decoding	(var)	See CODE TYPE	--
AUTH TYPE	Bit 3; AUTH	5.4.3.1	Type of source verification	4	4 ASCII characters	Yes
AUTH SIZE	Bit 3; AUTH	5.4.3.2	Size of verification data	2	Unsigned integer	--
AUTH STAMP	Bit 3; AUTH	5.4.3.3	Information for source verification	(var)	See CODE TYPE, CODE SIZE	--
SOURCE SIZE	Bit 4; SOURCE	5.4.4.1	Size of following source field	2	Unsigned integer	--
SOURCE	Bit 4; SOURCE	5.4.4.2	Creator's source reference	(var)	ASCII free text	--
NOTE SIZE	Bit 5; NOTE	5.4.5.1	Size of creator notes	2	Unsigned integer	--
NOTE	Bit 5; NOTE	5.4.5.2	Creator's notes about this module	(var)	ASCII free text	--

## MIL-STD-3014

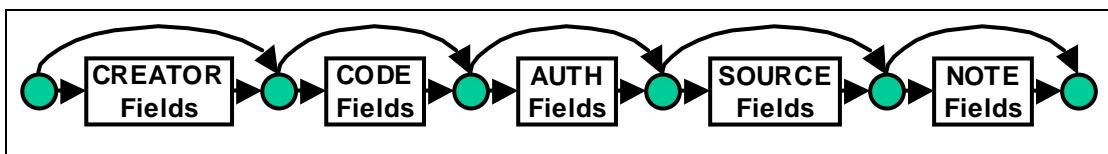


FIGURE 10. Order of optional header fields in syntax diagram format.

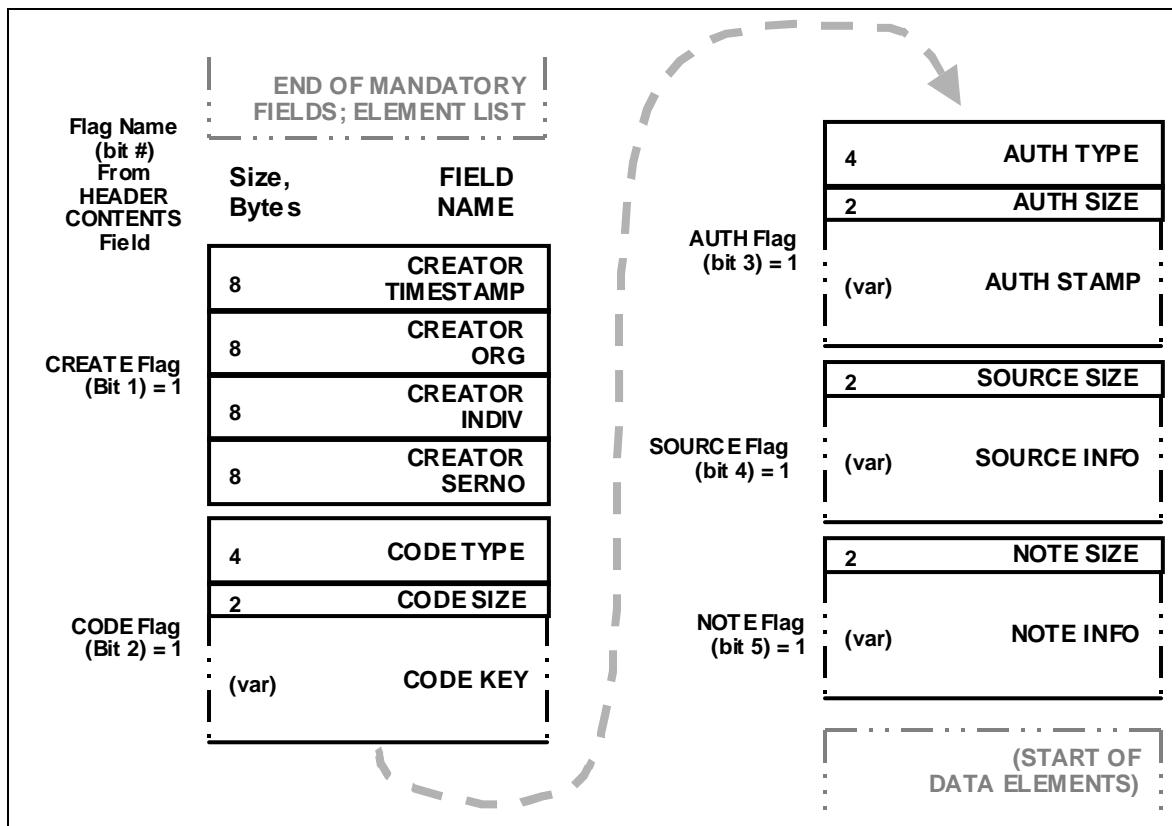


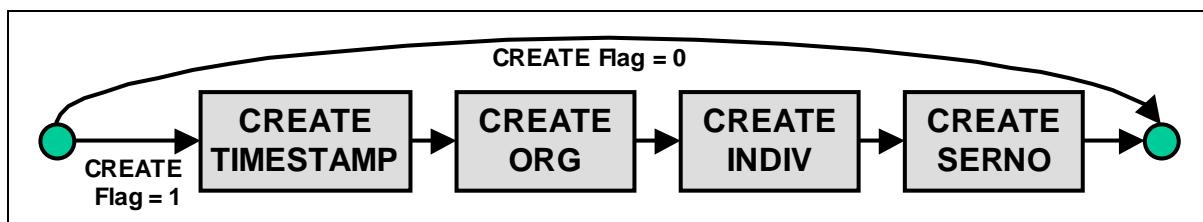
FIGURE 11. Order of optional header fields (when fully populated) in field size format.

The fields in this section shall be configured as described, if they are used. Use of these fields in any module is optional and is invoked by flags asserted in the mandatory HEADER CONTENTS field (see 5.2.3). Sequence of standard optional fields shall be as listed below.

MIL-STD-3014

### 5.4.1 Creator fields

There are four fields used to provide traceability to the module's creator. The fields are described individually, but shall be invoked as a set, by the assertion of the CREATE flag (bit 1) in the HEADER CONTENTS field. Sequence of the fields shall be as shown on figure 12.



**FIGURE 12.** Order of CREATE fields.

#### **5.4.1.1 CREATE TIMESTAMP field.**

- a. The CREATE TIMESTAMP field shall document and provide traceability of the module.
  - b. The CREATE TIMESTAMP field shall be formatted in the TIME DATE GROUP concatenated data element format.
  - c. The CREATE TIMESTAMP field shall be automatically inserted by the module editor, based on option settings, when the CREATE flag is asserted, at the time editing of the module is completed.

#### 5.4.1.2 CREATE ORG field.

- a. The CREATE ORG field shall document and provide traceability of the organization of the most recent editor of the module.
  - b. The CREATE ORG field shall be formatted as an 8-character ASCII field, with permissible registered organization identifier values shown in the Creator Registry located at <HTTP://MIL-STD-3014.NAVY.MIL>.
  - c. The CREATE ORG field shall be automatically inserted by the module editor at the same time as the CREATE DATE.
  - d. Units that are registered to create files with identification values in the CREATE field shall maintain internal registries for, and ensure configuration management of, the other CREATE fields (CREATE TIMESTAMP, CREATE INDIV, CREATE SERNO).

#### 5.4.1.3 CREATE INDIV field.

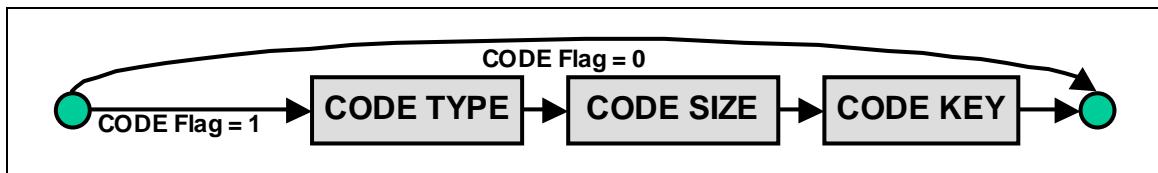
- a. The CREATE INDIV field shall document and provide traceability of the individual who most recently edited the module.
  - b. The CREATE INDIV field shall be formatted as an 8-character ASCII field, and must be filled in with a non-blank value according to the doctrine of the creating organization.
  - c. The CREATE INDIV field shall be automatically inserted by the module editor, based on option settings, when the CREATE flag is asserted, at the same time as the CREATE TIMESTAMP.

**MIL-STD-3014****5.4.1.4 CREATE SERNO field.**

- a. The CREATE SERNO field shall document and provide traceability of the most recent editor of the module.
- b. The CREATE SERNO field shall be formatted as a 32-bit unsigned integer field, and must be filled in with a non-blank value according to the doctrine of the creating organization. The value must be unique across the organization identified in the CREATE ORG field, for modules of the type identified by that Module's class code.
- c. The CREATE SERNO field shall be automatically inserted by the module editor, based on option settings, when the CREATE flag is asserted, at the same time as the CREATE TIMESTAMP.

**5.4.2 CODE-related fields.**

There are three CODE-related fields. The fields are described individually below, but are invoked together, as a package, by the assertion of the CODE flag (bit 2) in the HEADER CONTENTS flag field. Sequence of the fields shall be as listed on figure 13.



**FIGURE 13. Order of CODE fields.**

**5.4.2.1 CODE TYPE field.**

- a. The CODE TYPE field shall be used to identify the registered type of encoding applied to the contents of the module. Any coding shall be applied to the contents at the module level, irrespective of any other encoding of subordinate or superior modules. Such encoding shall be applied to the contents of the module as a single block, exclusive of the header. Encoding schemas include compression and encryption algorithms.
- b. The CODE TYPE field shall be formatted as a 4-character ASCII field, with permissible registered encoding schema values shown in the Code registry located in <HTTP://MIL-STD-3014.NAVY.MIL>.
- c. The CODE TYPE field shall be automatically inserted by the module editor, and the CODE flag asserted in the HEADER CONTENTS field, at the time the module's contents are encoded.

**5.4.2.2 CODE KEY SIZE field.**

- a. The CODE KEY SIZE field shall define the size of any reference information or key that must be conveyed to support decoding or decrypting of the module, such as a public key used in a public key interface encryption algorithm, or the checksum of a compression algorithm.
- b. The CODE KEY SIZE field shall be a 16-bit, positive-signed integer, defining the length of the key.

**MIL-STD-3014**

c. The CODE KEY SIZE field shall be automatically inserted by the module editor at the time the module contents are encoded. A code size value of zero shall be used when a code key is not required.

**5.4.2.3 CODE KEY field.**

a. The CODE KEY field shall include any reference information or key that must be conveyed to support decoding or decrypting of the module. Examples are a public key used in a public key interface encryption algorithm, or a checksum, or other similar data required for decoding or decryption.

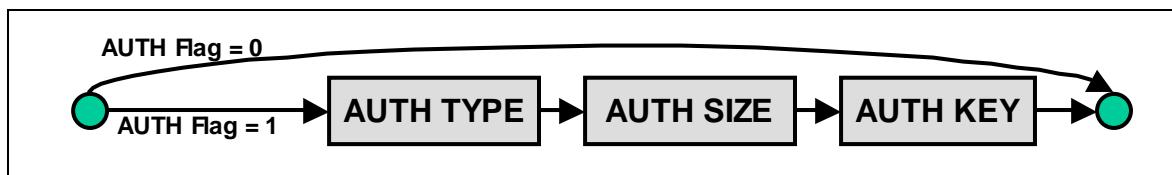
b. The CODE KEY format shall be as defined in the Code Type registry located in <HTTP://MIL-STD-3014.NAVY.MIL>.

c. The CODE KEY field shall be automatically inserted by the module editor at the time the module's contents are encoded.

d. The CODE KEY field is not a mandatory field. If the encoding or encryption process defined by the Code Type does not require the transmission of decryption data, the CODE KEY SIZE field value shall be set to zero.

**5.4.3 Authorizer-related fields.**

There are three Authorizer-related fields. The fields are described individually below, but are invoked together, as a package, by the assertion of the AUTH flag (bit 3) in the HDR CONTENT flag field. Sequence of the fields shall be as listed on figure 14.



**FIGURE 14. Order of AUTH fields.**

**5.4.3.1 AUTHORIZER TYPE field.**

a. The AUTHORIZER TYPE field shall identify the type of digital signature or other source verification data provided with the module.

b. The AUTHORIZER TYPE field shall be formatted as a 4-character ASCII field, with permissible registered encoding schema values shown in the Code registry.

c. The AUTHORIZER TYPE field shall be automatically inserted by the module editor, and the AUTH flag asserted in the HDR CONTENT field, at the time the module's contents are encoded.

**5.4.3.2 AUTHORIZER SIZE field.**

a. The AUTHORIZER SIZE field shall define the size of the digital signature information or key that must be conveyed to support source verification of the module.

**MIL-STD-3014**

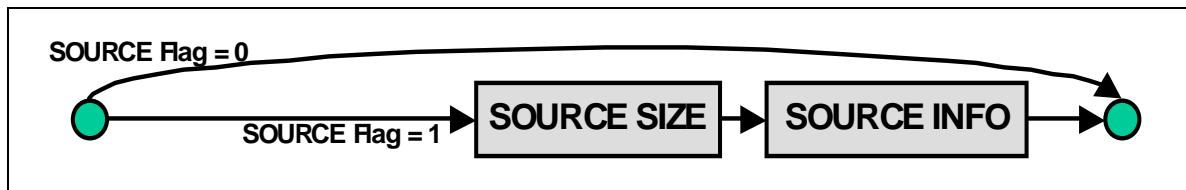
- b. The AUTHORIZER SIZE field shall be a 16-bit, positive-signed integer, which defines the length of the key.
- c. The AUTHORIZER SIZE field shall be automatically inserted by the module editor at the time the module's contents are encoded.

**5.4.3.3 AUTHORIZER STAMP field.**

- a. The AUTHORIZER STAMP KEY field shall include the digital signature information or key that is conveyed to support source verification of the module.
- b. The AUTHORIZER STAMP KEY format shall be as defined in the Authorization Type registry located in <HTTP://MIL-STD-3014.NAVY.MIL>.
- c. The AUTHORIZER STAMP KEY field shall be automatically inserted by the module editor at the time the module's contents are encoded.

**5.4.4 Source fields.**

There are two source-related fields. The fields are described individually below, but are invoked together, as a package, by the assertion of the SOURCE bit (see table II) in the HDR CONTENT flag field. Sequence of the fields shall be as listed on figure 15.



**FIGURE 15. Order of SOURCE fields.**

**5.4.4.1 SOURCE SIZE field.**

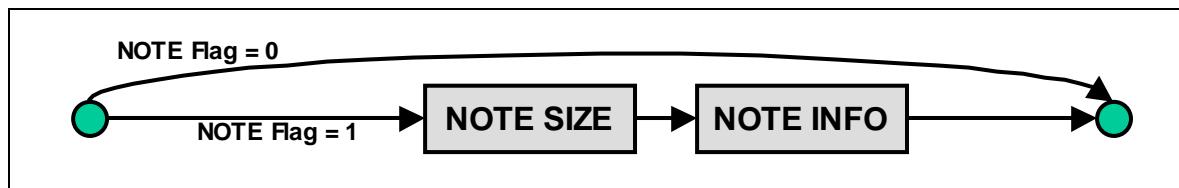
- a. The SOURCE SIZE field shall define the size of the SOURCE field that the module's authoring organization chooses to include in the module.
- b. The SOURCE SIZE field shall be a 16-bit, positive-signed integer defining the length of the key, up to 16K bytes.
- c. The SOURCE SIZE field shall be automatically inserted by the module editor, and the SOURCE bit (bit 3) asserted in the HEADER CONTENTS field, at the time the module's note is completed.

**5.4.4.2 SOURCE field.**

The SOURCE field is a free text field, up to 16K bytes, that may be used at the author's discretion or by organizational doctrine, to provide reference to the source of data used to generate this module. One of the most common references will be the CREATE fields from another module, from which this one was cloned and modified.

**MIL-STD-3014****5.4.5 NOTE-related fields.**

There are two NOTE-related fields. The fields are described individually below but are invoked together, as a package, by the assertion of the NOTE bit (see table II) in the HEADER CONTENTS flag field. Sequence of the fields shall be as listed on figure 16.

**FIGURE 16. Order of NOTE fields.****5.4.5.1 NOTE SIZE field.**

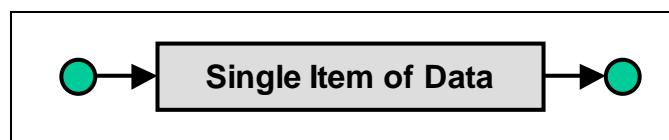
- a. The NOTE SIZE field shall define the size of any note that the module's authoring organization chooses to include in the module.
- b. The NOTE SIZE field shall be a 16-bit, positive-signed integer defining the length of the note, up to 16K bytes.
- c. The NOTE SIZE field shall be automatically inserted by the module editor at the time the module's note is completed.

**5.4.5.2 NOTE field.**

- a. The NOTE field shall be an ASCII text field, up to 16K bytes in length, to be used at the author's discretion, for purposes of documentation or enhanced awareness of future operators who might examine the module.
- b. The computer-based system that acts upon the information contained in the module shall perform its mission unaffected by the absence, presence, or content of the NOTE field. This requirement does not apply to software modules whose sole purpose is to forward the contents of the NOTE field for display to a human operator.

**5.5 Order of primitive data elements.**

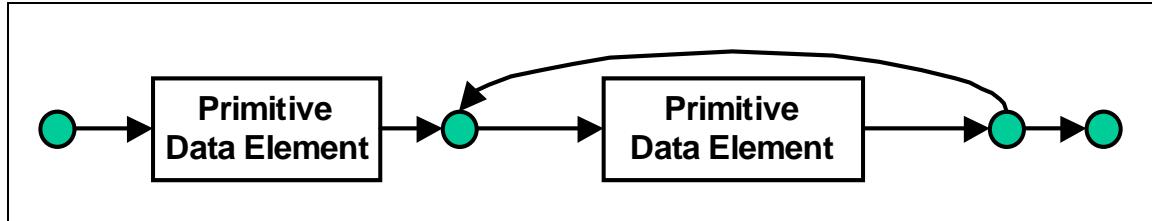
- a. Every primitive data element shall have a specific bit order, as specified and registered in the Primitive Data Element Registry located in <HTTP://MIL-STD-3014.NAVY.MIL>.
- b. The organization of each primitive data element shall be as shown on figure 17.

**FIGURE 17. Order of primitive data elements.**

## MIL-STD-3014

### 5.6 Order of concatenated data elements.

- a. Every concatenated data element shall be composed a fixed sequence of two or more primitive data elements, as specified in the Concatenated Data Element Registry located in <HTTP://MIL-STD-3014.NAVY.MIL>.
- b. The order of each concatenated data element shall be as shown on figure 18.



**FIGURE 18. Order of concatenated data elements.**

**MIL-STD-3014****6. NOTES**

(This section contains information of a general or explanatory nature that may be helpful but is not mandatory.)

**6.1 Intended use.**

This standard is intended to be used to standardize military digital data applications for weapon system interfaces and applies to all systems that generate, modify, transport, or use digital mission plans for autonomous weapon systems.

**6.1.1 Implementation.**

Implementation and application of the standard are the responsibility of each military service, with technical guidance and direction provided by appropriate service program offices.

The directories to which this standard refers are located online at <HTTP://MIL-STD-3014.NAVY.MIL> and are maintained by the managing office. This organization meets (quarterly) to review requests for additions and changes. Any organizational entity may request changes or additions by completing and submitting the form found on the appropriate registry listing page.

Users are encouraged to use existing registered data elements unless no existing registered element describes the nature and quality of data required for a specific application. Any developer of systems that use mission data files in accordance with this standard may apply for a class code addition. To apply, complete an electronic form available on the appropriate registry URL and submit it to the e-mail address shown on the form. The managing office will issue new class codes in accordance with the guidance and criteria defined in this section.

**6.1.1.1 Registry additions.**

Any organizational entity may apply for a new data element, defined by a new class code. Forms for such requests are located on links within each registry. The registry authority will review requests under the guidelines given in 6.1.1. The registry authority will rule on requests for new class codes at least once per calendar quarter.

The managing office considers requests for additions to the data element and module type registries using the following criteria:

- a. Does the addition represent a fundamentally different data type than those already available, or a requirement for additional data resolution that is adequately justified in the minds of the managing office? (Requests for alternative data forms that fail this test will generally be rejected.)
- b. Is the data form considered sufficiently unusual in application that it is unlikely to be usable by others in the foreseeable future? (Requests for alternative data forms that fail this test will be considered first as general-purpose data entities, available for use by anyone.)
- c. Is a request for “reserved but undefined” data types justified based on the nature of the system and its classification? (Requests for an apparently excessive number of unique data or module types will be met with a recommendation to use conventional data types, and manage classification through the use of encryption or other means.)

**MIL-STD-3014****6.1.1.2 Registry changes.**

The managing office considers requests for changes to the data element and module type registries using the following criteria:

- a. Does the current state of the data element represent a threat of harm if left for use in its current form? (Requests that fail to address this criterion adequately will usually be rejected.)
- b. Is there a means of determining the complete user base for the data element or module type, to ensure that the user base is not adversely impacted? (Requests for changes that satisfy a., above, but which inadequately address this user base concern, will likely be considered alternatively as a request for addition to the registry. A caution will be placed on the registry change form, identifying the problem and recommending its nonuse.)

Registries identifying user organizations (Creator and User registries) will, upon initial application, be considered valid for 12 months from the date of approval by the managing office. Information defining the organizations may be updated at any time during the 12 months, and will automatically be renewed for a 12-month period from the date of change notice and its verification by the managing office. Organizational designators for which the renewal has expired will be labeled "Expired validity; may be obsolete."

**6.1.2 Use of optional fields.****6.1.2.1 Creator fields.**

Creator fields are used to document the source of a module. They are automatically invoked by the module editor upon assertion of the CREATE flag. They provide traceability to the organization, individual, and time that the module was last edited.

**6.1.2.2 Code fields.**

Code fields are used to identify any encoding or encryption that has been applied to the module. The CODE TYPE field identifies the specific encoding or encryption process used, and its registry is located in <HTTP://MIL-STD-3014.NAVY.MIL>. The registry contains complete information on execution of the process, by reference to appropriate standards and user documents, and identification of a point of contact that can provide more information. The CODE KEY field, with its preceding CODE KEY SIZE field, transports any ancillary information needed by the decoding or decrypting algorithm, such as a PKI public key, or a checksum for a compression algorithm.

**6.1.2.3 Authorizer fields.**

Authorizer fields are used to verify authorization or legitimacy of the module by using a digital signature or similar process. The AUTHORIZER TYPE field identifies the specific process in use, and its registry is located in <HTTP://MIL-STD-3014.NAVY.MIL>. The registry contains complete information on execution of the process, by reference to appropriate standards and user documents, and identification of a point of contact that can provide more information. The AUTHORIZER STAMP field, with its preceding AUTHORIZER SIZE field, transports any validating, authorizing, or signature information needed by the authorizing algorithm.

**MIL-STD-3014****6.1.2.4 Source fields.**

Source fields are used to define any modules from which the module being edited was derived. The SOURCE SIZE field defines the size of the SOURCE field. The SOURCE field is a free text format, and typically includes the mandatory header fields and the CREATE fields from the module that was used as the basis for creating a given module.

**6.1.2.5 Note fields.**

The NOTE field, supported by its preceding NOTE SIZE field, allows the author of a module to include any text notes he deems useful for user awareness or archival documentation purposes. This field is proscribed from being used as a substitute for any data element, and may not contain mission data for use by the destination computer system.

**6.2 Acquisition requirements.**

Acquisition documents should specify the title, number, and date of the standard.

**6.3 Tailoring guidance.**

All permissible tailoring of content within the scope of this standard, including flexibility of format, use of header fields, and additions and changes to standard and reserved directory listings, is identified in section 5. No further tailoring of this standard is permitted.

**6.4 Keyword list.**

authorizer field  
class code  
code field  
concatenated data element  
creator field  
data element  
data file  
field code  
file format  
header  
mission data  
mission file  
mission plan  
mission planning  
module  
munitions  
note field  
optional field  
precision guided munitions  
primitive data element  
source field  
weapon  
weapon system

## MIL-STD-3014

### 6.5 Change notations.

This standard is marked with lines and gray highlighting to indicate the location of modifications generated by this change. Specific changes are indicated using blue font; bold, blue type face in tables; and blue bands to indicate changes made on figures. This was done as a convenience and the Government assumes no liability whatsoever for any inaccuracies in these notations. Bidders and contractors are cautioned to evaluate the requirements of this document based on the entire content irrespective of the change notations.

Custodians:

Army - MI  
Navy - AS  
Air Force - 11

Preparing activity:

Air Force - 11  
(Project: MCCR-2005-008)

Review activities:

Army - AC, AV, CR

Industry:

SAE

NOTE: The activities listed above were interested in this document as of the date on the cover. Since organizations and responsibilities can change, you should verify the currency of the information above using the ASSIST Online database at <http://assist.daps.dla.mil/online/start/> or <http://assist.daps.dla.mil/quicksearch/>.