# Introduction

[8] XML Schema Part 1: Structures. 2nd ed. Henry S. Thompson, et al., eds. W3C Recommendation. N.p.: W3C, October 2004. <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>. [new entry]

[9] XML Schema Part 2: Datatypes. 2nd ed. P. Biron and A. Malhotra, eds. W3CRecommendation 28. n.p.: W3C, 2004. <https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/> [moved from Annex H... remove from the Annex]

# Overview

# APM

# AEM

# ACM

# CONSTRUCTING AN ADM/XML INSTANCE

## OVERVIEW

### This section provides more detailed instructions for the user on how to create an XML message based on one of the ASCII-text KVN-formatted messages described in Sections 3 through 5 of this document.

### This section applies only to the XML representation of the ADM messages. The ADM/XML schemas are available on the SANA Web site. SANA is the registrar for the protocol registries created under CCSDS. The ADM XML schemas explicitly define the permitted data elements and values acceptable for the XML versions of the ADM messages. The location of the ADM/XML schemas is:

APM: https://sanaregistry.org/r/ndmxml/ndmxml-1.0-apm-2.0.xsd

AEM: https://sanaregistry.org/r/ndmxml/ndmxml-1.0-aem-2.0.xsd

ACM: https://sanaregistry.org/r/ndmxml/ndmxml-1.0-acm-2.0.xsd

### Where possible these schemas use simple types and complex types used by the constituent schemas that make up Navigation Data Messages (see Reference [5]).

## ADM/XML BASIC STRUCTURE

### Each ADM shall consist of a <header> and a <body>.

### The <body> shall consist of one or more <segment> constructs (one for the APM, one or more for the AEM, one for the ACM).

### Each <segment> shall consist of one <metadata>/<data> pair, as shown in Figure 6‑1.

NOTE: An AEM may have more than one segment, in which case the metadata/data pair is repeated in each segment.

|  |
| --- |
| <header></header><body> <segment> <metadata> </metadata> <data> </data> </segment></body> |

Figure 6‑1 : ADM XML Basic Structure

## ADM/XML TAGS

### An ADM XML tag shall be all uppercase if it corresponds directly to a KVN keyword from the Header, Metadata, or Data sections.

### The 'CCSDS\_A\*M\_VERS' keyword and value shall appear as XML attributes of the root element rather than as XML elements. This is an exception where there is not a strict correspondence between keywords in the KVN and tags in the XML implementations, specifically, the ‘CCSDS\_A\*M\_VERS’ keywords from the Headers for the APM, AEM, and ACM respectively.

### ADM XML tags related to the XML message structure (i.e., that do not correspond directly to a KVN keyword) shall be in ‘lowerCamelCase’ (e.g., <header>, <segment>, <metadata>, <attitudeStateType>, etc.).

## CONSTRUCTING AN ADM/XML INSTANCE

### OVERVIEW

This subsection provides more detailed instructions for the user on how to create an XML message based on the ASCII-text KVN-formatted messages described in Sections 3 through 5.

### XML VERSION

The first line in the instantiation shall specify the XML version:

<?xml version="1.0" encoding="UTF-8"?>

This line must appear on the first line of each instantiation, exactly as shown.

### BEGINNING THE INSTANTIATION: root element TAG

Each instantiation shall have a 'root element tag' that identifies the message type and other information such as where to find the applicable schema, required attributes, etc.

The root element tag in an ADM/XML instantiation shall be one of those listed in Table 6‑1.

Table 6‑1 : ADM/XML Root Element Tags

|  |  |
| --- | --- |
| **Root Element Tag** | **Message Type** |
| <apm></apm> | Attitude Parameter Message |
| <aem></aem> | Attitude Ephemeris Message |
| <acm></acm> | Attitude Comprehensive Message |

The XML Schema Instance namespace attribute must appear in the root element tag of all ADM/XML instantiations, exactly as shown:

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

If it is desired to validate an instantiation against the CCSDS Web-based schema, the xsi:noNamespaceSchemaLocation attribute must be coded as a single string of non-blank characters, with no line breaks, exactly as shown:

xsi:noNamespaceSchemaLocation="http://sanaregistry.org/r/ndmxml/ndmxml-1.0-master.xsd"

NOTE – The length of the value associated with the xsi:noNamespaceSchemaLocation attribute can cause the string to wrap to a new line; however, the string itself contains no breaks.

There are two attributes that are required in the root element tag of an ADM/XML single message instantiation, specifically, the CCSDS\_xxx\_VERS keyword that is also part of the standard KVN header, and the Blue Book version number.

The final attributes of the root element tag shall be ‘id’ and ‘version’.

The ‘id’ attribute shall be ‘id="CCSDS\_xxx\_VERS"’, where xxx = AEM, APM, or ACM.

The ‘version’ attribute shall be ‘version="2.0"’.

NOTE – The following example root element tag for an APM instantiation combines all the directions in the preceding several subsections:

<?xml version="1.0" encoding="UTF-8"?>

<apm xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>" xsi:noNamespaceSchemaLocation=

"<http://sanaregistry.org/r/ndmxml/ndmxml-1.0-master.xsd>" id="CCSDS\_APM\_VERS" version="2.0">

### THE ADM/XML HEADER SECTION

The ADMs (APM, AEM, ACM) shall share a standard header format, with tags <header> and </header> (see [5]).

Immediately following the <header> tag the message may have any number of <COMMENT></COMMENT> tag pairs.

The standard ADM header shall contain the following element tags:

1. <CREATION\_DATE>
2. <ORIGINATOR>
3. <MESSAGE\_ID>

NOTE – The rules for these keywords are specified in Tables 3-1, 4-2, 5-?. An example <header> section is shown immediately below:

<header>

<COMMENT>This is the common ADM/XML Header.</COMMENT>

<COMMENT>I can put as many comments here as I want,</COMMENT>

<COMMENT>including none.</COMMENT>

<CREATION\_DATE>2010-03-12T22:31:12.000</CREATION\_DATE>

<ORIGINATOR>AGENCYX</ORIGINATOR>

<MESSAGE\_ID>AGENCYX-1234</MESSAGE\_ID>

</header>

### THE ADM/XML BODY SECTION

After coding the <header>, the instantiation must include a <body></body> tag pair.

Inside the <body></body> tag pair must appear at least one <segment></segment> tag pair, depending on the particular ADM (APM, AEM, ACM).

Each <segment> must be made up of one <metadata></metadata> tag pair and one <data></data> tag pair.

### THE ADM/XML METADATA SECTION

All ADMs must have at least one Metadata section.

The Metadata Section shall be set off by the <metadata></metadata> tag combination.

Immediately following the <metadata> tag, the message may have any number of <COMMENT></COMMENT> tag pairs.

NOTE: The <COMMENT></COMMENT> placement is regulated by the XML schema. Standard XML comments, i.e. of the form <!-- comment content --> may be placed anywhere in the Metadata Section because they are ignored by the XML schema validator.

Between the <metadata> and </metadata> tags, the keywords shall be the same as those in the Metadata sections in Sections 3 through 5 of this document, with exceptions as noted in the subsections that discuss creating instantiations of the specific messages.

### THE ADM/XML DATA SECTION

All ADMs must have at least one data section.

The Data Section shall follow the Metadata Section and shall be set off by the <data></data> tag combination.

Immediately following the <data> tag, the message may have any number of <COMMENT></COMMENT> tag pairs.

NOTE: The <COMMENT></COMMENT> placement is regulated by the XML schema. Standard XML comments, i.e. of the form <!-- comment content --> may be placed anywhere in the Data Section because they are ignored by the XML schema validator.

Between the <data> and </data> tags, the keywords shall be the same as those in the data sections in Sections 3 through 5 of this document, with exceptions as noted in the subsections that discuss creating instantiations of the specific messages.

## Local Operations

### For use in a local operations environment, the NDM/XML schema set (which includes the ADM schemas) may be downloaded from the SANA web site to a local server that meets local requirements for operations robustness. See Reference [5].

### If a local version is used, the value associated with the xsi:noNamespaceSchemaLocation attribute must be changed to a URL that is accessible to the local server.

## CREATING AN APM INSTANTIATION

### An APM instantiation shall be delimited by the <apm></apm> root element tags using the standard attributes documented in 6.4.3.

NOTE – Figures <<TBD>> provide example APM instantiations.

### The final attributes of the <apm> tag shall be ‘id’ and ‘version’; the order in which these attributes are specified is not significant.

### The ‘id’ attribute shall be ‘id="CCSDS\_APM\_VERS"’.

### The ‘version’ attribute for the version of the APM shall be ‘version="2.0"’.

### The standard ADM/XML header shall follow the <apm> tag (see 6.4.4).

### The APM <body> shall consist of a single <segment>.

### The keywords in the <metadata> and <data> sections shall be those specified in Section 3.

### Tags for keywords shall be all uppercase, as in Section 3.

### Several of the APM/XML keywords may have a unit attribute, if desired by the APM producer, as illustrated in the following table:

Table 6‑2 APM XML Units

| **Keyword** | **Units** | **Example** |
| --- | --- | --- |
| Q1\_DOT | 1/s | <Q1\_DOT units="1/s">numeric-value</Q1\_DOT> |
| Q2\_DOT | 1/s | <Q2\_DOT units="1/s">numeric-value</Q2\_DOT> |
| Q3\_DOT | 1/s | <Q3\_DOT units="1/s">numeric-value</Q3\_DOT> |
| QC\_DOT | 1/s | <QC\_DOT units="1/s">numeric-value</QC\_DOT> |
| ANGLE\_1 | deg | <ANGLE\_1 units="deg">numeric-value</ANGLE\_1> |
| ANGLE\_2 | deg | <ANGLE\_2 units="deg">numeric-value</ANGLE\_2> |
| ANGLE\_3 | deg | <ANGLE\_3 units="deg">numeric-value</ANGLE\_3> |
| ANGLE\_1\_DOT | deg/s | <ANGLE\_1\_DOT units="deg/s">numeric-value</ANGLE\_1\_DOT> |
| ANGLE\_2\_DOT | deg/s | <ANGLE\_2\_DOT units="deg/s">numeric-value</ANGLE\_2\_DOT> |
| ANGLE\_3\_DOT | deg/s | <ANGLE\_3\_DOT units="deg/s">numeric-value</ANGLE\_3\_DOT> |
| ANGVEL\_X | deg/s | <ANGVEL\_X units="deg/s">numeric-value</ANGVEL\_X> |
| ANGVEL\_Y | deg/s | <ANGVEL\_Y units="deg/s">numeric-value</ANGVEL\_Y> |
| ANGVEL\_Z | deg/s | <ANGVEL\_Z units="deg/s">numeric-value</ANGVEL\_Z> |
| SPIN\_ALPHA | deg | <SPIN\_ALPHA units="deg">numeric-value</SPIN\_ALPHA> |
| SPIN\_DELTA | deg | <SPIN\_DELTA units="deg">numeric-value</SPIN\_DELTA> |
| SPIN\_ANGLE | deg | <SPIN\_ANGLE units="deg">numeric-value</SPIN\_ANGLE> |
| SPIN\_ANGLE\_VEL | deg/s | <SPIN\_ANGLE\_VEL units="deg/s">numeric-value</SPIN\_ANGLE\_VEL> |
| NUTATION | deg | <NUTATION units="deg">numeric-value</NUTATION> |
| NUTATION\_PER | s | <NUTATION\_PER units="s">numeric-value</NUTATION\_PER> |
| NUTATION\_PHASE | deg | <NUTATION\_PHASE units="deg">numeric-value</NUTATION\_PHASE> |
| IXX | kg\*m\*\*2 | <IXX units="kg\*m\*\*2">numeric-value</IXX> |
| IYY | kg\*m\*\*2 | <IYY units="kg\*m\*\*2">numeric-value</IYY> |
| IZZ | kg\*m\*\*2 | <IZZ units="kg\*m\*\*2">numeric-value</IZZ> |
| IXY | kg\*m\*\*2 | <IXY units="kg\*m\*\*2">numeric-value</IXY> |
| IXZ | kg\*m\*\*2 | <IXZ units="kg\*m\*\*2">numeric-value</IXZ> |
| IYZ | kg\*m\*\*2 | <IYZ units="kg\*m\*\*2">numeric-value</IYZ> |
| MAN\_DURATION | s | <MAN\_DURATION units="s">numeric-value</MAN\_DURATION> |
| MAN\_TOR\_X | N\*m | <MAN\_TOR\_X units="N\*m">numeric-value</MAN\_TOR\_X> |
| MAN\_TOR\_Y | N\*m | <MAN\_TOR\_Y units="N\*m">numeric-value</MAN\_TOR\_Y> |
| MAN\_TOR\_Z | N\*m | <MAN\_TOR\_Z units="N\*m">numeric-value</MAN\_TOR\_Z> |

### SPECIAL TAGS IN the APM/xml BODY

NOTE – In addition to the APM keywords specified in Section 3, there are several special tags associated with the APM body as described in the next few subsections. The information content in the APM is separated into constructs referred to as ‘logical blocks’. Special tags in the APM are used to encapsulate the information in the logical blocks of the APM.

The APM/XML tags used to delimit the logical blocks of the APM shall be drawn from the following table:

Table 6‑3 Special Tags in the APM/XML Body

|  |  |
| --- | --- |
| **APM Logical Block** | **Associated APM/XML Tag** |
| Attitude Quaternion | <quaternionState> <quaternion>  <quaternionDerivative>The <quaternionState> consists of the <quaternion> tag that contains the components of the quaternion itself, and the <quaternionDerivative> tag that contains the rate of change of the quaternion components |
| Euler Angle Elements | <eulerAngleElements> |
| Angular Velocity Vector | <angularVelocity> |
| Spin | <spin> |
| Inertia | <inertia> |
| Maneuver Parameters | <maneuverParameters> |

Between the begin tag and end tag (e.g., between <eulerAngleElements> and </eulerAngleElements>), the user shall place the keywords required by the specific logical block as specified in Section 3.

### DISCUSSION

This non-normative subsection discusses and provides examples of the use of quaternion tags in the APM.

The XML representations of quaternions in the ADM constituent messages share a common quaternion definition. However, there are some differences in those definitions in the underlying KVN definitions of the APM and AEM. The following examples are meant to illustrate the standard for representing quaternions in the APM.

Here is an example APM quaternion construct:

 <quaternionState>

 <EPOCH>2004-100T00:00:00Z</EPOCH>

 <REF\_FRAME\_A>ICRF</REF\_FRAME\_A>

 <REF\_FRAME\_B>ICRF</REF\_FRAME\_B>

 <quaternion>

 <Q1>0.00005</Q1>

 <Q2>0.87543</Q2>

 <Q3>0.40949</Q3>

 <QC>0.25678</QC>

 </quaternion>

 </quaternionState>

Here is an example APM quaternion construct with the optional derivative:

 <quaternionState>

 <EPOCH>2004-100T00:00:00Z</EPOCH>

 <REF\_FRAME\_A>ICRF</REF\_FRAME\_A>

 <REF\_FRAME\_B>ICRF</REF\_FRAME\_B>

 <quaternion>

 <Q1>0.00005</Q1>

 <Q2>0.87543</Q2>

 <Q3>0.40949</Q3>

 <QC>0.25678</QC>

 </quaternion>

 <quaternionDerivative>

 <Q1\_DOT>0.002</Q1\_DOT>

 <Q2\_DOT>0.003</Q2\_DOT>

 <Q3\_DOT>0.004</Q3\_DOT>

 <QC\_DOT>0.001</QC\_DOT>

 </quaternionDerivative>

 </quaternionState>

## CREATING AN AEM INSTANTIATION

### An AEM instantiation shall be delimited with the <aem></aem> root element tags using the standard attributes documented in 6.4.3.

NOTE – Figures <<TBD>> provide example AEM instantiations.

### The final attributes of the <aem> tag shall be ‘id’ and ‘version’; the order in which these attributes are specified is not significant.

### The ‘id’ attribute shall be ‘id="CCSDS\_AEM\_VERS"’.

### The ‘version’ attribute for the version of the AEM shall be ‘version="2.0"’.

### The standard ADM/XML header shall follow the <aem> tag (see 6.4.4).

### The AEM <body> shall consist of one or more <segment> constructs (see [5], section 3.4).

### The keywords in the <metadata> and <data> sections shall be those specified in Section 4.

### Tags for keywords shall be all uppercase as in Section 4.

### Although units are not specified in the KVN representation of the AEM, several of the AEM/XML keywords may have a unit attribute, if desired by the AEM producer, as illustrated in the following table:

Table 6‑4 AEM XML Units

| **Keyword** | **Units** | **Example** |
| --- | --- | --- |
| Q1\_DOT | 1/s | <Q1\_DOT units="1/s">numeric-value</Q1\_DOT> |
| Q2\_DOT | 1/s | <Q2\_DOT units="1/s">numeric-value</Q2\_DOT> |
| Q3\_DOT | 1/s | <Q3\_DOT units="1/s">numeric-value</Q3\_DOT> |
| QC\_DOT | 1/s | <QC\_DOT units="1/s">numeric-value</QC\_DOT> |
| ANGLE\_1 | deg | <ANGLE\_1 units="deg">numeric-value</ANGLE\_1> |
| ANGLE\_2 | deg | <ANGLE\_2 units="deg">numeric-value</ANGLE\_2> |
| ANGLE\_3 | deg | <ANGLE\_3 units="deg">numeric-value</ANGLE\_3> |
| ANGLE\_1\_DOT | deg/s | <ANGLE\_1\_DOT units="deg/s">numeric-value</ANGLE\_1\_DOT> |
| ANGLE\_2\_DOT | deg/s | <ANGLE\_2\_DOT units="deg/s">numeric-value</ANGLE\_2\_DOT> |
| ANGLE\_3\_DOT | deg/s | <ANGLE\_3\_DOT units="deg/s">numeric-value</ANGLE\_3\_DOT> |
| ANGVEL\_X | deg/s | <ANGVEL\_X units="deg/s">numeric-value</ANGVEL\_X> |
| ANGVEL\_Y | deg/s | <ANGVEL\_Y units="deg/s">numeric-value</ANGVEL\_Y> |
| ANGVEL\_Z | deg/s | <ANGVEL\_Z units="deg/s">numeric-value</ANGVEL\_Z> |
| SPIN\_ALPHA | deg | <SPIN\_ALPHA units="deg">numeric-value</SPIN\_ALPHA> |
| SPIN\_DELTA | deg | <SPIN\_DELTA units="deg">numeric-value</SPIN\_DELTA> |
| SPIN\_ANGLE | deg | <SPIN\_ANGLE units="deg">numeric-value</SPIN\_ANGLE> |
| SPIN\_ANGLE\_VEL | deg/s | <SPIN\_ANGLE\_VEL units="deg/s">numeric-value</SPIN\_ANGLE\_VEL> |
| NUTATION | deg | <NUTATION units="deg">numeric-value</NUTATION> |
| NUTATION\_PER | s | <NUTATION\_PER units="s">numeric-value</NUTATION\_PER> |
| NUTATION\_PHASE | deg | <NUTATION\_PHASE units="deg">numeric-value</NUTATION\_PHASE> |

### SPECIAL TAGS IN the AEM BODY

NOTE – In addition to the AEM keywords specified in Section 4, there are several special tags associated with the AEM body as described in the next few subsections.

The <attitudeState> tag shall be used to encapsulate the keywords associated with the structure of one of the attitude ephemeris data line types.

The ADM/XML tags used within the <attitudeState> structure shall be drawn from the following table:

Table 6‑5 Special Tags in the AEM/XML Body

|  |  |
| --- | --- |
| **AEM ‘ATTITUDE\_TYPE’ Metadata Value** | **Associated ADM/XML Tag in the <attitudeState>** |
| QUATERNION | <quaternionState> |
| QUATERNION/DERIVATIVE | <quaternionDerivative> |
| QUATERNION/ANGVEL | <quaternionAngvel> |
| EULER\_ANGLE | <eulerAngle> |
| EULER\_ANGLE/DERIVATIVE | <eulerAngleDerivative> |
| EULER\_ANGLE/ANGVEL | <eulerAngleAngvel> |
| SPIN | <spin> |
| SPIN/NUTATION | <spinNutation> |

Between the begin tag and end tag (e.g., between <quaternionState> and </quaternionState>), the user shall place the values required by the specific ephemeris data line type as specified in Section 4, Table 4-4.

In the XML representation of the AEM, the components of the <attitudeState> ephemeris data line must be represented with keywords (i.e., a tag).

The <attitudeState> keywords shall be the same as those defined for the same construct in the APM.

NOTE – In the KVN representations of the ephemeris data lines, keywords are not used. Rather, the components of the ephemeris data line appear in an order defined by the specific ephemeris data line type. In the XML representation, the tags described are fundamental to the format.

###  DISCUSSION

This non-normative subsection discusses and provides examples of the use of quaternion tags in the AEM.

The XML representations of quaternions in the ADM constituent messages share a common quaternion definition. However, there are some differences in those definitions in the underlying KVN definitions of the APM and AEM. As in the KVN representation of the quaternion, it is possible to code the tags for the individual components of the quaternion (Q1, Q2, Q3, QC) in either of the standard orders (i.e., scalar component first or last). The following examples are meant to illustrate the standard for representing quaternions in the AEM.

Here is an example AEM quaternion for a ‘QUATERNION’ ephemeris data line:

 <attitudeState>

 <quaternionState>

 <EPOCH>2004-100T00:00:00</EPOCH>

 <quaternion>

 <Q1>0.00005</Q1>

 <Q2>0.87543</Q2>

 <Q3>0.40949</Q3>

 <QC>0.25678</QC>

 </quaternion>

 </quaternionState>

 </attitudeState>

Here is an example AEM quaternion for a ‘QUATERNION/DERIVATIVE’ ephemeris data line:

 <attitudeState>

 <quaternionDerivative>

 <EPOCH>2004-100T00:00:00</EPOCH>

 <quaternion>

 <Q1>0.00005</Q1>

 <Q2>0.87543</Q2>

 <Q3>0.40949</Q3>

 <QC>0.25678</QC>

 </quaternion>

 <quaternionDerivative>

 <Q1\_DOT>0.002</Q1\_DOT>

 <Q2\_DOT>0.003</Q2\_DOT>

 <Q3\_DOT>0.004</Q3\_DOT>

 <QC\_DOT>0.001</QC\_DOT>

 </quaternionDerivative>

 </quaternionDerivative>

 </attitudeState>

## CREATING AN ACM INSTANTIATION

### SECTION FORTHCOMING AS THE ACM IS DEFINED... NOTHING TO ADD AT THIS TIME.

# ATTITUDE DATA MessageS KVN syntax

# ATTITUDE DATA MessageS XML syntax

NOTE: Could be put in Section 7 also (i.e., one syntax section, divided between KVN and XML.

## OVERVIEW

### XML instantiations of an ADM shall observe the syntax described in this chapter.

## ADM LINES IN XML

### Each ADM file shall consist of a set of ADM lines. Each ADM line shall be one of the following:

* XML version line;
* an XML-formatted line; or
* a blank line.

### Each ADM line must not exceed 254 ASCII characters and spaces (excluding line termination character[s]). [Question: Different limit for ACM?]

### Only printable ASCII characters and blanks shall be used. Control characters (such as TAB, etc.) shall not be used, with the exception of the line termination characters specified below. [Note: Frank has recommended as statement/requirement regarding XML "escaped" characters such as &amp; &lt; &gt; &quot; &apos;, but it is not yet clear what the statement/requirement should be.]

### Blank lines may be used at any position within the file. Blank lines shall have no assignable meaning, and may be ignored.

### All lines shall be terminated by a single Carriage Return or a single Line Feed, or a Carriage Return/Line Feed pair or a Line Feed/Carriage Return pair.

## VALUES IN THE ADM/XML

### Each mandatory XML tag must be present and contain a valid value.

### Integer values shall follow the conventions of the integer data type per Reference [reference here to XML datatypes document [9] (see above)]. Additional restrictions on the allowable range of values permitted for any integer data element may also be defined in the ADM XML Schema.

NOTE – Examples of such restrictions may include a defined range (e.g., 0 - 100, 1 - 10, etc.), a set of enumerated values (e.g., 0,1,2,4,8), a pre-defined specific variation such as positiveInteger, or a user-defined data type variation.

### Non-integer numeric values may be expressed in either fixed-point or floating-point notation. Numeric values shall follow the conventions of the double data type per Reference [reference here to XML datatypes document [9] (see above)]. Additional restrictions on the allowable range of values permitted for any numeric data element may also be defined in the ADM XML Schema.

NOTE – Examples of such restrictions may include a defined range (e.g., 0.0-100.0, etc.), or a user-defined data type variation.

### Text values shall follow the conventions of the string data type per Reference [reference here to XML datatypes document [9] (see above)]. Additional restrictions on the allowable range or values permitted for any data element may also be defined in the ADM XML Schema.

NOTE – Examples of such restrictions may include a set of enumerated values (e.g., ‘YES’/‘NO’) or other user-defined data type variation.

### Text values in ADM/XML instantiations (i.e., the values between the opening and closing tags), shall consist of either all uppercase or all lowercase characters; an exception is made for values between the <COMMENT> and </COMMENT> tags, which may be in any case desired by the user. Otherwise, instantiations shall not mix uppercase and lowercase characters in values.

### In value fields that represent a time tag, values shall follow the conventions of the ndm:epochType data type used in all CCSDS NDM/XML schemas.

## UNITS IN THE ADM/XML

The units in the ADM/XML shall be the same units used in the KVN-formatted ADM described in Sections 3 through 5. XML attributes shall be used to explicitly define the units or other important information associated with the given data element (see Annex C for examples).

## COMMENTS IN ADM/XML

Comments are optional and must be displayed as values between the <COMMENT> and </COMMENT> tags. Comments may be in any case desired by the user.

**NOTE to Alain: Examples forthcoming...**