

Subj: MESSAGE TEXT FORMAT XML SCHEMA RE-FACTOR

1. Background

a. The current XML Schema for US and NATO MTF are designed for piecemeal implementation of messages based on commonly defined XML nodes which are maintained in relational databases. This makes normalization, re-use, and configuration management difficult. Because the current XML Schema design does not support implementation of the entire standard, there are persistent mismatches of messages implemented and versions across systems.

b. The re-factor of these standards to a Garden of Eden XML Schema Design model allows provisioning of consolidated XML Schema files with a manageable sizes to allow complete implementation of every message using a uniform and re-usable methodology. Alignment with the US National Information Exchange Model (NIEM) methodology will allow the development and re-use of standardized Information Exchange Product Documentation (IEPD) resources which will support valid implementations and promote interoperability with external specifications.

2. Goals

a. This project will use the Extensible Style-sheet Language for Transformation (XSLT) to convert existing XML Schema products to Garden of Eden design pattern, and will reduce file sizes by normalizing Types, using attributes in annotation elements, and applying fixed values where appropriate to reduce ambiguity and eliminate rules which specify required content. The resulting XML Schema products will support all requirements for the text based slash delimited MTF format, and will avoid alteration of XML element names except where absolutely necessary for naming de-confliction of global nodes. Except for a few cases, XML instances from the current standard will validate against the re-factored XML Schema if common name-spaces are assigned.

b. This project is intended to support holistic implementation of the standards in order to facilitate use in web services, allow conversion between US and NATO specification, conversion between message versions, and interoperability with other standard formats.

3. USMTF XML Schema Design. Adjustments are made to the format and content of all XML Schema in order to reduce size, and eliminate redundant or unnecessary information.

4. Process. The process for converting from current standard formats is provided for purposes of integrity, testing and verification. For most use cases, the generated XML Schema can be used without repeating this conversion. Reference implementation products are intended for re-use and distribution in order to promote interoperability and uniformity in implementations. These products should load the provided re-factored XML Schema and provide intended MTF messaging functionality for every message and message component.

a. Normalization Rationale

(1) The creation of a "Type" for every field in MTF simply by adding "Type" or "SimpleType" to each field name does not leverage the concept of Typing appropriately. Many fields have identical content restrictions. By comparing regular expressions, numerical content, and enumerations the number of Simple Types are significantly reduced and the XML implementation of the MIL STDs employs reuse, extension and restriction effectively.

(3) Base Types are implemented at the field level to serve the requirement to be able to add security markings at the field level. The Baseline XML Schema accomplish this at the Set level which requires extension for every field each time it is used. The implementation of FieldStringBaseType, Field EnumerationBaseType, FieldIntegerBaseType, and FieldIntegerBaseType allow fields to be re-used by referencing instead of extension except where they have the nillable attribute. This significantly reduces the size of the XML Schema files.

5. Annotations

(1) Content of annotations is converted from elements to attributes in order to reduce size caused by closing tags. Names are changed to plain language terms instead of database field names. Elements are used for Examples and Documents which have multiple items. When documentation and appinfo items are duplicative, the information is retained in the documentation node and removed from the appinfo node. Re-formatted appinfo nodes are demonstrated in Tables 1-4.

# MESSSAGE TEXT FORMAT XML SCHEMA REFACTOR

<b>Baseline Message Elements:</b> <appinfo> <MtfName/> <MtfIdentifier/> <MtfIndexReferenceNumber/> <MtfSponsor/> <MtfPurpose/> <MtfRelatedDocument/> <MtfRemark/> <MtfNote/> <MtfStructuralRelationship/> <VersionIndicator/> <AlternativeType/> <SetFormatPositionName/> <SetFormatPositionNumber/> <SetFormatPositionConcept/> <SetFormatPositionUseDescription/> <OccurrenceCategory/> <Repeatability/> <SegmentStructureName/> <SegmentStructureConcept/> <SegmentStructureUseDescription/> </appinfo> <b>Ommitted Message Elements:</b> <appinfo> <MtfIndexReferenceNumber/> <MtfStructuralRelationship/> <AlternativeType/> <OccurrenceCategory/> <Repeatability/> <InitialSetFormatPosition/> </appinfo>	<b>Ommitted Baseline Field Elements:</b> <appinfo> <FieldFormatIndexReferenceNumber/> <FudNumber/> <MinimumLength/> <MaximumLength/> <LengthLimitation/> <UnitOfMeasure/> <Type/> <EntryType/> <DataType/> <MtfRegularExpression/> <MinimumInclusiveValue/> <MaximumInclusiveValue/> <LengthVariable/> <MinimumDecimalPlaces/> <MaximumDecimalPlaces/> <ElementalFfirmFudnSequence/> </appinfo> <b>Ommitted Baseline Enumeration Elements:</b> <appinfo> <EntryType/> <DataType/> <Explanation/> <DataItem/> </appinfo>	<b>Refactor Message Elements:</b> <appinfo> <Msg name="MtfName" identifier="MtfIdentifier" sponsor="MtfSponsor" purpose="MtfPurpose" remark="MtfRemark" note="MtfNote" version="VersionIndicator"> <Document>MtfRelatedDocument1</Document> <Document>MtfRelatedDocument2</Document> </Msg> </appinfo> <b>Refactor Segment Elements:</b> <appinfo> <Segment name="SegmentStructureName" concept="SegmentStructureConcept" usage="SegmentStructureUseDescription"/> </appinfo> <b>Refactor Set Elements:</b> <appinfo> <Set positionName="SetFormatPositionName" position="SetFormatPositionNumber" concept="SetFormatPositionConcept" usage="SetFormatPositionUseDescription"/> </appinfo>
--	--	---

Table 1: Re-factor of appinfo items for Messages.

<b>Baseline Set Elements:</b> <appinfo> <SetFormatName/> <SetFormatIdentifier/> <ColumnarIndicator/> <GroupOffFieldsIndicator/> <RepeatabilityForGroupOffFields/> <SetFormatNote/> <SetFormatExample/> <SetFormatRemark/> <SetFormatSponsor/> <SetFormatRelatedDocuments/> <VersionIndicator/> <FieldFormatPositionName/> <FieldFormatPositionNumber/> <OccurrenceCategory/> <FieldFormatPositionConcept/> <ColumnHeader/> <AlphabeticIdentifier/> <Justification/> <FieldDescriptor/> <AssignedFfirmFudUseDescription/> <FieldFormatIndexReferenceNumber/> <FieldFormatName/> <FieldFormatStructure/> <FieldFormatDefinition/> <FieldFormatRemark/> <FieldFormatRelatedDocument/> <FieldFormatSponsor/> </appinfo> <b>Ommitted Baseline Field Elements:</b> <appinfo> <GroupOffFieldsIndicator/> <RepeatabilityForGroupOffFields/> <OccurrenceCategory/> <FieldFormatIndexReferenceNumber/> <FieldFormatStructure/> </appinfo>	<b>Refactor Field Elements in Sets.xsd:</b> <appinfo> <Field name="FieldFormatName or FieldDescriptor" positionName="FieldFormatPositionName" position="FieldFormatPositionNumber" concept="FieldFormatPositionConcept" columnHeader="ColumnHeader" identifier="AlphabeticIdentifier" alignment="Justification" usage="AssignedFfirmFudUseDescription" remark="FieldFormatRemark" sponsor="FieldFormatSponsor"> <Example>SetFormatExample1</Example> <Example>SetFormatExample2</Example> <Document>FieldFormatRelatedDocument1</Document> <Document>FieldFormatRelatedDocument2</Document> </Field> </appinfo> <b>Refactor Set Elements:</b> <appinfo> <Set name="SetFormatName" id="SetFormatIdentifier" column="ColumnarIndicator" note="SetFormatNote" remark="SetFormatRemark" sponsor="SetFormatSponsor" version="VersionIndicator"> <Example>SetFormatExample1</Example> <Example>SetFormatExample2</Example> <Document>SetFormatRelatedDocuments1</Document> <Document>SetFormatRelatedDocuments2</Document> </Set> </appinfo>
--	--

Table 2: Re-factor of appinfo items for Sets.

# MESSSAGE TEXT FORMAT XML SCHEMA REFACTOR

<b>Baseline Field Elements:</b> <pre> &lt;appinfo&gt;   &lt;FudName/&gt;   &lt;FudExplanation/&gt;   &lt;FieldFormatIndexReferenceNumber/&gt;   &lt;FudNumber/&gt;   &lt;VersionIndicator/&gt;   &lt;MinimumLength/&gt;   &lt;MaximumLength/&gt;   &lt;LengthLimitation/&gt;   &lt;UnitOfMeasure/&gt;   &lt;Type/&gt;   &lt;FudSponsor/&gt;   &lt;FudRelatedDocument/&gt;   &lt;EntryType/&gt;   &lt;DataType/&gt;   &lt;Explanation/&gt;   &lt;MtfRegularExpression/&gt;   &lt;MinimumInclusiveValue/&gt;   &lt;MaximumInclusiveValue/&gt;   &lt;LengthVariable/&gt;   &lt;MinimumDecimalPlaces/&gt;   &lt;MaximumDecimalPlaces/&gt;   &lt;ElementalFfirmFudnSequence/&gt; &lt;/appinfo&gt; </pre>	<b>Ommitted Baseline Field Elements:</b> <pre> &lt;appinfo&gt;   &lt;FieldFormatIndexReferenceNumber/&gt;   &lt;FudNumber/&gt;   &lt;MinimumLength/&gt;   &lt;MaximumLength/&gt;   &lt;LengthLimitation/&gt;   &lt;UnitOfMeasure/&gt;   &lt;Type/&gt;   &lt;EntryType/&gt;   &lt;DataType/&gt;   &lt;MtfRegularExpression/&gt;   &lt;MinimumInclusiveValue/&gt;   &lt;MaximumInclusiveValue/&gt;   &lt;LengthVariable/&gt;   &lt;MinimumDecimalPlaces/&gt;   &lt;MaximumDecimalPlaces/&gt;   &lt;ElementalFfirmFudnSequence/&gt; &lt;/appinfo&gt; </pre>	<b>Baseline Enumeration Elements:</b> <pre> &lt;appinfo&gt;   &lt;EntryType/&gt;   &lt;DataType/&gt;   &lt;Explanation/&gt;   &lt;DataCode/&gt;   &lt;DataItem/&gt;   &lt;DataItemSequenceNumber/&gt;   &lt;DataItemSponsor/&gt; &lt;/appinfo&gt; </pre> <b>Ommitted Baseline Enumeration Elements:</b> <pre> &lt;appinfo&gt;   &lt;EntryType/&gt;   &lt;DataType/&gt;   &lt;Explanation/&gt;   &lt;DataItem/&gt; &lt;/appinfo&gt; </pre> <b>Refactor Field Elements:</b> <pre> &lt;appinfo&gt;   &lt;Field name="FudName"     explanation="FudExplanation or Explanation"     version="VersionIndicator"     sponsor="FudSponsor"/&gt;   &lt;Document&gt;FudRelatedDocument1&lt;/Document&gt;   &lt;Document&gt;FudRelatedDocument2&lt;/Document&gt; &lt;/appinfo&gt; </pre> <b>Refactor Enumeration Elements:</b> <pre> &lt;appinfo&gt;   &lt;Enum name="DataItem"/&gt; &lt;/appinfo&gt; </pre>
--	---	---

Table 3: Re-factor of appinfo items for Fields.

## a. Fields Re-factor

(1) **Field Base Types.** The requirement to apply security markings to every field is currently accomplished by extending Simple Type fields at the Set level to add the ICM attribute group. This reduces the ability to reference fields at the Set level and causes unnecessary repetition in the XML Schema design. The Re-Factor approach is to create FieldBaseTypes which carry the ICM attribute group and can be used to add other field level extensions. All fields are provided as Complexity which extend the base types.

<pre> &lt;complexType name="FieldStringBaseType"&gt;   &lt;simpleContent&gt;     &lt;extension base="xsd:string"&gt;       &lt;attributeGroup ref="ism:SecurityAttributesOptionGroup"/&gt;     &lt;/extension&gt;   &lt;/simpleContent&gt; &lt;/complexType&gt; </pre>	<pre> &lt;complexType name="FieldEnumeratedBaseType"&gt;   &lt;simpleContent&gt;     &lt;extension base="xsd:string"&gt;       &lt;attributeGroup ref="ism:SecurityAttributesOptionGroup"/&gt;     &lt;/extension&gt;   &lt;/simpleContent&gt; &lt;/complexType&gt; </pre>
<pre> &lt;complexType name="FieldIntegerBaseType"&gt;   &lt;simpleContent&gt;     &lt;extension base="xsd:integer"&gt;       &lt;attributeGroup ref="ism:SecurityAttributesOptionGroup"/&gt;     &lt;/extension&gt;   &lt;/simpleContent&gt; &lt;/complexType&gt; </pre>	<pre> &lt;complexType name="FieldDecimalBaseType"&gt;   &lt;simpleContent&gt;     &lt;extension base="xsd:decimal"&gt;       &lt;attributeGroup ref="ism:SecurityAttributesOptionGroup"/&gt;     &lt;/extension&gt;   &lt;/simpleContent&gt; &lt;/complexType&gt; </pre>

Table 4: Field Base Types.

(2) **Strings.** By comparing Regular Expressions the number of types extended by FieldStringBase Type is reduced. Types which include length and value restrictions in the Regular Expression are removed. String types are defined using pattern for content, and XML Schema elements to specify lengths at the element level. The provided example "AlphaNumericBlankSpecialTextType" is re-used 877 times in the USMTF GoE\_fields.xsd schema.

## MESSSAGE TEXT FORMAT XML SCHEMA REFACTOR

<p><b>Baseline String Field Global Types:</b></p> <pre> &lt;simpleType name="StringElementType"&gt;   &lt;annotation/&gt;   &lt;restriction base="string"&gt;     &lt;annotation/&gt;     &lt;minLength value="1"/&gt;     &lt;maxLength value="8"/&gt;     &lt;pattern value="[\-\.,\(\)\?A-Z0-9&amp;#x20;]{1,8}"/&gt;   &lt;/restriction&gt; &lt;/simpleType&gt; &lt;element name="String" type="StringElementType"/&gt; </pre>	<p><b>Refactor String Field Global Types:</b></p> <pre> &lt;complexType name="AlphaNumericBlankSpecialTextType"&gt;   &lt;simpleContent&gt;     &lt;restriction base="FieldStringBaseType"&gt;       &lt;pattern value="[\-\.,\(\)\?A-Z0-9 ]+"/&gt;     &lt;/restriction&gt;   &lt;/simpleContent&gt; &lt;/complexType&gt;  <b>Refactor String Field Global Elements:</b>  &lt;element name="String"&gt;   &lt;complexType&gt;     &lt;simpleContent&gt;       &lt;restriction base="AlphaNumericBlankSpecialTextType"&gt;         &lt;minLength value="1"/&gt;         &lt;maxLength value="8"/&gt;       &lt;/restriction&gt;     &lt;/simpleContent&gt;   &lt;/complexType&gt; &lt;/element&gt; </pre>
---	---

Table 5: Re-factor of String Fields

(3) Enumerations. Enumerations do not require Regular Expressions or types. the XML Schema integer is extended by FieldIntegerBaseType to include security markings.

<p><b>Baseline Enumerated Field Global Types:</b></p> <pre> &lt;simpleType name="EnumerationElementType"&gt;   &lt;restriction base="string"&gt;     &lt;enumeration value="YES"/&gt;     &lt;enumeration value="NO"/&gt;   &lt;/restriction&gt; &lt;/simpleType&gt;  &lt;element name="EnumerationElement" type="YesNoType"/&gt; </pre>	<p><b>Refactor Enumerated Field Global Types:</b></p> <pre> &lt;complexType name="YesNoType"&gt;   &lt;simpleContent&gt;     &lt;restriction base="FieldEnumeratedBaseType"&gt;       &lt;enumeration value="YES"/&gt;       &lt;enumeration value="NO"/&gt;     &lt;/restriction&gt;   &lt;/simpleContent&gt; &lt;/complexType&gt;  &lt;element name="EnumerationElement" type="YesNoType"/&gt; </pre>
--	---

Table 5: Re-Factor of Enumerated String Fields

(4) Integers. Numeric types do not require Regular Expressions. The XML Schema decimal type is extended by FieldIntegerBaseType to include security markings, and each Integer field restricts FieldIntegerBaseType to specify length and value parameters.

<p><b>Baseline Integer Field Global Types and Elements:</b></p> <pre> &lt;simpleType name="IntegerElementType"&gt;   &lt;restriction base="xsd:integer"&gt;     &lt;minInclusive value="-9999"/&gt;     &lt;maxInclusive value="99999"/&gt;     &lt;pattern value="[-0-9]{5}"/&gt;   &lt;/restriction&gt; &lt;/simpleType&gt;  &lt;element name="IntegerElement" type="IntegerElementType"/&gt; </pre>	<p><b>Baseline Integer Field Global Elements:</b></p> <pre> &lt;element name="IntegerElement"&gt;   &lt;complexType&gt;     &lt;simpleContent&gt;       &lt;restriction base="FieldIntegerBaseType"&gt;         &lt;minInclusive value="-9999"/&gt;         &lt;maxInclusive value="99999"/&gt;       &lt;/restriction&gt;     &lt;/simpleContent&gt;   &lt;/complexType&gt; &lt;/element&gt; </pre>
--	--

Table 5: Re-factor of Integer Fields

(5) Decimals. Numeric types do not require Regular Expressions. The XML Schema decimal type is extended by FieldDecimalBaseType to include security markings, and each Decimal field restricts FieldDecimalBaseType to specify length and value parameters.

# MESSSAGE TEXT FORMAT XML SCHEMA REFACTOR

<p><b>Baseline Decimal Field Global Elements:</b></p> <pre> &lt;simpleType name="DepthInFeet"&gt;   &lt;restriction base="xsd:decimal"&gt;     &lt;minInclusive value=".1"/&gt;     &lt;maxInclusive value="9999.9"/&gt;     &lt;pattern value="[0-9]{0,4}\.[0-9]{1}" /&gt;   &lt;/restriction&gt; &lt;/simpleType&gt; &lt;/element&gt; </pre>	<p><b>Baseline Decimal Field Global Elements:</b></p> <pre> &lt;element name="DepthInFeet"&gt;   &lt;complexType&gt;     &lt;simpleContent&gt;       &lt;restriction base="FieldDecimalBaseType"&gt;         &lt;minInclusive value=".1"/&gt;         &lt;maxInclusive value="9999.9"/&gt;         &lt;fractionDigits value="1"/&gt;         &lt;totalDigits value="5"/&gt;       &lt;/restriction&gt;     &lt;/simpleContent&gt;   &lt;/complexType&gt; &lt;/element&gt; </pre>
--	--

Table 7: Re-Factor of Decimal Fields

- b. Set Re-Factor. Sets are appended with 'Set' in order to prevent element name collisions and allow global use.

<p><b>Baseline Set Global Types and Elements:</b></p> <pre> &lt;complexType name="WeatherEventMovementType"&gt;   &lt;complexContent&gt;     &lt;extension base="SetBaseType"&gt;       &lt;sequence&gt;         &lt;element name="DirectionOfWeatherEventMovement"           minOccurs="1" maxOccurs="1" nillable="true"&gt;           &lt;complexType&gt;             &lt;simpleContent&gt;               &lt;extension base="f:DirectionOfWeatherEventMovementType"&gt;                 &lt;attribute name="ffSeq" type="unsignedShort" fixed="1"/&gt;                 &lt;attribute name="ffirmFudn" type="string" fixed="FF493-44"/&gt;                 &lt;attributeGroup ref="ism:SecurityAttributesOptionGroup"/&gt;               &lt;/extension&gt;             &lt;/simpleContent&gt;           &lt;/complexType&gt;         &lt;/element&gt;         &lt;element name="SpeedOfWeatherEventMovement"           minOccurs="0" maxOccurs="1"&gt;           &lt;complexType&gt;             &lt;complexContent&gt;               &lt;extension base="c:SpeedOfWeatherEventMovementType"&gt;                 &lt;attribute name="ffSeq" type="unsignedShort" fixed="2"/&gt;                 &lt;attribute name="ffirmFudn" type="string" fixed="FF357-1"/&gt;                 &lt;attributeGroup ref="ism:SecurityAttributesOptionGroup"/&gt;               &lt;/extension&gt;             &lt;/complexContent&gt;           &lt;/complexType&gt;         &lt;/element&gt;         &lt;element name="DistanceTraveledByWeatherEvent"           minOccurs="0" maxOccurs="1"&gt;           &lt;complexType&gt;             &lt;complexContent&gt;               &lt;extension base="c:MovementOfWeatherEventType"&gt;                 &lt;attribute name="ffSeq" type="unsignedShort" fixed="3"/&gt;                 &lt;attribute name="ffirmFudn" type="string" fixed="FF388-1"/&gt;                 &lt;attributeGroup ref="ism:SecurityAttributesOptionGroup"/&gt;               &lt;/extension&gt;             &lt;/complexContent&gt;           &lt;/complexType&gt;         &lt;/element&gt;       &lt;/sequence&gt;       &lt;attribute name="setid" type="string" fixed="MVMT"/&gt;     &lt;/extension&gt;   &lt;/complexContent&gt; &lt;/complexType&gt; </pre>	<p><b>Refactor Set Field Global Elements:</b></p> <pre> &lt;complexType name="WeatherEventMovementSetType"&gt;   &lt;complexContent&gt;     &lt;extension base="SetBaseType"&gt;       &lt;sequence&gt;         &lt;element name="DirectionOfWeatherEventMovement"           minOccurs="1" maxOccurs="1" nillable="true"&gt;           &lt;complexType&gt;             &lt;simpleContent&gt;               &lt;restriction base="field:FieldIntegerBaseType"&gt;                 &lt;minInclusive value="000"/&gt;                 &lt;maxInclusive value="359"/&gt;               &lt;/restriction&gt;             &lt;/simpleContent&gt;           &lt;/complexType&gt;         &lt;/element&gt;         &lt;element ref="field:SpeedOfWeatherEventMovement"/&gt;         &lt;element name="DistanceTraveledByWeatherEvent"           type="field:MovementOfWeatherEventType"/&gt;       &lt;/sequence&gt;       &lt;attribute name="setid" type="string" fixed="MVMT"/&gt;     &lt;/extension&gt;   &lt;/complexContent&gt; &lt;/complexType&gt; </pre>
--	--

Table 8: Re-Factor of Sets

## MESSSAGE TEXT FORMAT XML SCHEMA REFACTOR

- c. **Segments Re-factor.** Segments are extracted from messages to provide the opportunity for re-use. All Segments are extended with the SegmentBaseType which is used to add ISM attribute group at the Segment level.

Baseline Segments Global Types and Elements:	Refactor Segments Global Types:
<pre> &lt;element name="ReportedUnitLocationSegment" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;complexType&gt;     &lt;sequence&gt;       &lt;element name="OrderLocationType" minOccurs="1" maxOccurs="1"&gt;         &lt;complexType&gt;           &lt;complexContent&gt;             &lt;extension base="s:OrderLocationTypeType"&gt;               &lt;attribute name="setSeq" type="unsignedShort" fixed="51"/&gt;             &lt;/extension&gt;           &lt;/complexContent&gt;         &lt;/complexType&gt;       &lt;/element&gt;     &lt;/sequence&gt;     &lt;attribute name="segSeq" type="unsignedShort" fixed="51"/&gt;   &lt;/complexType&gt; &lt;/element&gt; </pre>	<pre> &lt;complexType name="ReportedUnitLocationSegmentType"&gt;   &lt;complexContent&gt;     &lt;extension base="SegmentBaseType"&gt;       &lt;sequence&gt;         &lt;element ref="set:OrderLocationSetType"/&gt;         &lt;element ref="set:Area Set" /&gt;         &lt;element ref="set:GridLocation Set" /&gt;         &lt;element ref="set:Line"/&gt;         &lt;element name="TimeOfReportedUnitLocation" minOccurs="0" maxOccurs="1" type="set:TimeSetType"/&gt;       &lt;/sequence&gt;     &lt;/extension&gt;   &lt;/complexContent&gt; &lt;/complexType&gt; </pre>

Table 9: Re-Factor of Segments

- d. **Messages Re-factor**

- (1) **General Text Re-factor.** This proposed change adds descriptive field names which are specific to the required content and eliminate the need to distinguish XML nodes by appending numbers. This will affect XML instances and will require mitigation with Baseine XML instances.

Baseline Message GeneralText:	Refactor GeneralText
<pre> &lt;element name="GeneralText_1" minOccurs="0" maxOccurs="1"&gt;   &lt;xsd:annotation&gt;     &lt;xsd:documentation&gt;The GENTEXT/SITUATION set describes the overall situation. It may be appropriate to simply make reference to the command's current operations/exercise order/plan.&lt;/xsd:documentation&gt;   &lt;xsd:appinfo&gt;     &lt;SetFormatPositionName&gt;GENERAL TEXT&lt;/SetFormatPositionName&gt;     &lt;SetFormatPositionNumber&gt;7&lt;/SetFormatPositionNumber&gt;     &lt;SetFormatPositionConcept&gt;The GENTEXT/SITUATION set describes the overall situation. It may be appropriate to simply make reference to the command's current operations/exercise order/plan.   &lt;/SetFormatPositionConcept&gt;     &lt;SetFormatPositionUseDescription&gt;Field 1 of the GENTEXT set must equal "SITUATION".&lt;/SetFormatPositionUseDescription&gt;     &lt;OccurrenceCategory&gt;Operationally Determined&lt;/OccurrenceCategory&gt;     &lt;Repeatability&gt;1&lt;/Repeatability&gt;   &lt;/xsd:appinfo&gt; &lt;/xsd:annotation&gt;   &lt;complexType&gt;     &lt;complexContent&gt;       &lt;extension base="s:GeneralTextType"&gt;         &lt;attribute name="setSeq" type="xsd:unsignedShort" fixed="24" /&gt;         &lt;attributeGroup ref="ism:SecurityAttributesGroup" /&gt;       &lt;/extension&gt;     &lt;/complexContent&gt;   &lt;/complexType&gt; &lt;/element&gt; &lt;element name="GeneralText_2" minOccurs="0" maxOccurs="1"&gt;   &lt;complexType&gt;     &lt;complexContent&gt;       &lt;extension base="s:GeneralTextType"&gt;         &lt;attribute name="setSeq" type="xsd:unsignedShort" fixed="25" /&gt;         &lt;attributeGroup ref="ism:SecurityAttributesGroup" /&gt;       &lt;/extension&gt;     &lt;/complexContent&gt;   &lt;/complexType&gt; &lt;/element&gt; </pre>	<pre> &lt;element name="GenTextSituation"&gt;   &lt;complexType&gt;     &lt;complexContent&gt;       &lt;extension base="set:GeneralTextSetType"&gt;         &lt;sequence&gt;           &lt;element name="GentextTextIndicator" type="field:AlphaNumericBlankSpecialTextType" minOccurs="1" fixed="SITUATION"/&gt;           &lt;element ref="field:FreeTextField" minOccurs="1"/&gt;         &lt;/sequence&gt;       &lt;/extension&gt;     &lt;/complexContent&gt;   &lt;/complexType&gt; &lt;/element&gt; &lt;element name="GenTextMission"&gt;   &lt;complexType&gt;     &lt;complexContent&gt;       &lt;extension base="set:GeneralTextSetType"&gt;         &lt;sequence&gt;           &lt;element name="GentextTextIndicator" type="field:AlphaNumericBlankSpecialTextType" minOccurs="1" fixed="MISSION"/&gt;           &lt;element ref="field:FreeTextField" minOccurs="1"/&gt;         &lt;/sequence&gt;       &lt;/extension&gt;     &lt;/complexContent&gt;   &lt;/complexType&gt; &lt;/element&gt; </pre>

## MESSSAGE TEXT FORMAT XML SCHEMA REFACTOR

(2) Heading Information Re-factor. This proposed change adds descriptive field names which are specific to the required content and eliminate the need to distinguish XML nodes by appending numbers. This will affect XML instances and will require mitigation with Baseline XML instances.

Baseline HeadingInformation:	Refactor HeadingInformation
<pre> &lt;xsd:element name="HeadingInformation_1" minOccurs="1" maxOccurs="1"&gt;   &lt;xsd:annotation&gt;     &lt;xsd:documentation&gt;The HEADING/RATIONS set&lt;/xsd:documentation&gt;   &lt;/xsd:annotation&gt;   &lt;xsd:appinfo&gt;     &lt;SetFormatPositionName&gt;HEADING INFORMATION     &lt;/SetFormatPositionName&gt;     &lt;SetFormatPositionUseDescription&gt;Field 1 of the HEADING     set must equal "RATIONS".&lt;/SetFormatPositionUseDescription&gt;     &lt;OccurrenceCategory&gt;Mandatory&lt;/OccurrenceCategory&gt;     &lt;Repeatability&gt;1&lt;/Repeatability&gt;   &lt;/xsd:appinfo&gt; &lt;/xsd:element&gt;   &lt;xsd:complexType&gt;     &lt;xsd:complexContent&gt;       &lt;xsd:extension base="s:HeadingInformationType"&gt;         &lt;xsd:attribute name="setSeq" type="xsd:unsignedShort" fixed="10" /&gt;         &lt;xsd:attributeGroup ref="ism:SecurityAttributesGroup" /&gt;       &lt;/xsd:extension&gt;     &lt;/xsd:complexContent&gt;   &lt;/xsd:complexType&gt; &lt;/xsd:element&gt; </pre>	<pre> &lt;element name="HeadingInformationRations"&gt;   &lt;complexType&gt;     &lt;complexContent&gt;       &lt;extension base="set:HeadingInformationSetType"&gt;         &lt;sequence&gt;           &lt;element name="FieldAssignment"             type="field:AlphaNumericBlankSpecialTextType"             minOccurs="1"             fixed="RATIONS"/&gt;           &lt;element ref="field:FreeTextField" minOccurs="1"/&gt;         &lt;/sequence&gt;       &lt;/extension&gt;     &lt;/complexContent&gt;   &lt;/complexType&gt; &lt;/element&gt; </pre>

Table 11: Re-Factor of HeadingInformation

(3) Message Identifier Re-factor. This proposed change adds fixed values to the Message XML Schema in order to allow validation of Standard, MessageTextFormatIdentifier, and VersionOfMessageFormat using XML validation instead of requiring rules.

Baseline MessageIdentifier:	Refactor MessageIdentifier
<pre> &lt;complexType name="MessageIdentifierType"&gt;   &lt;complexContent&gt;     &lt;extension base="SetBaseType"&gt;       &lt;sequence&gt;         &lt;element name="StandardOfMessageTextFormat"           minOccurs="1" maxOccurs="1" nillable="true"&gt;           &lt;complexType&gt;             &lt;simpleContent&gt;               &lt;extension base="f:StandardOfMessageTextFormatType"&gt;                 &lt;attribute name="ffSeq" type="unsignedShort" fixed="1"/&gt;                 &lt;attribute name="ffirmFudn" type="string" fixed="FF180-2"/&gt;                 &lt;attributeGroup ref="ism:SecurityAttributesOptionGroup"/&gt;               &lt;/extension&gt;             &lt;/simpleContent&gt;           &lt;/complexType&gt;         &lt;/element&gt;         &lt;element name="VersionOfMessageTextFormat"           minOccurs="1" maxOccurs="1" nillable="true"&gt;           &lt;complexType&gt;             &lt;simpleContent&gt;               &lt;extension base="f:VersionOfMessageTextFormatType"&gt;                 &lt;attribute name="ffSeq" type="unsignedShort" fixed="2"/&gt;                 &lt;attribute name="ffirmFudn" type="string" fixed="FF180-3"/&gt;                 &lt;attributeGroup ref="ism:SecurityAttributesOptionGroup"/&gt;               &lt;/extension&gt;             &lt;/simpleContent&gt;           &lt;/complexType&gt;         &lt;/element&gt;       &lt;/sequence&gt;     &lt;/extension&gt;   &lt;/complexContent&gt; &lt;/complexType&gt; </pre>	<pre> &lt;element name="MessageIdentifier" minOccurs="1" maxOccurs="1"&gt;   &lt;complexType&gt;     &lt;complexContent&gt;       &lt;extension base="set:SetBaseType"&gt;         &lt;sequence&gt;           &lt;element name="Standard"             type="field:AlphaNumericBlankSpecialInitDataLoadIDType"             minOccurs="1" maxOccurs="1"             nillable="true" fixed="MIL-STD-6040(SERIES)"&gt;           &lt;/element&gt;           &lt;element name="MessageTextFormatIdentifier"             type="field:AlphaNumericBlankSpecialTextType"             minOccurs="1" maxOccurs="1" nillable="true"             fixed="ABSTAT"&gt;           &lt;/element&gt;           &lt;element name="Version"             type="field:VersionOfMessageTextFormatType"             minOccurs="1" maxOccurs="1" nillable="true"             fixed="B.1.01.12"&gt;           &lt;/element&gt;         &lt;/sequence&gt;       &lt;/extension&gt;     &lt;/complexContent&gt;   &lt;/complexType&gt; &lt;/element&gt; </pre>

Table 11: Re-Factor of MessageIdentifier



## MESSAGE TEXT FORMAT XML SCHEMA REFACTOR

### 6. Summary.

a. String type normalization reduces the number of string Simple Types from 1624 to 153 without impacting message content.

b. Enumeration normalization reduces the number of enumerated Simple Types from 1766 to 1473 without impacting message content.

c. Numeric normalization eliminates 1337 integer Simple Types and 245 decimal Simple Types.

7. Process. This process is retained for purposes of verification, testing and maintenance. It is not necessary for implementers to repeat the effort.

a. XSLT Settings. All scripts are XSLT 2.0 and are set up to run with named templates using `xsl:document` nodes and `xsl:result-document` definitions for inputs and outputs.

a. The `NormalizedSimpleTypes.xsd` file was created using a variety of methods which analyze and compare Regular Expressions. This required subjective decisions which are beyond the scope of this document, and may be adjusted. The XSLT scripts to accomplish this are in the `USMTF/XSLT/Normalization/work` directory.

b. The XSLT scripts to generate the normalized Simple Types are located in the `USMTF/XSLT/Normalization` directory. Data products are located in the `USMTF/XSD/Normalized/work` directory.

c. The `Strings.xsl`, `Numerics.xsl`, and `Enumerations.xsl` scripts can be executed in any order. Results are written to the `USMTF/XSD/Normalized` directory.

d. The `GoE_Fields.xsl` script consolidates the `Strings.xsd`, `Integers.xsd`, `Decimals.xsd` and `Enumerations.xsd` files into the `GoE_field.xsd` document which is stored in the `USMTF/XSD/GoE_Schema` directory.

e. The `GoE_Sets.xsl` script refactors the baseline `sets.xsd` XML Schema to conform with proposed changes. It references the result of `GoE_Fields.xsl` so must be executed after generation of `GoE_field.xsd`.

f. The `GoE_Segments.xsl` script refactors the baseline `messages.xsd` XML Schema to extract `GoE_Segments.xsd` to conform with proposed changes. It references the result of `GoE_Fields.xsl` and `GoE_Sets.xsl` so must be executed after generation of `GoE_fields.xsd` and `GoE_sets.xsd`.

g. The `GoE_Messages.xsl` script refactors the baseline `messages.xsd` XML Schema to extract `GoE_Messages.xsd` to conform with proposed changes. It references the result of `GoE_Fields.xsl` and `GoE_Sets.xsl` so must be executed after generation of `GoE_fields.xsd`, `GoE_sets.xsd` and `GoE_segments.xsd`.

### 8. Conclusion.

a. This document represents a major XML Schema Design proposal for USMTF XML. The same proposal has been generated for NATO MTF XML Schema. Normalization and reduction of both XML standards is intended to support interoperability within and between MTF standards, as well as with external standards.

b. Both proposals should be considered with respect to verifiable accuracy and effective implementation support.