

# Schema documentation for eml-gbif-profile.xsd

november 18, 2024

## Table of Contents

Namespace: "" .....	4
Schema(s) .....	4
Main schema eml-gbif-profile.xsd .....	4
Element(s) .....	4
Element dataset .....	4
Element alternateIdentifier .....	7
Element dataset / shortName .....	8
Element title .....	8
Element dataset / creator .....	9
Element agentType / organizationName .....	11
Element individualName .....	11
Element individualName / salutation .....	12
Element individualName / givenName .....	12
Element individualName / surName .....	13
Element agentType / positionName .....	13
Element address .....	13
Element address / deliveryPoint .....	15
Element address / city .....	15
Element address / administrativeArea .....	15
Element address / postalCode .....	16
Element address / country .....	16
Element phone .....	16
Element electronicMailAddress .....	17
Element onlineUrl .....	17
Element userId .....	18
Element dataset / metadataProvider .....	18
Element associatedParty .....	20
Element role .....	22
Element pubDate .....	22
Element language .....	22
Element abstract .....	23
Element TextType / section .....	24
Element SectionType / title .....	25
Element SectionType / para .....	26
Element ParagraphType / value .....	28
Element ParagraphType / itemizedlist .....	29
Element ListType / listitem .....	29
Element ListType / listitem / para .....	30
Element ParagraphType / orderedlist .....	32
Element ParagraphType / emphasis .....	32
Element ParagraphType / emphasis / value .....	33
Element ParagraphType / subscript .....	34
Element SubSuperScriptType / value .....	35
Element SubSuperScriptType / subscript .....	36
Element SubSuperScriptType / superscript .....	37
Element ParagraphType / superscript .....	39
Element ParagraphType / literalLayout .....	40
Element ParagraphType / literalLayout / value .....	40
Element ParagraphType / ulink .....	41
Element ParagraphType / ulink / citetitle .....	42
Element ListType / listitem / itemizedlist .....	42
Element ListType / listitem / orderedlist .....	43
Element SectionType / section .....	43
Element TextType / para .....	44
Element keywordSet .....	46
Element keywordSet / keyword .....	47
Element keywordSet / keywordThesaurus .....	47
Element additionalInfo .....	47
Element para .....	48
Element ulink .....	48

Element ulink / citetitle .....	49
Element intellectualRights .....	50
Element licensed .....	50
Element licensed / licenseName .....	51
Element licensed / url .....	51
Element licensed / identifier .....	52
Element distribution .....	52
Element online .....	53
Element url .....	53
Element coverage .....	54
Element geographicCoverage .....	54
Element geographicCoverage / geographicDescription .....	55
Element geographicCoverage / boundingCoordinates .....	56
Element geographicCoverage / boundingCoordinates / westBoundingCoordinate .....	57
Element geographicCoverage / boundingCoordinates / eastBoundingCoordinate .....	57
Element geographicCoverage / boundingCoordinates / northBoundingCoordinate .....	57
Element geographicCoverage / boundingCoordinates / southBoundingCoordinate .....	57
Element temporalCoverage .....	58
Element temporalCoverage / rangeOfDates .....	59
Element temporalCoverage / rangeOfDates / beginDate .....	59
Element calendarDate .....	60
Element temporalCoverage / rangeOfDates / endDate .....	60
Element temporalCoverage / singleDateTime .....	61
Element taxonomicCoverage .....	61
Element taxonomicCoverage / generalTaxonomicCoverage .....	62
Element taxonomicCoverage / taxonomicClassification .....	62
Element taxonomicCoverage / taxonomicClassification / taxonRankName .....	63
Element taxonomicCoverage / taxonomicClassification / taxonRankValue .....	63
Element taxonomicCoverage / taxonomicClassification / commonName .....	64
Element purpose .....	64
Element introduction .....	65
Element gettingStarted .....	66
Element acknowledgements .....	68
Element dataset / maintenance .....	69
Element description .....	69
Element dataset / maintenance / maintenanceUpdateFrequency .....	70
Element dataset / maintenance / changeHistory .....	70
Element dataset / maintenance / changeHistory / changeScope .....	71
Element dataset / maintenance / changeHistory / oldValue .....	72
Element dataset / maintenance / changeHistory / changeDate .....	72
Element dataset / maintenance / changeHistory / comment .....	72
Element dataset / contact .....	73
Element dataset / publisher .....	75
Element methods .....	76
Element methods / methodStep .....	77
Element methods / sampling .....	77
Element methods / sampling / studyExtent .....	78
Element methods / sampling / samplingDescription .....	79
Element methods / qualityControl .....	79
Element project .....	80
Element projectType / personnel .....	81
Element projectType / funding .....	83
Element projectType / award .....	83
Element awardType / funderName .....	84
Element awardType / funderIdentifier .....	85
Element awardType / awardNumber .....	85
Element awardType / title .....	86
Element awardType / awardUrl .....	86
Element projectType / studyAreaDescription .....	87
Element descriptor .....	87
Element descriptorValue .....	88
Element projectType / designDescription .....	88
Element projectType / relatedProject .....	89
Element relatedProjectType / personnel .....	89
Element dataset / literatureCited .....	91
Element CitationListType / bibtex .....	91
Element additionalMetadata .....	92
Element additionalMetadata / metadata .....	94
Element additionalMetadata / metadata / gbif .....	96
Element additionalMetadata / metadata / gbif / dateStamp .....	98
Element additionalMetadata / metadata / gbif / hierarchyLevel .....	98
Element additionalMetadata / metadata / gbif / citation .....	99

Element additionalMetadata / metadata / gbif / bibliography .....	99
Element additionalMetadata / metadata / gbif / bibliography / citation .....	100
Element additionalMetadata / metadata / gbif / physical .....	100
Element objectName .....	101
Element characterEncoding .....	102
Element dataFormat .....	102
Element dataFormat / textFormat .....	106
Element dataFormat / textFormat / numHeaderLines .....	111
Element dataFormat / textFormat / numFooterLines .....	111
Element dataFormat / textFormat / recordDelimiter .....	112
Element dataFormat / textFormat / physicalLineDelimiter .....	112
Element dataFormat / textFormat / numPhysicalLinesPerRecord .....	113
Element dataFormat / textFormat / maxRecordLength .....	113
Element dataFormat / textFormat / attributeOrientation .....	114
Element dataFormat / textFormat / simpleDelimited .....	114
Element dataFormat / textFormat / simpleDelimited / fieldDelimiter .....	116
Element dataFormat / textFormat / simpleDelimited / collapseDelimiters .....	116
Element dataFormat / textFormat / simpleDelimited / quoteCharacter .....	117
Element dataFormat / textFormat / simpleDelimited / literalCharacter .....	118
Element dataFormat / textFormat / complex .....	118
Element dataFormat / textFormat / complex / textFixed .....	120
Element dataFormat / textFormat / complex / textFixed / fieldWidth .....	121
Element dataFormat / textFormat / complex / textFixed / lineNumber .....	122
Element dataFormat / textFormat / complex / textFixed / fieldStartColumn .....	122
Element dataFormat / textFormat / complex / textDelimited .....	123
Element dataFormat / textFormat / complex / textDelimited / fieldDelimiter .....	124
Element dataFormat / textFormat / complex / textDelimited / collapseDelimiters .....	125
Element dataFormat / textFormat / complex / textDelimited / lineNumber .....	126
Element dataFormat / textFormat / complex / textDelimited / quoteCharacter .....	126
Element dataFormat / textFormat / complex / textDelimited / literalCharacter .....	127
Element dataFormat / externallyDefinedFormat .....	128
Element dataFormat / externallyDefinedFormat / formatName .....	128
Element dataFormat / externallyDefinedFormat / formatVersion .....	128
Element additionalMetadata / metadata / gbif / resourceLogoUrl .....	129
Element additionalMetadata / metadata / gbif / collection .....	129
Element additionalMetadata / metadata / gbif / collection / parentCollectionIdentifier ...	130
Element additionalMetadata / metadata / gbif / collection / collectionIdentifier .....	130
Element additionalMetadata / metadata / gbif / collection / collectionName .....	131
Element additionalMetadata / metadata / gbif / formationPeriod .....	131
Element additionalMetadata / metadata / gbif / specimenPreservationMethod .....	131
Element additionalMetadata / metadata / gbif / livingTimePeriod .....	132
Element jgtiCuratorialUnit .....	132
Element jgtiCuratorialUnit / jgtiUnitType .....	133
Element jgtiCuratorialUnit / jgtiUnits .....	134
Element jgtiCuratorialUnit / jgtiUnitRange .....	134
Element jgtiCuratorialUnit / jgtiUnitRange / beginRange .....	135
Element jgtiCuratorialUnit / jgtiUnitRange / endRange .....	135
Simple Type(s) .....	136
Simple Type NonEmptyStringType .....	136
Simple Type IDType .....	137
Simple Type SystemType .....	137
Simple Type ScopeType .....	137
Simple Type yearDate .....	138
Simple Type MaintUpFreqType .....	138
Simple Type descriptorEnum .....	139
Complex Type(s) .....	139
Complex Type i18nString .....	139
Complex Type agentType .....	140
Complex Type agentWithRoleType .....	142
Complex Type TextType .....	144
Complex Type SectionType .....	145
Complex Type ParagraphType .....	146
Complex Type ListType .....	149
Complex Type SubSuperScriptType .....	149
Complex Type calendarDate .....	151
Complex Type description .....	151
Complex Type projectType .....	151
Complex Type awardType .....	153
Complex Type relatedProjectType .....	155
Complex Type CitationListType .....	155
Complex Type citationType .....	156
Attribute(s) .....	156

Attribute userId / @directory .....	156
Attribute agentType / @id .....	157
Attribute agentType / @system .....	157
Attribute agentType / @scope .....	157
Attribute ParagraphType / ulink / @url .....	157
Attribute ulink / @url .....	157
Attribute url / @function .....	158
Attribute distribution / @scope .....	158
Attribute descriptor / @citableClassificationSystem .....	158
Attribute descriptor / @name .....	158
Attribute relatedProjectType / @id .....	158
Attribute projectType / @id .....	159
Attribute citationType / @identifier .....	159
Attribute jgtiCuratorialUnit / jgtiUnits / @uncertaintyMeasure .....	159
Attribute eml:eml / @packageId .....	159
Attribute eml:eml / @scope .....	159
Attribute eml:eml / @system .....	160
Namespace: "https://eml.ecoinformatics.org/eml-2.2.0" .....	160
Schema(s) .....	160
Imported schema eml.xsd .....	160
Element(s) .....	160
Element eml:eml .....	160
Namespace: "http://www.w3.org/XML/1998/namespace" .....	161
Schema(s) .....	161
Imported schema xml.xsd .....	161
Imported schema xml.xsd .....	162
Attribute(s) .....	163
Attribute @xml:lang .....	163
Attribute @xml:space .....	164
Attribute @xml:base .....	165
Attribute @xml:id .....	165
Attribute Group(s) .....	165
Attribute Group xml:specialAttrs .....	165
Namespace: "http://purl.org/dc/terms/" .....	166
Schema(s) .....	166
Imported schema dc.xsd .....	166
Element(s) .....	166
Element replaces .....	166
Element URI .....	166
Element issued .....	166
Element type .....	167
Element title .....	167
Element source .....	167
Element subject .....	167
Element relation .....	167
Element description .....	167
Element created .....	168
Element modified .....	168

## Namespace: ""

### Schema(s)

#### Main schema eml-gbif-profile.xsd

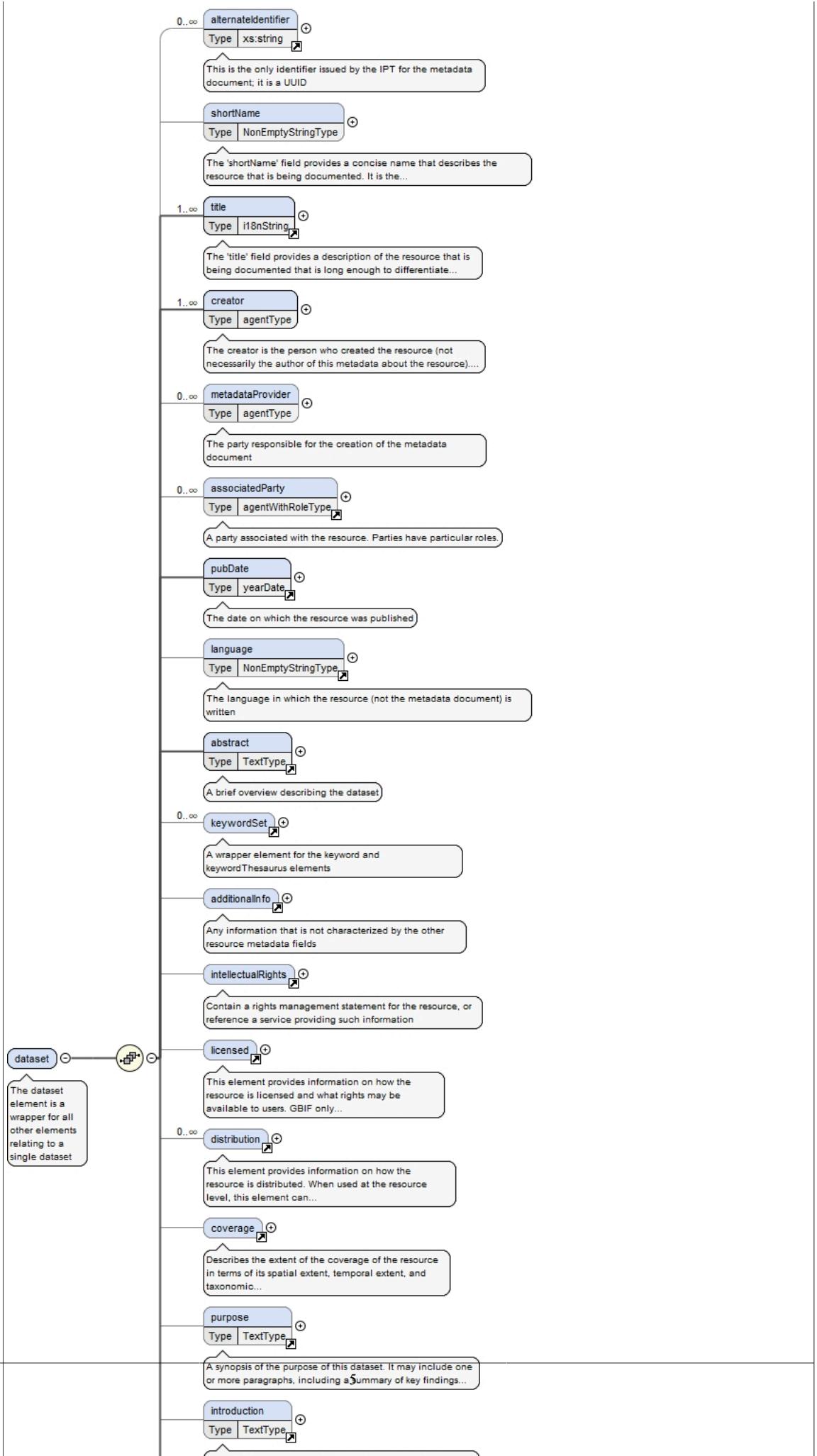
Namespace	No namespace
Properties	attribute form default: unqualified element form default: qualified version: 1.3

### Element(s)

#### Element dataset

Namespace	No namespace
Annotations	The dataset element is a wrapper for all other elements relating to a single dataset

## Diagram



Properties	content:	complex
Used by	Element	eml:eml
Model	alternateIdentifier*, shortName{0,1}, title+, creator+, metadataProvider*, associatedParty*, pubDate, language{0,1}, abstract, keywordSet*, additionalInfo{0,1}, intellectualRights{0,1}, licensed{0,1}, distribution*, coverage{0,1}, purpose{0,1}, introduction{0,1}, gettingStarted{0,1}, acknowledgements{0,1}, maintenance{0,1}, contact+, publisher{0,1}, methods{0,1}, project{0,1}, literatureCited*	
Children	abstract, acknowledgements, additionalInfo, alternateIdentifier, associatedParty, contact, coverage, creator, distribution, gettingStarted, intellectualRights, introduction, keywordSet, language, licensed, literatureCited, maintenance, metadataProvider, methods, project, pubDate, publisher, purpose, shortName, title	
Instance	<pre>&lt;dataset&gt;   &lt;alternateIdentifier&gt;{0,unbounded}&lt;/alternateIdentifier&gt;   &lt;shortName&gt;{0,1}&lt;/shortName&gt;   &lt;title xml:lang=""&gt;{1,unbounded}&lt;/title&gt;   &lt;creator id="" scope="" system=""&gt;{1,unbounded}&lt;/creator&gt;   &lt;metadataProvider id="" scope="" system=""&gt;{0,unbounded}&lt;/metadataProvider&gt;   &lt;associatedParty id="" scope="" system=""&gt;{0,unbounded}&lt;/associatedParty&gt;   &lt;pubDate&gt;{1,1}&lt;/pubDate&gt;   &lt;language&gt;{0,1}&lt;/language&gt;   &lt;abstract xml:lang=""&gt;{1,1}&lt;/abstract&gt;   &lt;keywordSet&gt;{0,unbounded}&lt;/keywordSet&gt;   &lt;additionalInfo&gt;{0,1}&lt;/additionalInfo&gt;   &lt;intellectualRights&gt;{0,1}&lt;/intellectualRights&gt;   &lt;licensed&gt;{0,1}&lt;/licensed&gt;   &lt;distribution scope=""&gt;{0,unbounded}&lt;/distribution&gt;   &lt;coverage&gt;{0,1}&lt;/coverage&gt;   &lt;purpose xml:lang=""&gt;{0,1}&lt;/purpose&gt;   &lt;introduction xml:lang=""&gt;{0,1}&lt;/introduction&gt;   &lt;gettingStarted xml:lang=""&gt;{0,1}&lt;/gettingStarted&gt;   &lt;acknowledgements xml:lang=""&gt;{0,1}&lt;/acknowledgements&gt;   &lt;maintenance&gt;{0,1}&lt;/maintenance&gt;   &lt;contact id="" scope="" system=""&gt;{1,unbounded}&lt;/contact&gt;   &lt;publisher id="" scope="" system=""&gt;{0,1}&lt;/publisher&gt;   &lt;methods&gt;{0,1}&lt;/methods&gt;   &lt;project id=""&gt;{0,1}&lt;/project&gt;   &lt;literatureCited&gt;{0,unbounded}&lt;/literatureCited&gt; &lt;/dataset&gt;</pre>	
Source	<pre>&lt;xss:element name="dataset"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;The dataset element is a wrapper for all other elements relating to a single dataset&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element ref="alternateIdentifier" minOccurs="0" maxOccurs="unbounded"/&gt;       &lt;xss:element name="shortName" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;The 'shortName' field provides a concise name that describes the resource that is being documented. It is the appropriate place to store a filename associated with other storage systems.&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;       &lt;xss:element ref="title" maxOccurs="unbounded"/&gt;       &lt;!-- Original creator of the dataset/resource, not necessarily the agent authoring the eml file nor the primary contact --&gt;       &lt;xss:element name="creator" type="agentType" maxOccurs="unbounded"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;The creator is the person who created the resource (not necessarily the author of this metadata about the resource). This is the person or institution to contact with questions about the use, interpretation of a dataset.&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;       &lt;!-- "Metadata author" the agent authoring the eml file --&gt;       &lt;xss:element name="metadataProvider" type="agentType" minOccurs="0" maxOccurs="unbounded"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;The party responsible for the creation of the metadata document&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;       &lt;xss:element ref="associatedParty" minOccurs="0" maxOccurs="unbounded"/&gt;       &lt;xss:element ref="pubDate"/&gt;       &lt;xss:element ref="language" minOccurs="0"/&gt;       &lt;xss:element ref="abstract"/&gt;       &lt;xss:element ref="keywordSet" minOccurs="0" maxOccurs="unbounded"/&gt;       &lt;xss:element ref="additionalInfo" minOccurs="0"/&gt;       &lt;xss:element ref="intellectualRights" minOccurs="0"/&gt;       &lt;xss:element ref="licensed" minOccurs="0"/&gt;       &lt;xss:element ref="distribution" minOccurs="0" maxOccurs="unbounded"/&gt;</pre>	

```

<xs:element ref="coverage" minOccurs="0" />
<xs:element ref="purpose" minOccurs="0" />
<xs:element ref="introduction" minOccurs="0" />
<xs:element ref="gettingStarted" minOccurs="0" />
<xs:element ref="acknowledgements" minOccurs="0" />
<xs:element minOccurs="0" name="maintenance">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="description" />
            <xs:element name="maintenanceUpdateFrequency" type="MaintUpFreqType" />
            <xs:element name="changeHistory" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>A description of changes made to the data since its release.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="changeScope" type="NonEmptyStringType" />
                <xs:annotation>
                    <xs:documentation>The expression should unambiguously identify the entity(s) and attribute(s) that were changed.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="oldValue" type="NonEmptyStringType" />
                <xs:annotation>
                    <xs:documentation>The previous value or an expression that describes the previous value of the data.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="changeDate" type="xs:date" />
                <xs:annotation>
                    <xs:documentation>The date the changes were applied.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="comment" type="NonEmptyStringType" minOccurs="0" />
                <xs:annotation>
                    <xs:documentation>Explanation or justification for the change made to the data.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<!-- Current primary contact for the dataset. The creator of the resource might be dead, left the organisation or doesn't want to be bothered. -->
<xs:element name="contact" type="agentType" maxOccurs="unbounded" />
<xs:element name="publisher" type="agentType" minOccurs="0" />
    <xs:annotation>
        <xs:documentation>The publisher of this data set. This is typically an institution that is making the data available in a published (ie, citable) format.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element ref="methods" minOccurs="0" />
<xs:element ref="project" minOccurs="0" />
<xs:element name="literatureCited" type="CitationListType" minOccurs="0" maxOccurs="unbounded" />
    <xs:annotation>
        <xs:documentation>A citation to articles or products which were referenced in the dataset or its associated metadata. The list represents the bibliography of works related to the dataset, whether for reference, comparison, or others purposes.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

## Element alternateIdentifier

Namespace	No namespace
Annotations	This is the only identifier issued by the IPT for the metadata document; it is a UUID
Diagram	<p>This is the only identifier issued by the IPT for the metadata document; it is a UUID</p> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>

Type	xs:string
Properties	content: simple
Used by	Element dataset
Source	<pre>&lt;xs:element name="alternateIdentifier" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is the only identifier issued by the IPT for the metadata document; it is a UUID&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element dataset / shortName

Namespace	No namespace				
Annotations	The 'shortName' field provides a concise name that describes the resource that is being documented. It is the appropriate place to store a filename associated with other storage systems.				
Diagram	<pre> classDiagram     class shortName {         &lt;&lt;NonEmptyStringType&gt;&gt;     }     class NonEmptyStringType {         &lt;&lt;Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...&gt;&gt;     }     shortName --o NonEmptyStringType   </pre> <p>The 'shortName' field provides a concise name that describes the resource that is being documented. It is the...</p> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p>				
Type	NonEmptyStringType				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[\\s]*[\\S][\\s\\S]*</td> </tr> </table>	minLength	1	pattern	[\\s]*[\\S][\\s\\S]*
minLength	1				
pattern	[\\s]*[\\S][\\s\\S]*				
Source	<pre>&lt;xs:element name="shortName" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The 'shortName' field provides a concise name that describes the resource that is being documented. It is the appropriate place to store a filename associated with other storage systems.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>				

## Element title

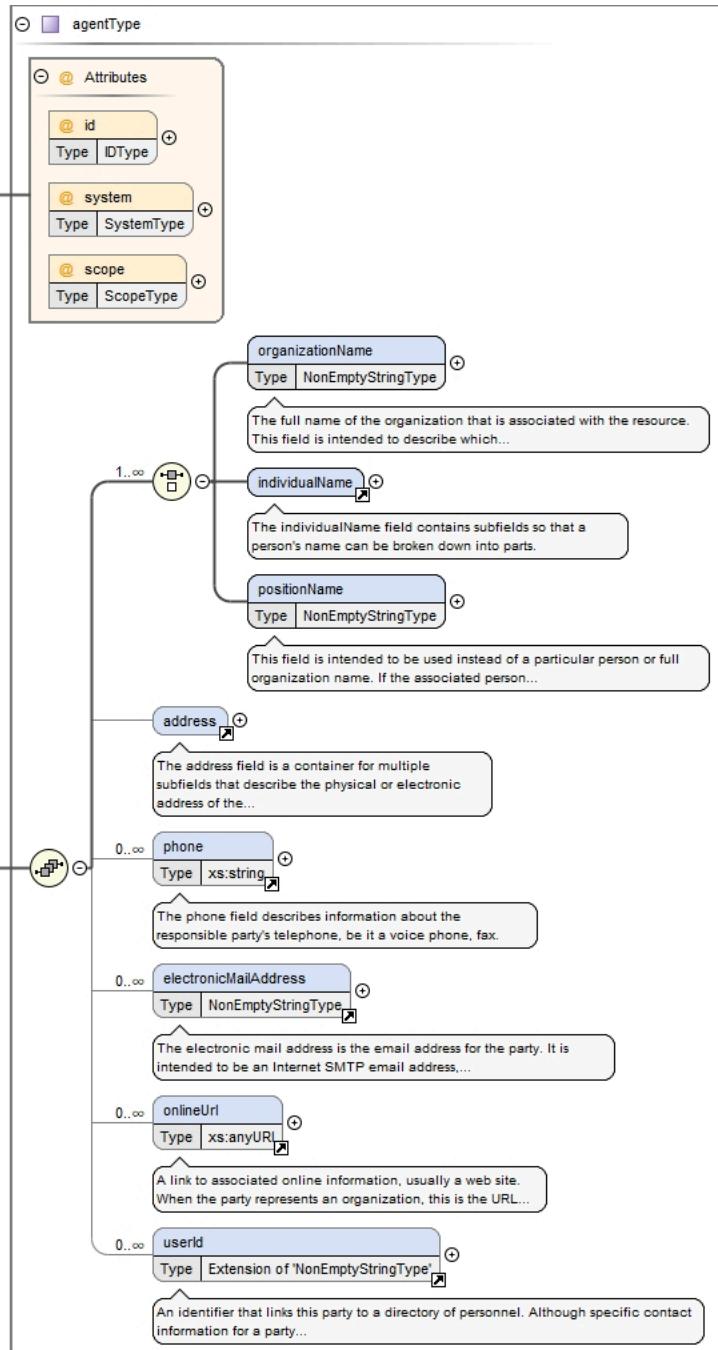
Namespace	No namespace
Annotations	The 'title' field provides a description of the resource that is being documented that is long enough to differentiate it from other similar resources. Multiple titles may be provided, particularly when trying to express the title in more than one language (use the "xml:lang" attribute to indicate the language if not English/en).
Diagram	<pre> classDiagram     class title {         &lt;&lt;i18nString&gt;&gt;         &lt;&lt;NonEmptyStringType&gt;&gt;         &lt;&lt;@ xml:lang&gt;&gt;     }     class i18nString {         &lt;&lt;NonEmptyStringType&gt;&gt;         &lt;&lt;@ xml:lang&gt;&gt;     }     class NonEmptyStringType {         &lt;&lt;Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...&gt;&gt;     }     title --o i18nString     i18nString --o NonEmptyStringType   </pre> <p>The 'title' field provides a description of the resource that is being documented that is long enough to differentiate...</p> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p> <p>@ xml:lang</p> <p>&lt;div&gt; &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt; &lt;p&gt; denotes an attribute whose value is a language code for the natural...</p>
Type	i18nString
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:string</li> <li>• NonEmptyStringType</li> <li>• i18nString</li> </ul>
Properties	content: complex

Used by	Element Complex Types	dataset projectType, relatedProjectType	
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>xml:lang</b>	union of(xs:language, restriction of xs:string)	optional
	<pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc- editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag- registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </pre>		
Source	<pre> &lt;xs:element name="title" type="i18nString"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The 'title' field provides a description of the resource that is being documented that is long enough to differentiate it from other similar resources. Multiple titles may be provided, particularly when trying to express the title in more than one language (use the "xml:lang" attribute to indicate the language if not English/en).&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>		

## Element dataset / creator

Namespace	No namespace
Annotations	The creator is the person who created the resource (not necessarily the author of this metadata about the resource). This is the person or institution to contact with questions about the use, interpretation of a dataset.

## Diagram



Type	agentType
Properties	content: complex maxOccurs: unbounded
Model	( <code>organizationName</code>   <code>individualName</code>   <code>positionName</code> ) , <code>address</code> {0,1} , <code>phone</code> * , <code>electronicMailAddress</code> * , <code>onlineUrl</code> * , <code>userId</code> *
Children	address, <code>electronicMailAddress</code> , <code>individualName</code> , <code>onlineUrl</code> , <code>organizationName</code> , <code>phone</code> , <code>positionName</code> , <code>userId</code>
Instance	<pre> &lt;creator id="" scope="" system=""&gt;   &lt;organizationName&gt;{1,1}&lt;/organizationName&gt;   &lt;individualName&gt;{1,1}&lt;/individualName&gt;   &lt;positionName&gt;{1,1}&lt;/positionName&gt;   &lt;address&gt;{0,1}&lt;/address&gt;   &lt;phone&gt;{0,unbounded}&lt;/phone&gt;   &lt;electronicMailAddress&gt;{0,unbounded}&lt;/electronicMailAddress&gt;   &lt;onlineUrl&gt;{0,unbounded}&lt;/onlineUrl&gt;   &lt;userId directory=""&gt;{0,unbounded}&lt;/userId&gt; &lt;/creator&gt; </pre>

Attributes	QName	Type	Use
	<b>id</b>	IDType	optional
	<b>scope</b>	ScopeType	optional
	<b>system</b>	SystemType	optional
Source	<pre>&lt;xs:element name="creator" type="agentType" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The creator is the person who created the resource (not necessarily the author of this metadata about the resource). This is the person or institution to contact with questions about the use, interpretation of a dataset.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

## Element agentType / organizationName

Namespace	No namespace
Annotations	The full name of the organization that is associated with the resource. This field is intended to describe which institution or overall organization is associated with the resource being described.
Diagram	<pre> classDiagram     class organizationName {         Type NonEmptyStringType     }     NonEmptyStringType     organizationName "0..1" --&gt; NonEmptyStringType     Note over organizationName: The full name of the organization that is associated with the resource. This field is intended to describe which...     Note over NonEmptyStringType: Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...   </pre>
Type	NonEmptyStringType
Properties	content: simple
Facets	minLength 1 pattern <code>[\s]*[\S][\s\S]*</code>
Source	<pre>&lt;xs:element name="organizationName" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The full name of the organization that is associated with the resource. This field is intended to describe which institution or overall organization is associated with the resource being described.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element individualName

Namespace	No namespace
Annotations	The individualName field contains subfields so that a person's name can be broken down into parts.
Diagram	<pre> classDiagram     class individualName {         "0..1" --&gt; salutation         "0..1" --&gt; givenName         "0..1" --&gt; surName     }     class salutation {         Type NonEmptyStringType     }     class givenName {         Type NonEmptyStringType     }     class surName {         Type NonEmptyStringType     }     Note over individualName: The individualName field contains subfields so that a person's name can be broken down into parts.     Note over salutation: The salutation field is used in addressing an individual with a particular title, such as Dr., Ms., Mrs., Mr., etc.     Note over givenName: The given name field can be used for first name of the individual associated with the resource, or for any other names...     Note over surName: The surname field is used for the last name of the individual associated with the resource. This is typically the...   </pre>
Properties	content: complex
Used by	Complex Types agentType, agentWithRoleType
Model	salutation*, givenName{0,1}, surName
Children	givenName, salutation, surName
Instance	<pre>&lt;individualName&gt;   &lt;salutation&gt;{0,unbounded}&lt;/salutation&gt;   &lt;givenName&gt;{0,1}&lt;/givenName&gt;   &lt;surName&gt;{1,1}&lt;/surName&gt; &lt;/individualName&gt;</pre>

Source	<pre> &lt;xs:element name="individualName"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The individualName field contains subfields so that a person's name can be broken down into parts.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="salutation" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The salutation field is used in addressing an individual with a particular title, such as Dr., Ms., Mrs., Mr., etc.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="givenName" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The given name field can be used for first name of the individual associated with the resource, or for any other names that are not intended to be alphabetized, (as appropriate).&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="surName" type="NonEmptyStringType"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The surname field is used for the last name of the individual associated with the resource. This is typically the family name of an individual, for example, the name by which s/he is referred to in citations.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>
--------	---

## Element individualName / salutation

Namespace	No namespace						
Annotations	The salutation field is used in addressing an individual with a particular title, such as Dr., Ms., Mrs., Mr., etc.						
Diagram	<p>The diagram shows a UML class named 'salutation'. It has a self-referencing association loop labeled 'NonEmptyStringType'. A callout box points to this association with the text: 'The salutation field is used in addressing an individual with a particular title, such as Dr., Ms., Mrs., Mr., etc.' Another callout box points to the 'NonEmptyStringType' label with the text: 'Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...'. The 'NonEmptyStringType' label is enclosed in a rounded rectangle.</p>						
Type	NonEmptyStringType						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	unbounded
content:	simple						
minOccurs:	0						
maxOccurs:	unbounded						
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ]*[ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ]*[ \s \S ]*		
minLength	1						
pattern	[ \s ]*[ \S ]*[ \s \S ]*						
Source	<pre> &lt;xs:element name="salutation" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The salutation field is used in addressing an individual with a particular title, such as Dr., Ms., Mrs., Mr., etc.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>						

## Element individualName / givenName

Namespace	No namespace		
Annotations	The given name field can be used for first name of the individual associated with the resource, or for any other names that are not intended to be alphabetized, (as appropriate).		
Diagram	<p>The diagram shows a UML class named 'givenName'. It has a self-referencing association loop labeled 'NonEmptyStringType'. A callout box points to this association with the text: 'The given name field can be used for first name of the individual associated with the resource, or for any other names...'. Another callout box points to the 'NonEmptyStringType' label with the text: 'Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...'. The 'NonEmptyStringType' label is enclosed in a rounded rectangle.</p>		
Type	NonEmptyStringType		
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple
content:	simple		

	minOccurs:	0
Facets	minLength	1
	pattern	[ \s ]*[ \S ]*[ \s \S ]*

Source

```
<xs:element name="givenName" type="NonEmptyStringType" minOccurs="0">
  <xs:annotation>
    <xs:documentation>The given name field can be used for first name of the individual associated with the resource, or for any other names that are not intended to be alphabetized, (as appropriate).</xs:documentation>
  </xs:annotation>
</xs:element>
```

### Element individualName / surName

Namespace	No namespace
Annotations	The surname field is used for the last name of the individual associated with the resource. This is typically the family name of an individual, for example, the name by which s/he is referred to in citations.
Diagram	
Type	NonEmptyStringType
Properties	content: simple
Facets	minLength 1
	pattern [ \s ]*[ \S ]*[ \s \S ]*
Source	<pre>&lt;xs:element name="surName" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The surname field is used for the last name of the individual associated with the resource. This is typically the family name of an individual, for example, the name by which s/he is referred to in citations.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element agentType / positionName

Namespace	No namespace
Annotations	This field is intended to be used instead of a particular person or full organization name. If the associated person who holds the role changes frequently, then Position Name would be used for consistency. E.g., GBIF Data Manager.
Diagram	
Type	NonEmptyStringType
Properties	content: simple
Facets	minLength 1
	pattern [ \s ]*[ \S ]*[ \s \S ]*
Source	<pre>&lt;xs:element name="positionName" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This field is intended to be used instead of a particular person or full organization name. If the associated person who holds the role changes frequently, then Position Name would be used for consistency. E.g., GBIF Data Manager.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element address

Namespace	No namespace
Annotations	The address field is a container for multiple subfields that describe the physical or electronic address of the responsible party for a resource.

Diagram	<pre> classDiagram     class address {         deliveryPoint "0..oo"         Type NonEmptyStringType         city         Type NonEmptyStringType         administrativeArea         Type NonEmptyStringType         postalCode         Type NonEmptyStringType         country         Type NonEmptyStringType     }     address &lt; -- deliveryPoint     address &lt; -- city     address &lt; -- administrativeArea     address &lt; -- postalCode     address &lt; -- country   </pre> <p>The address field is a container for multiple subfields that describe the physical or electronic address of the...</p> <p>The delivery point field is used for the physical address for postal communication, e.g., GBIF Secretariat,...</p> <p>The city field is used for the city name of the contact associated with a particular resource.</p> <p>The administrative area field is the equivalent of a 'state' in the U.S., or Province in Canada. This field is intended...</p> <p>The postal code is equivalent to a U.S. zip code, or the number used for routing to an international address.</p> <p>The country field is used for the name of the contact's country.</p>
Properties	content: complex
Used by	Complex Types agentType, agentWithRoleType
Model	deliveryPoint*, city{0,1}, administrativeArea{0,1}, postalCode{0,1}, country{0,1}
Children	administrativeArea, city, country, deliveryPoint, postalCode
Instance	<pre> &lt;address&gt;   &lt;deliveryPoint&gt;{0,unbounded}&lt;/deliveryPoint&gt;   &lt;city&gt;{0,1}&lt;/city&gt;   &lt;administrativeArea&gt;{0,1}&lt;/administrativeArea&gt;   &lt;postalCode&gt;{0,1}&lt;/postalCode&gt;   &lt;country&gt;{0,1}&lt;/country&gt; &lt;/address&gt;   </pre>
Source	<pre> &lt;xs:element name="address"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The address field is a container for multiple subfields that describe the physical or electronic address of the responsible party for a resource.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="deliveryPoint" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The delivery point field is used for the physical address for postal communication, e.g., GBIF Secretariat, Universitetsparken 15&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="city" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The city field is used for the city name of the contact associated with a particular resource.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="administrativeArea" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The administrative area field is the equivalent of a 'state' in the U.S., or Province in Canada. This field is intended to accommodate the many types of international administrative areas.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="postalCode" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The postal code is equivalent to a U.S. zip code, or the number used for routing to an international address.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="country" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The country field is used for the name of the contact's country.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>

```

</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

## Element address / deliveryPoint

Namespace	No namespace						
Annotations	The delivery point field is used for the physical address for postal communication, e.g., GBIF Secretariat, Universitetsparken 15						
Diagram	<p>The diagram shows a class named 'deliveryPoint' with a multiplicity of 0..1. It is associated with a 'NonEmptyStringType' via a directed edge. A callout box points to the 'NonEmptyStringType' box, stating: 'The delivery point field is used for the physical address for postal communication, e.g., GBIF Secretariat,...'. Another callout box points to the 'NonEmptyStringType' box, stating: 'Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...'. The 'NonEmptyStringType' box has a self-loop arrow.</p>						
Type	NonEmptyStringType						
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	unbounded
content:	simple						
minOccurs:	0						
maxOccurs:	unbounded						
Facets	<table> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ] [ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ] [ \s \S ]*		
minLength	1						
pattern	[ \s ]*[ \S ] [ \s \S ]*						
Source	<pre> &lt;xs:element name="deliveryPoint" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The delivery point field is used for the physical address for postal communication, e.g., GBIF Secretariat, Universitetsparken 15&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>						

## Element address / city

Namespace	No namespace				
Annotations	The city field is used for the city name of the contact associated with a particular resource.				
Diagram	<p>The diagram shows a class named 'city' with a multiplicity of 0..1. It is associated with a 'NonEmptyStringType' via a directed edge. A callout box points to the 'NonEmptyStringType' box, stating: 'The city field is used for the city name of the contact associated with a particular resource...'. Another callout box points to the 'NonEmptyStringType' box, stating: 'Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...'. The 'NonEmptyStringType' box has a self-loop arrow.</p>				
Type	NonEmptyStringType				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ] [ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ] [ \s \S ]*
minLength	1				
pattern	[ \s ]*[ \S ] [ \s \S ]*				
Source	<pre> &lt;xs:element name="city" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The city field is used for the city name of the contact associated with a particular resource.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>				

## Element address / administrativeArea

Namespace	No namespace
Annotations	The administrative area field is the equivalent of a 'state' in the U.S., or Province in Canada. This field is intended to accommodate the many types of international administrative areas.
Diagram	<p>The diagram shows a class named 'administrativeArea' with a multiplicity of 0..1. It is associated with a 'NonEmptyStringType' via a directed edge. A callout box points to the 'NonEmptyStringType' box, stating: 'The administrative area field is the equivalent of a 'state' in the U.S., or Province in Canada. This field is intended...'. Another callout box points to the 'NonEmptyStringType' box, stating: 'Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...'. The 'NonEmptyStringType' box has a self-loop arrow.</p>

Type	NonEmptyStringType
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Facets	<p>minLength 1</p> <p>pattern <code>[ \s]*[\s][\s\S]*</code></p>
Source	<pre>&lt;xs:element name="administrativeArea" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The administrative area field is the equivalent of a 'state' in the U.S., or Province in Canada. This field is intended to accommodate the many types of international administrative areas.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element address / postalCode

Namespace	No namespace
Annotations	The postal code is equivalent to a U.S. zip code, or the number used for routing to an international address.
Diagram	<pre> classDiagram     class postalCode {         Type NonEmptyStringType     }     class NonEmptyStringType     postalCode "0..1" -- "1..1" NonEmptyStringType     Note over postalCode: The postal code is equivalent to a U.S. zip code, or the number used for routing to an international address.     Note over NonEmptyStringType: Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...   </pre>
Type	NonEmptyStringType
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Facets	<p>minLength 1</p> <p>pattern <code>[ \s]*[\s][\s\S]*</code></p>
Source	<pre>&lt;xs:element name="postalCode" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The postal code is equivalent to a U.S. zip code, or the number used for routing to an international address.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element address / country

Namespace	No namespace
Annotations	The country field is used for the name of the contact's country.
Diagram	<pre> classDiagram     class country {         Type NonEmptyStringType     }     class NonEmptyStringType     country "0..1" -- "1..1" NonEmptyStringType     Note over country: The country field is used for the name of the contact's country.     Note over NonEmptyStringType: Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...   </pre>
Type	NonEmptyStringType
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Facets	<p>minLength 1</p> <p>pattern <code>[ \s]*[\s][\s\S]*</code></p>
Source	<pre>&lt;xs:element name="country" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The country field is used for the name of the contact's country.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element phone

Namespace	No namespace
-----------	--------------

Annotations	The phone field describes information about the responsible party's telephone, be it a voice phone, fax.
Diagram	<pre> graph LR     phone[phone] --&gt; xsString[xs:string]     subgraph Documentation [ ]         direction TB         D1["The phone field describes information about the responsible party's telephone, be it a voice phone, fax."]         D2["Built-in primitive type. The string datatype represents character strings in XML."]         D1 --- D2     end </pre>
Type	xs:string
Properties	content: simple
Used by	Complex Types agentType, agentWithRoleType
Source	<pre> &lt;xss:element name="phone" type="xs:string"&gt;     &lt;xss:annotation&gt;         &lt;xss:documentation&gt;The phone field describes information about the responsible party's telephone, be it a voice phone, fax.&lt;/xss:documentation&gt;     &lt;/xss:annotation&gt; &lt;/xss:element&gt; </pre>

## Element electronicEmailAddress

Namespace	No namespace				
Annotations	The electronic mail address is the email address for the party. It is intended to be an Internet SMTP email address, which should consist of a username followed by the @ symbol, followed by the email server domain name address.				
Diagram	<pre> graph LR     eamail[electronicEmailAddress] --&gt; NonEmptyStringType[NonEmptyStringType]     subgraph Documentation [ ]         direction TB         D1["The electronic mail address is the email address for the party. It is intended to be an Internet SMTP email address...."]         D2["Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that..."]         D1 --- D2     end </pre>				
Type	NonEmptyStringType				
Properties	content: simple				
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ]*[ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ]*[ \s \S ]*
minLength	1				
pattern	[ \s ]*[ \S ]*[ \s \S ]*				
Used by	Complex Types agentType, agentWithRoleType				
Source	<pre> &lt;xss:element name="electronicEmailAddress" type="NonEmptyStringType"&gt;     &lt;xss:annotation&gt;         &lt;xss:documentation&gt;The electronic mail address is the email address for the party. It is intended to be an Internet SMTP email address, which should consist of a username followed by the @ symbol, followed by the email server domain name address.&lt;/xss:documentation&gt;     &lt;/xss:annotation&gt; &lt;/xss:element&gt; </pre>				

## Element onlineUrl

Namespace	No namespace
Annotations	A link to associated online information, usually a web site. When the party represents an organization, this is the URL to a website or other online information about the organization. If the party is an individual, it might be their personal web site or other related online information about the party.
Diagram	<pre> graph LR     onlineUrl[onlineUrl] --&gt; xsAnyURI[xs:anyURI]     subgraph Documentation [ ]         direction TB         D1["A link to associated online information, usually a web site. When the party represents an organization, this is the URL...."]         D2["Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI)."]         D1 --- D2     end </pre>
Type	xs:anyURI
Properties	content: simple
Used by	Complex Types agentType, agentWithRoleType

Source	<pre>&lt;xs:element name="onlineUrl" type="xs:anyURI"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A link to associated online information, usually a web site. When the party represents an organization, this is the URL to a website or other online information about the organization. If the party is an individual, it might be their personal web site or other related online information about the party.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>
--------	--

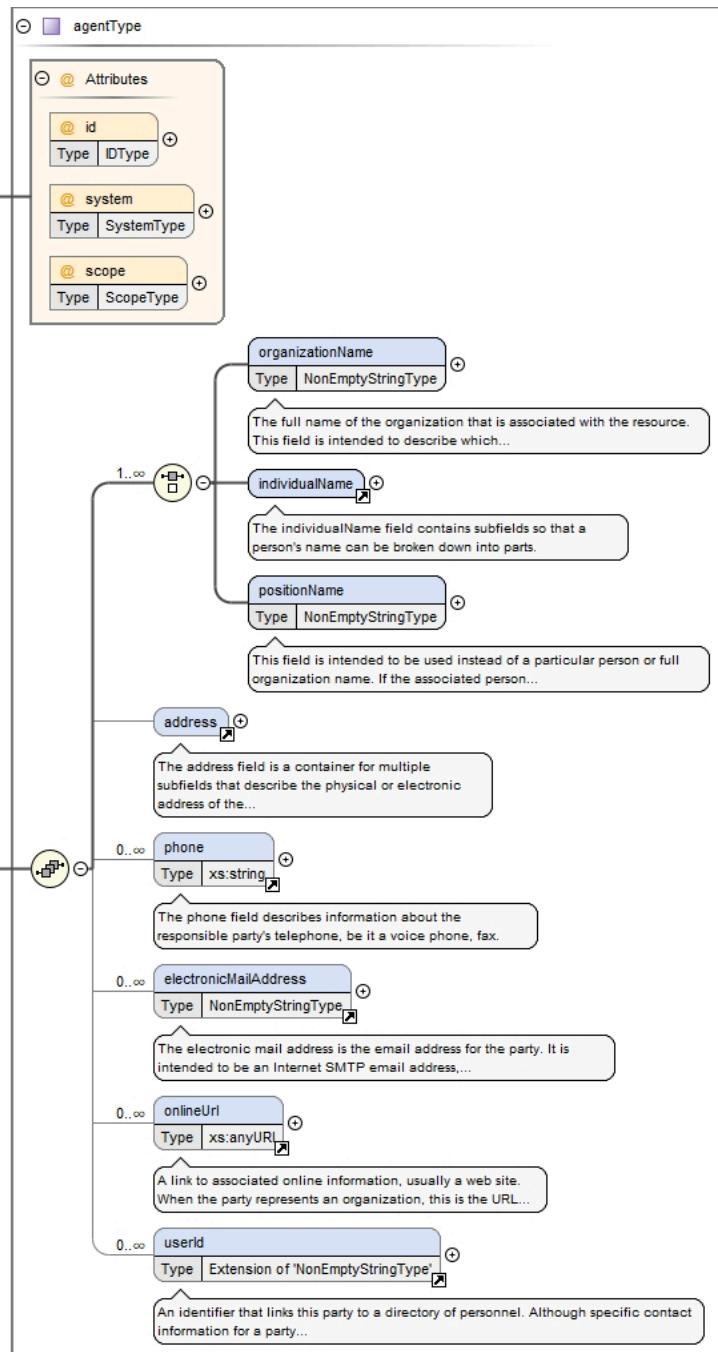
## Element userId

Namespace	No namespace						
Annotations	An identifier that links this party to a directory of personnel. Although specific contact information for a party might change, the underlying correspondence to a real individual does not. This identifier provides a pointer within a personnel directory that may contain further, and possibly more current, information about the party.						
Diagram							
Type	extension of NonEmptyStringType						
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:string</li> <li>• NonEmptyStringType</li> </ul>						
Properties	content: complex						
Used by	Complex Types agentType, agentWithRoleType						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>directory</td> <td>NonEmptyStringType</td> <td>required</td> </tr> </tbody> </table> <p>This attribute names the directory system to which this userId applies. This will generally be a URL that shows how to look up information, for example an LDAP url. However, it could also be a non-parsable description of the directory system if that is all that is available.</p>	QName	Type	Use	directory	NonEmptyStringType	required
QName	Type	Use					
directory	NonEmptyStringType	required					
Source	<pre>&lt;xs:element name="userId"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;An identifier that links this party to a directory of personnel. Although specific contact information for a party might change, the underlying correspondence to a real individual does not. This identifier provides a pointer within a personnel directory that may contain further, and possibly more current, information about the party.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="NonEmptyStringType"&gt;         &lt;xs:attribute name="directory" use="required" type="NonEmptyStringType"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;This attribute names the directory system to which this userId applies. This will generally be a URL that shows how to look up information, for example an LDAP url. However, it could also be a non-parsable description of the directory system if that is all that is available.&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:attribute&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>						

## Element dataset / metadataProvider

Namespace	No namespace
Annotations	The party responsible for the creation of the metadata document

## Diagram



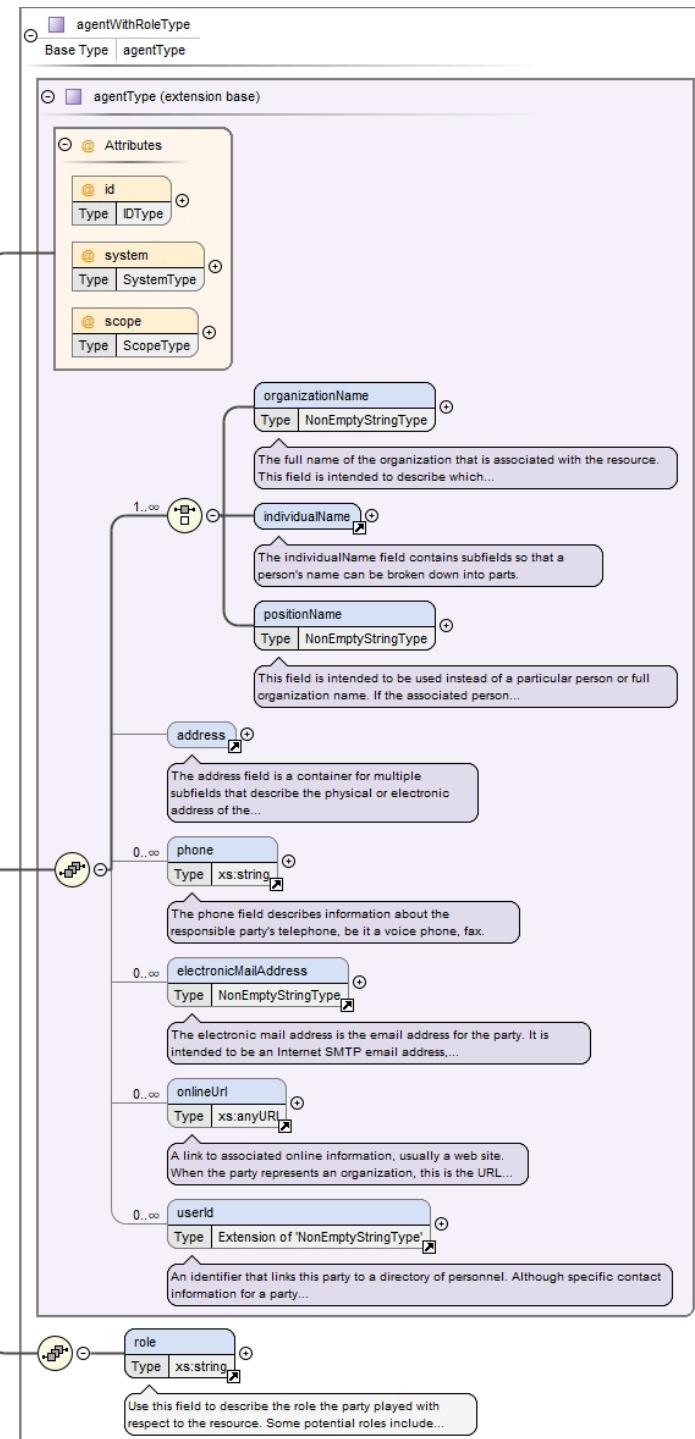
Type	agentType
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Model	(organizationName   individualName   positionName) , address{0,1} , phone* , electronicMailAddress* , onlineUrl* , userId*
Children	address, electronicMailAddress, individualName, onlineUrl, organizationName, phone, positionName, userId
Instance	<pre>&lt;metadataProvider id="" scope="" system=""&gt;   &lt;organizationName&gt;{1,1}&lt;/organizationName&gt;   &lt;individualName&gt;{1,1}&lt;/individualName&gt;   &lt;positionName&gt;{1,1}&lt;/positionName&gt;   &lt;address&gt;{0,1}&lt;/address&gt;   &lt;phone&gt;{0,unbounded}&lt;/phone&gt;   &lt;electronicMailAddress&gt;{0,unbounded}&lt;/electronicMailAddress&gt;   &lt;onlineUrl&gt;{0,unbounded}&lt;/onlineUrl&gt;   &lt;userId directory=""&gt;{0,unbounded}&lt;/userId&gt; &lt;/metadataProvider&gt;</pre>

Attributes	QName	Type	Use
	<b>id</b>	IDType	optional
	<b>scope</b>	ScopeType	optional
	<b>system</b>	SystemType	optional
Source	<pre>&lt;xs:element name="metadataProvider" type="agentType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The party responsible for the creation of the metadata document&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

### Element associatedParty

Namespace	No namespace
Annotations	A party associated with the resource. Parties have particular roles.

## Diagram



Type	<code>agentWithRoleType</code>
Type hierarchy	<ul style="list-style-type: none"> <li>• <code>agentType</code></li> <li>• <code>agentWithRoleType</code></li> </ul>
Properties	content: complex
Used by	Element dataset
Model	( <code>organizationName</code>   <code>individualName</code>   <code>positionName</code> ) , <code>address</code> {0,1} , <code>phone</code> * , <code>electronicMailAddress</code> * , <code>onlineUrl</code> * , <code>userId</code> * , <code>role</code>
Children	<code>address</code> , <code>electronicMailAddress</code> , <code>individualName</code> , <code>onlineUrl</code> , <code>organizationName</code> , <code>phone</code> , <code>positionName</code> , <code>role</code> , <code>userId</code>
Instance	<pre>&lt;associatedParty id="" scope="" system=""&gt;   &lt;organizationName&gt;{1,1}&lt;/organizationName&gt;   &lt;individualName&gt;{1,1}&lt;/individualName&gt;   &lt;positionName&gt;{1,1}&lt;/positionName&gt;</pre>

	<pre> &lt;address&gt;{0,1}&lt;/address&gt; &lt;phone&gt;{0,unbounded}&lt;/phone&gt; &lt;electronicEmailAddress&gt;{0,unbounded}&lt;/electronicEmailAddress&gt; &lt;onlineUrl&gt;{0,unbounded}&lt;/onlineUrl&gt; &lt;userId directory=""&gt;{0,unbounded}&lt;/userId&gt; &lt;role&gt;{1,1}&lt;/role&gt; &lt;/associatedParty&gt; </pre>												
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td><b>id</b></td><td>IDType</td><td>optional</td></tr> <tr> <td><b>scope</b></td><td>ScopeType</td><td>optional</td></tr> <tr> <td><b>system</b></td><td>SystemType</td><td>optional</td></tr> </tbody> </table>	QName	Type	Use	<b>id</b>	IDType	optional	<b>scope</b>	ScopeType	optional	<b>system</b>	SystemType	optional
QName	Type	Use											
<b>id</b>	IDType	optional											
<b>scope</b>	ScopeType	optional											
<b>system</b>	SystemType	optional											
Source	<pre> &lt;xs:element name="associatedParty" type="agentWithRoleType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A party associated with the resource. Parties have particular roles.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>												

## Element role

Namespace	No namespace
Annotations	Use this field to describe the role the party played with respect to the resource. Some potential roles include technician, reviewer, principal investigator, and many others.
Diagram	<p>The diagram shows a class named 'role' with a multiplicity of 1..* at its end. It has an association with a 'xs:string' type, also with a multiplicity of 1..*. A callout box points to the 'role' class with the text: 'Use this field to describe the role the party played with respect to the resource. Some potential roles include...'. Another callout box points to the 'xs:string' type with the text: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xs:string
Properties	content: simple
Used by	Complex Type agentWithRoleType
Source	<pre> &lt;xs:element name="role" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Use this field to describe the role the party played with respect to the resource. Some potential roles include technician, reviewer, principal investigator, and many others.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>

## Element pubDate

Namespace	No namespace
Annotations	The date on which the resource was published
Diagram	<p>The diagram shows a class named 'pubDate' with a multiplicity of 1..* at its end. It has an association with a 'yearDate' type, also with a multiplicity of 1..*. A callout box points to the 'pubDate' class with the text: 'The date on which the resource was published'. Another callout box points to the 'yearDate' type with the text: 'A type allowing a year or date value. This type is the union of the built-in types for year and date. Example: 1999, or...'</p>
Type	yearDate
Properties	content: simple
Used by	Element dataset
Source	<pre> &lt;xs:element name="pubDate" type="yearDate"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The date on which the resource was published&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>

## Element language

Namespace	No namespace
-----------	--------------

Annotations	The language in which the resource (not the metadata document) is written				
Diagram	<pre> classDiagram     class language {         &lt;&lt;language&gt;&gt;         &lt;&lt;Type   NonEmptyStringType&gt;&gt;     }     NonEmptyStringType &lt; -- language   </pre> <p>The language in which the resource (not the metadata document) is written</p> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p>				
Type	NonEmptyStringType				
Properties	content: simple				
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ]*[ \s\S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ]*[ \s\S ]*
minLength	1				
pattern	[ \s ]*[ \S ]*[ \s\S ]*				
Used by	Element dataset				
Source	<pre> &lt;xss:element name="language" type="NonEmptyStringType"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;The language in which the resource (not the metadata document) is written&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;   </pre>				

## Element abstract

Namespace	No namespace									
Annotations	A brief overview describing the dataset									
Diagram	<pre> classDiagram     class abstract {         &lt;&lt;abstract&gt;&gt;         &lt;&lt;Type   TextType&gt;&gt;     }     TextType &lt; -- abstract     abstract &lt; -- @xml:lang     abstract &lt; -- section     abstract &lt; -- para   </pre> <p>A brief overview describing the dataset</p> <p>TextType Mixed true</p> <p>@xml:lang</p> <p>&lt;div&gt; &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt; &lt;p&gt;denotes an attribute whose value is a language code for the natural...</p> <p>section</p> <p>para</p> <p>The "text" element allows for both formatted and unformatted text blocks to be included in EML. It can contain a number...</p>									
Type	TextType									
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>mixed:</td> <td>true</td> </tr> </table>	content:	complex	mixed:	true					
content:	complex									
mixed:	true									
Used by	<table border="1"> <tr> <td>Element</td> <td>dataset</td> </tr> <tr> <td>Complex Types</td> <td>projectType, relatedProjectType</td> </tr> </table>	Element	dataset	Complex Types	projectType, relatedProjectType					
Element	dataset									
Complex Types	projectType, relatedProjectType									
Model	section*   para*									
Children	para, section									
Instance	<pre> &lt;abstract xml:lang=""&gt;   &lt;section xml:lang=""&gt;{0,unbounded}&lt;/section&gt;   &lt;para xml:lang=""&gt;{0,unbounded}&lt;/para&gt; &lt;/abstract&gt;   </pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>xml:lang</td> <td>union of(xs:language, restriction of xs:string)</td> <td>optional</td> </tr> <tr> <td></td> <td>&lt;div&gt; &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</td> <td></td> </tr> </tbody> </table>	QName	Type	Use	xml:lang	union of(xs:language, restriction of xs:string)	optional		<div> <h3>lang (as an attribute name)</h3>	
QName	Type	Use								
xml:lang	union of(xs:language, restriction of xs:string)	optional								
	<div> <h3>lang (as an attribute name)</h3>									

QName	Type	Use
		<p>&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p style="padding-left: 2em;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="padding-left: 2em;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="padding-left: 2em;">&lt;p&gt;See BCP 47 at  <a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a> and the IANA language subtag registry at  <a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a> for further information.&lt;/p&gt;</p> <p style="padding-left: 2em;">&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p>
Source	<pre>&lt;xs:element name="abstract" type="TextType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A brief overview describing the dataset&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>	

## Element TextType / section

Namespace	No namespace						
Annotations	The "section" element allows for grouping related paragraphs of text together, with an optional title. This markup is a subset of DocBook.						
Diagram	<pre> classDiagram     class SectionType {         @xml:lang         title : i18nString         para : ParagraphType         &lt;&gt; SectionType     }     SectionType &lt; -- section : SectionType     section --&gt; para     section --&gt; section   </pre>						
Type	SectionType						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	title{0,1} , (para   section)						
Children	para, section, title						
Instance	<pre>&lt;section xml:lang=""&gt;   &lt;title xml:lang=""&gt;{0,1}&lt;/title&gt;   &lt;para xml:lang=""&gt;{1,1}&lt;/para&gt;   &lt;section xml:lang=""&gt;{1,1}&lt;/section&gt; &lt;/section&gt;</pre>						

Attributes	QName	Type	Use
	xml:lang	union of(xs:language, restriction of xs:string)	optional
	<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc- editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag- registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt;</pre>		
Source	<pre>&lt;xs:element name="section" type="SectionType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The "section" element allows for grouping related paragraphs of text together, with an optional title. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

## Element sectionType / title

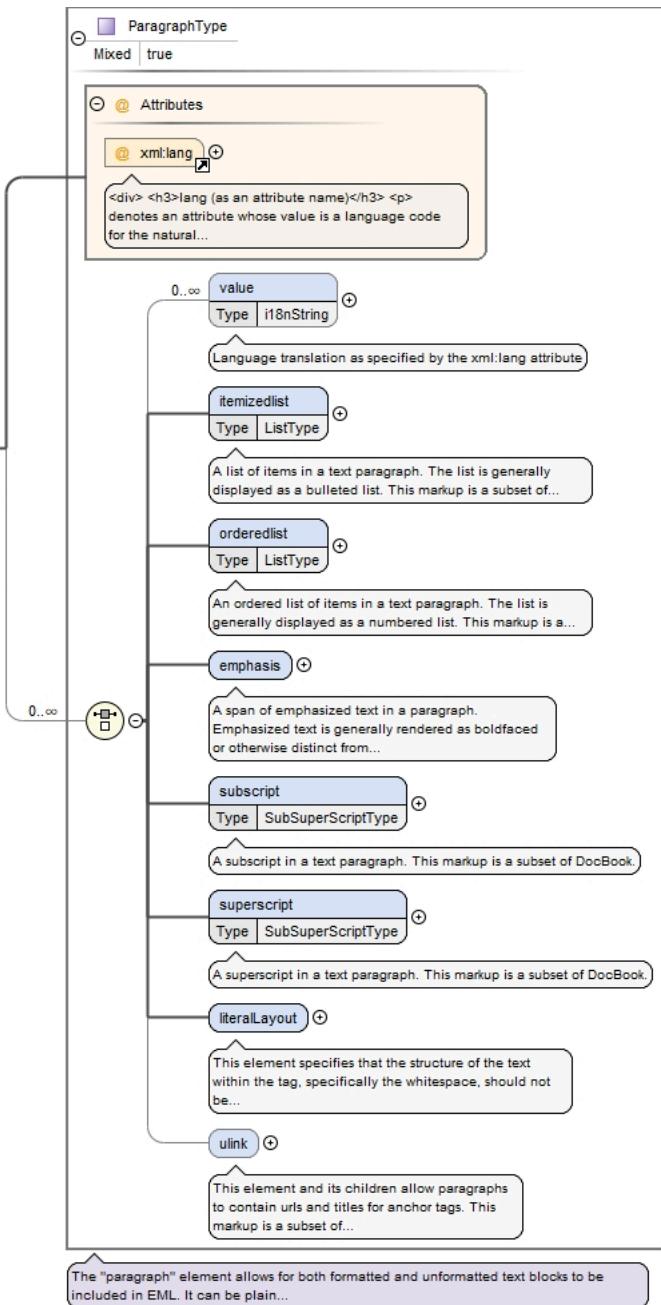
Namespace	No namespace									
Annotations	The optional title for a section. This markup is a subset of DocBook.									
Diagram	<p>The diagram illustrates the type hierarchy for the 'title' element. It starts with a 'title' element, which is a simple type derived from 'i18nString'. 'i18nString' is a base type for 'NonEmptyStringType', which is described as a Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that... An attribute '@ xml:lang' is shown with a note explaining its meaning.</p>									
Type	i18nString									
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:string</li> <li>• NonEmptyStringType</li> <li>• i18nString</li> </ul>									
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0					
content:	complex									
minOccurs:	0									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>xml:lang</td> <td>union of(xs:language, restriction of xs:string)</td> <td>optional</td> </tr> <tr> <td></td> <td colspan="2"> <pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;</pre> </td></tr> </tbody> </table>	QName	Type	Use	xml:lang	union of(xs:language, restriction of xs:string)	optional		<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;</pre>	
QName	Type	Use								
xml:lang	union of(xs:language, restriction of xs:string)	optional								
	<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;</pre>									

QName	Type	Use
		<p>&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p>&lt;p&gt;See BCP 47 at  <a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a> and the IANA language subtag registry at  <a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a> for further information.&lt;/p&gt;</p> <p>&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p>
Source		<pre>&lt;xs:element name="title" type="i18nString" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The optional title for a section. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element sectionType / para

Namespace	No namespace
Annotations	The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.

## Diagram



Type	ParagraphType
Properties	content: complex mixed: true
Model	<code>value*   itemizedlist   orderedlist   emphasis   subscript   superscript   literalLayout   ulink {0,1}</code>
Children	emphasis, itemizedlist, literalLayout, orderedlist, subscript, superscript, ulink, value
Instance	<pre>&lt;para xml:lang=""&gt;   &lt;value xml:lang=""&gt;{0,unbounded}&lt;/value&gt;   &lt;itemizedlist&gt;{1,1}&lt;/itemizedlist&gt;   &lt;orderedlist&gt;{1,1}&lt;/orderedlist&gt;   &lt;emphasis xml:lang=""&gt;{1,1}&lt;/emphasis&gt;   &lt;subscript xml:lang=""&gt;{1,1}&lt;/subscript&gt;   &lt;superscript xml:lang=""&gt;{1,1}&lt;/superscript&gt;   &lt;literalLayout&gt;{1,1}&lt;/literalLayout&gt;   &lt;ulink url=""&gt;{0,1}&lt;/ulink&gt; &lt;/para&gt;</pre>

Attributes	QName	Type	Use
	<b>xml:lang</b>	union of(xs:language, restriction of xs:string)	optional
	<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc- editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag- registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt;</pre>		
Source	<pre>&lt;xs:element name="para" type="ParagraphType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

## Element ParagraphType / value

Namespace	No namespace									
Annotations	Language translation as specified by the xml:lang attribute									
Diagram	<pre> classDiagram     class value {         &lt;&lt;i18nString&gt;&gt;         &lt;&lt;Type&gt;&gt;     }     class i18nString {         &lt;&lt;Base Type   NonEmptyStringType&gt;&gt;     }     class NonEmptyStringType {         &lt;&lt;NonEmptyStringType&gt;&gt;         &lt;&lt;@ Attributes&gt;&gt;         &lt;&lt;@ xml:lang&gt;&gt;     }      value &lt; -- i18nString     i18nString &lt; -- NonEmptyStringType     NonEmptyStringType &lt; -- NonEmptyStringType   </pre> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p> <p>@ xml:lang</p> <p>&lt;div&gt; &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt; &lt;p&gt;denotes an attribute whose value is a language code for the natural...</p>									
Type	i18nString									
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:string</li> <li>• NonEmptyStringType</li> <li>• i18nString</li> </ul>									
Properties	<table> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded			
content:	complex									
minOccurs:	0									
maxOccurs:	unbounded									
Attributes	<table> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>xml:lang</b></td><td>union of(xs:language, restriction of xs:string)</td><td>optional</td></tr> <tr> <td></td><td colspan="2"> <pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;</pre> </td></tr> </tbody> </table>	QName	Type	Use	<b>xml:lang</b>	union of(xs:language, restriction of xs:string)	optional		<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;</pre>	
QName	Type	Use								
<b>xml:lang</b>	union of(xs:language, restriction of xs:string)	optional								
	<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;</pre>									

QName	Type	Use
	<pre> &lt;h4&gt;Notes&lt;/h4&gt; &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt; &lt;p&gt;See BCP 47 at &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc- editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at &lt;a href="http://www.iana.org/assignments/language-subtag- registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt; &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </pre>	
Source	<pre> &lt;xs:element name="value" type="i18nString" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Language translation as specified by the xml:lang attribute&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>	

## Element ParagraphType / itemizedlist

Namespace	No namespace
Annotations	A list of items in a text paragraph. The list is generally displayed as a bulleted list. This markup is a subset of DocBook.
Diagram	<pre> classDiagram     class itemizedlist {         Type         ListType     }     class ListType     class listitem      itemizedlist "1..oo" --&gt; ListType     ListType "*" --&gt; "1..oo" listitem     listitem &lt; -- itemizedlist </pre> <p>The diagram illustrates the UML class structure for the itemizedlist element. It shows three classes: itemizedlist, ListType, and listitem. The itemizedlist class has two associations: one with ListType (multiplicity 1..oo) and one with listitem (multiplicity *). The ListType class has a generalization relationship with itemizedlist (multiplicity 1..oo). The listitem class is a generalization of itemizedlist.</p>
Type	ListType
Properties	content: complex
Model	listitem+
Children	listitem
Instance	<pre> &lt;itemizedlist&gt;   &lt;listitem&gt;{1,unbounded}&lt;/listitem&gt; &lt;/itemizedlist&gt; </pre>
Source	<pre> &lt;xs:element name="itemizedlist" type="ListType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A list of items in a text paragraph. The list is generally displayed as a bulleted list. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>

## Element ListType / listitem

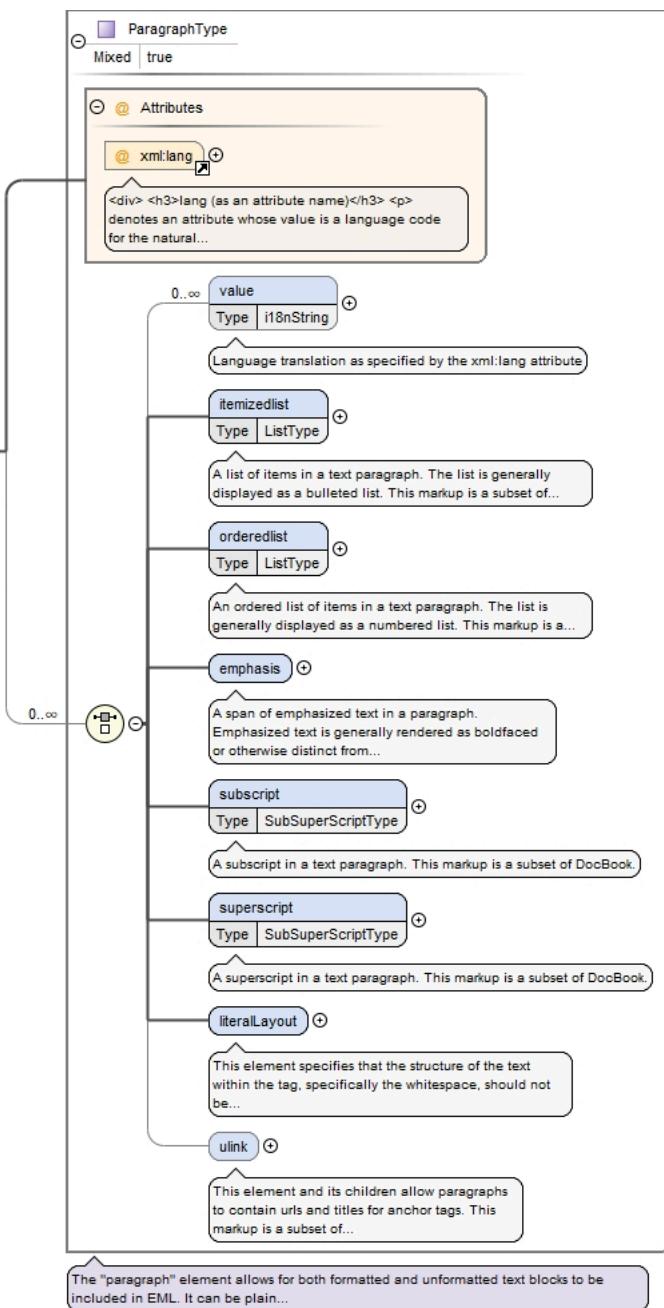
Namespace	No namespace
Annotations	An item in a list of items. Each list item is formatted as a bulleted or numbered item depending on the list type in which it resides. List items contain paragraphs which in turn can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.

Diagram	<pre> classDiagram     para &lt; -- listitem     para &lt; -- itemizedlist     para &lt; -- orderedlist     listitem *--o itemizedlist     listitem *--o orderedlist     listitem --o para     itemizedlist --o para     orderedlist --o para   </pre> <p>The diagram illustrates the schema structure for EML list elements. It shows four main types: <b>para</b>, <b>listitem</b>, <b>itemizedlist</b>, and <b>orderedlist</b>. <b>listitem</b> is a multiplicity of 1..oo and has associations with both <b>itemizedlist</b> and <b>orderedlist</b>. Both <b>itemizedlist</b> and <b>orderedlist</b> inherit from <b>para</b>. Each type has a detailed description box:</p> <ul style="list-style-type: none"> <li><b>para</b>: The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain...</li> <li><b>listitem</b>: An item in a list of items. Each list item is formatted as a bulleted or numbered item depending on the list type in...</li> <li><b>itemizedlist</b>: A list of items in a text paragraph. The list is generally displayed as a bulleted list. This markup is a subset of...</li> <li><b>orderedlist</b>: An ordered list of items in a text paragraph. The list is generally displayed as a numbered list. This markup is a...</li> </ul>						
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">1</td></tr> <tr> <td style="padding: 2px;">maxOccurs:</td><td style="padding: 2px;">unbounded</td></tr> </table>	content:	complex	minOccurs:	1	maxOccurs:	unbounded
content:	complex						
minOccurs:	1						
maxOccurs:	unbounded						
Model	para   itemizedlist   orderedlist						
Children	itemizedlist, orderedlist, para						
Instance	<pre> &lt;listitem&gt;   &lt;para xml:lang=""&gt;{1,1}&lt;/para&gt;   &lt;itemizedlist&gt;{1,1}&lt;/itemizedlist&gt;   &lt;orderedlist&gt;{1,1}&lt;/orderedlist&gt; &lt;/listitem&gt;   </pre>						
Source	<pre> &lt;xs:element name="listitem" minOccurs="1" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;An item in a list of items. Each list item is formatted as a bulleted or numbered item depending on the list type in which it resides. List items contain paragraphs which in turn can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:choice minOccurs="1" maxOccurs="unbounded"&gt;       &lt;xs:element name="para" type="ParagraphType"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="itemizedlist" type="ListType"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A list of items in a text paragraph. The list is generally displayed as a bulleted list. This markup is a subset of DocBook.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="orderedlist" type="ListType"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;An ordered list of items in a text paragraph. The list is generally displayed as a numbered list. This markup is a subset of DocBook.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:choice&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>						

## Element ListType / listitem / para

Namespace	No namespace
Annotations	<p>The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.</p>

## Diagram



Type	ParagraphType
Properties	content: complex mixed: true
Model	<code>value*   itemizedlist   orderedlist   emphasis   subscript   superscript   literalLayout   ulink {0,1}</code>
Children	emphasis, itemizedlist, literalLayout, orderedlist, subscript, superscript, ulink, value
Instance	<pre>&lt;para xml:lang=""&gt;   &lt;value xml:lang=""&gt;{0,unbounded}&lt;/value&gt;   &lt;itemizedlist&gt;{1,1}&lt;/itemizedlist&gt;   &lt;orderedlist&gt;{1,1}&lt;/orderedlist&gt;   &lt;emphasis xml:lang=""&gt;{1,1}&lt;/emphasis&gt;   &lt;subscript xml:lang=""&gt;{1,1}&lt;/subscript&gt;   &lt;superscript xml:lang=""&gt;{1,1}&lt;/superscript&gt;   &lt;literalLayout&gt;{1,1}&lt;/literalLayout&gt;   &lt;ulink url=""&gt;{0,1}&lt;/ulink&gt; &lt;/para&gt;</pre>

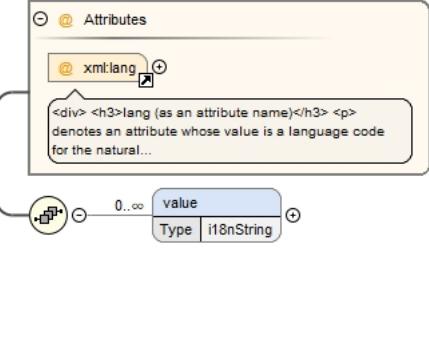
Attributes	QName	Type	Use
	xml:lang	union of(xs:language, restriction of xs:string)	optional
	<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc- editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag- registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt;</pre>		
Source	<pre>&lt;xs:element name="para" type="ParagraphType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

## Element ParagraphType / orderedlist

Namespace	No namespace
Annotations	An ordered list of items in a text paragraph. The list is generally displayed as a numbered list. This markup is a subset of DocBook.
Diagram	<pre> classDiagram     class orderedlist {         Type         ListType     }     class ListType {         1..oo listitem     }     orderedlist "1" -- "1..oo" ListType   </pre> <p>An ordered list of items in a text paragraph. The list is generally displayed as a numbered list. This markup is a subset of DocBook.</p> <p>A list of items in a text paragraph. The ListType is used by both orderedlist elements and itemizedlist elements. This...</p>
Type	ListType
Properties	content: complex
Model	listitem+
Children	listitem
Instance	<pre>&lt;orderedlist&gt;   &lt;listitem&gt;{1,unbounded}&lt;/listitem&gt; &lt;/orderedlist&gt;</pre>
Source	<pre>&lt;xs:element name="orderedlist" type="ListType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;An ordered list of items in a text paragraph. The list is generally displayed as a numbered list. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element ParagraphType / emphasis

Namespace	No namespace
Annotations	A span of emphasized text in a paragraph. Emphasized text is generally rendered as boldfaced or otherwise distinct from the surrounding text. This markup is a subset of DocBook.

Diagram										
Properties	<p>content: complex</p> <p>mixed: true</p>									
Model	value*									
Children	value									
Instance	<pre>&lt;emphasis xml:lang=""&gt;   &lt;value xml:lang=""&gt;{0 ,unbounded}&lt;/value&gt; &lt;/emphasis&gt;</pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td>xml:lang</td><td>union of(xs:language, restriction of xs:string)</td><td>optional</td></tr> <tr> <td></td><td> <p>&lt;div&gt;</p> <p style="padding-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p style="padding-left: 20px;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;See BCP 47 at</p> <p style="padding-left: 40px;"><a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a>and the IANA language subtag registry at</p> <p style="padding-left: 40px;"><a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a>for further information.&lt;/p&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> </td><td></td></tr> </tbody> </table>	QName	Type	Use	xml:lang	union of(xs:language, restriction of xs:string)	optional		<p>&lt;div&gt;</p> <p style="padding-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p style="padding-left: 20px;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;See BCP 47 at</p> <p style="padding-left: 40px;"><a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a>and the IANA language subtag registry at</p> <p style="padding-left: 40px;"><a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a>for further information.&lt;/p&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p>	
QName	Type	Use								
xml:lang	union of(xs:language, restriction of xs:string)	optional								
	<p>&lt;div&gt;</p> <p style="padding-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p style="padding-left: 20px;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;See BCP 47 at</p> <p style="padding-left: 40px;"><a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a>and the IANA language subtag registry at</p> <p style="padding-left: 40px;"><a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a>for further information.&lt;/p&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p>									
Source	<pre>&lt;xs:element name="emphasis"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A span of emphasized text in a paragraph. Emphasized text is generally rendered as boldfaced or otherwise distinct from the surrounding text. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType mixed="true"&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="value" type="i18nString" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute ref="xml:lang"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>									

## Element ParagraphType / emphasis / value

Namespace	No namespace
-----------	--------------

Diagram	<pre> classDiagram     class i18nString {         &lt;&lt;Base Type   NonEmptyStringType&gt;&gt;     }     class NonEmptyStringType {         &lt;&lt;Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...&gt;&gt;     }     class value {         &lt;&lt;Type   i18nString&gt;&gt;     }     i18nString &lt; -- NonEmptyStringType     NonEmptyStringType &lt; -- value     </pre> <p>The diagram illustrates the type hierarchy. It starts with the <b>i18nString</b> class, which is annotated with <code>&lt;&lt;Base Type   NonEmptyStringType&gt;&gt;</code>. An arrow points from <b>i18nString</b> to the <b>NonEmptyStringType</b> class, which is annotated with <code>&lt;&lt;Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...&gt;&gt;</code>. Another arrow points from <b>NonEmptyStringType</b> to the <b>value</b> class, which is annotated with <code>&lt;&lt;Type   i18nString&gt;&gt;</code>.</p>									
Type	i18nString									
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:string</li> <li>• NonEmptyStringType</li> <li>• i18nString</li> </ul>									
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded			
content:	complex									
minOccurs:	0									
maxOccurs:	unbounded									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>xml:lang</b></td> <td>union of(xs:language, restriction of xs:string)</td> <td>optional</td> </tr> <tr> <td></td> <td> <pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </pre> </td> <td></td> </tr> </tbody> </table>	QName	Type	Use	<b>xml:lang</b>	union of(xs:language, restriction of xs:string)	optional		<pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </pre>	
QName	Type	Use								
<b>xml:lang</b>	union of(xs:language, restriction of xs:string)	optional								
	<pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </pre>									
Source	<code>&lt;xs:element name="value" type="i18nString" minOccurs="0" maxOccurs="unbounded" /&gt;</code>									

## Element ParagraphType / subscript

Namespace	No namespace
Annotations	A subscript in a text paragraph. This markup is a subset of DocBook.

Diagram	<p>The diagram illustrates the UML class <code>SubSuperScriptType</code>. It is a <code>Mixed</code> type. It has an attribute <code>@xml:lang</code> of type <code>xs:string</code>, which is described as denoting an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification. The class has three children: <code>value</code> (0..∞), <code>subscript</code> (0..∞), and <code>superscript</code> (0..∞). Each of these children is also a <code>SubSuperScriptType</code>. A note states that a <code>subscript</code> or <code>superscript</code> in a text paragraph is a subset of DocBook.</p>												
Type	SubSuperScriptType												
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">mixed:</td><td style="padding: 2px;">true</td></tr> </table>	content:	complex	mixed:	true								
content:	complex												
mixed:	true												
Model	<code>value*   subscript   superscript</code>												
Children	subscript, superscript, value												
Instance	<pre> &lt;subscript xml:lang=""&gt;   &lt;value xml:lang=""&gt;{0,unbounded}&lt;/value&gt;   &lt;subscript xml:lang=""&gt;{1,1}&lt;/subscript&gt;   &lt;superscript xml:lang=""&gt;{1,1}&lt;/superscript&gt; &lt;/subscript&gt; </pre>												
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">QName</th><th style="width: 25%;">Type</th><th style="width: 25%;">Use</th><th style="width: 25%;"></th></tr> </thead> <tbody> <tr> <td><code>xml:lang</code></td><td>union of(xs:language, restriction of xs:string)</td><td>optional</td><td></td></tr> <tr> <td></td><td> <pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </pre> </td><td></td><td></td></tr> </tbody> </table>	QName	Type	Use		<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional			<pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </pre>		
QName	Type	Use											
<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional											
	<pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </pre>												
Source	<pre> &lt;xs:element name="subscript" type="SubSuperScriptType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A subscript in a text paragraph. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>												

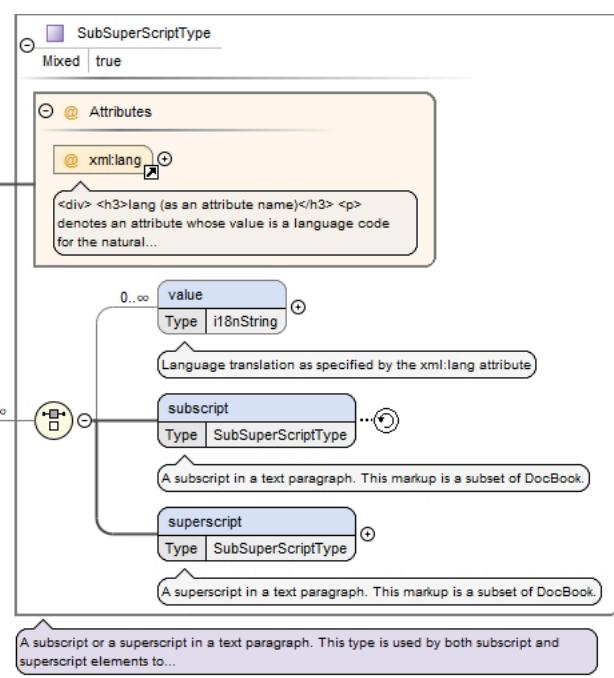
## Element SubSuperScriptType / value

Namespace	No namespace
-----------	--------------

Annotations	Language translation as specified by the xml:lang attribute						
Diagram	<p>The diagram illustrates the type hierarchy for the <code>i18nString</code> element. It starts with the <code>i18nString</code> class, which is a <code>NonEmptyStringType</code>. This type is described as a non-empty string with a specific content pattern. Below it is the <code>xs:string</code> class, which is a <code>NonEmptyStringType</code>. Finally, the <code>@xml:lang</code> attribute is shown, which is of type <code>xs:string</code>. A note explains that the <code>xml:lang</code> attribute denotes an attribute whose value is a language code for the natural language of the content.</p>						
Type	<code>i18nString</code>						
Type hierarchy	<ul style="list-style-type: none"> <li><code>xs:string</code> <ul style="list-style-type: none"> <li><code>NonEmptyStringType</code></li> <li><code>i18nString</code></li> </ul> </li> </ul>						
Properties	<table> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><code>xml:lang</code></td> <td>union of(<code>xs:language</code>, restriction of <code>xs:string</code>)</td> <td>optional</td> </tr> </tbody> </table> <p><code>&lt;div&gt;</code>  <code>&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</code>  <code>&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</code>  <code>&lt;/div&gt;</code>  <code>&lt;div&gt;</code>  <code>&lt;h4&gt;Notes&lt;/h4&gt;</code>  <code>&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</code>  <code>&lt;p&gt;See BCP 47 at</code>  <code>&lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;</code>and the IANA language subtag registry at  <code>&lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;</code>for further information.  <code>&lt;p&gt;The union allows for the 'un-declaration' of <code>xml:lang</code> with the empty string.&lt;/p&gt;</code>  <code>&lt;/div&gt;</code></p>	QName	Type	Use	<code>xml:lang</code>	union of( <code>xs:language</code> , restriction of <code>xs:string</code> )	optional
QName	Type	Use					
<code>xml:lang</code>	union of( <code>xs:language</code> , restriction of <code>xs:string</code> )	optional					
Source	<pre> &lt;xs:element name="value" type="i18nString" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Language translation as specified by the xml:lang attribute&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>						

## Element SubSuperScriptType / subscript

Namespace	No namespace
Annotations	A subscript in a text paragraph. This markup is a subset of DocBook.

Diagram									
Type	SubSuperScriptType								
Properties	<p>content: complex</p> <p>mixed: true</p>								
Model	value*   subscript   superscript								
Children	subscript, superscript, value								
Instance	<pre>&lt;subscript xml:lang=""&gt;   &lt;value xml:lang=""&gt;{0..unbounded}&lt;/value&gt;   &lt;subscript xml:lang=""&gt;{1..1}&lt;/subscript&gt;   &lt;superscript xml:lang=""&gt;{1..1}&lt;/superscript&gt; &lt;/subscript&gt;</pre>								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>xml:lang</td> <td>union of(xs:language, restriction of xs:string)</td> <td>optional</td> </tr> <tr> <td></td> <td> <p>&lt;div&gt;</p> <p style="padding-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p style="padding-left: 20px;">&lt;/div&gt;</p> <p style="margin-left: 20px;"><b>Notes</b></p> <p style="margin-left: 40px;">Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.</p> <p style="margin-left: 40px;">See BCP 47 at <a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a> and the IANA language subtag registry at <a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a> for further information.</p> <p style="margin-left: 40px;">The union allows for the 'un-declaration' of xml:lang with the empty string.</p> <p &gt;="" &lt;="" &lt;xs:annotation&gt;="" &lt;xs:documentation&gt;a="" a="" docbook.&lt;="" in="" is="" markup="" of="" paragraph.="" pre="" style="margin-left: 20px;&gt;&lt;/div&gt;&lt;/p&gt; &lt;/td&gt; &lt;td&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/tbody&gt; &lt;/table&gt; &lt;/td&gt;&lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Source&lt;/td&gt;&lt;td&gt; &lt;pre&gt;&lt;xs:element name=" subscript="" subscript"="" subset="" text="" this="" type="SubSuperScriptType" xs:annotation&gt;="" xs:documentation&gt;="" xs:element&gt;<=""> </p></td></tr> </tbody></table>	QName	Type	Use	xml:lang	union of(xs:language, restriction of xs:string)	optional		<p>&lt;div&gt;</p> <p style="padding-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p style="padding-left: 20px;">&lt;/div&gt;</p> <p style="margin-left: 20px;"><b>Notes</b></p> <p style="margin-left: 40px;">Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.</p> <p style="margin-left: 40px;">See BCP 47 at <a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a> and the IANA language subtag registry at <a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a> for further information.</p> <p style="margin-left: 40px;">The union allows for the 'un-declaration' of xml:lang with the empty string.</p> <p &gt;="" &lt;="" &lt;xs:annotation&gt;="" &lt;xs:documentation&gt;a="" a="" docbook.&lt;="" in="" is="" markup="" of="" paragraph.="" pre="" style="margin-left: 20px;&gt;&lt;/div&gt;&lt;/p&gt; &lt;/td&gt; &lt;td&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/tbody&gt; &lt;/table&gt; &lt;/td&gt;&lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Source&lt;/td&gt;&lt;td&gt; &lt;pre&gt;&lt;xs:element name=" subscript="" subscript"="" subset="" text="" this="" type="SubSuperScriptType" xs:annotation&gt;="" xs:documentation&gt;="" xs:element&gt;<=""> </p>
QName	Type	Use							
xml:lang	union of(xs:language, restriction of xs:string)	optional							
	<p>&lt;div&gt;</p> <p style="padding-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="padding-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p style="padding-left: 20px;">&lt;/div&gt;</p> <p style="margin-left: 20px;"><b>Notes</b></p> <p style="margin-left: 40px;">Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.</p> <p style="margin-left: 40px;">See BCP 47 at <a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a> and the IANA language subtag registry at <a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a> for further information.</p> <p style="margin-left: 40px;">The union allows for the 'un-declaration' of xml:lang with the empty string.</p> <p &gt;="" &lt;="" &lt;xs:annotation&gt;="" &lt;xs:documentation&gt;a="" a="" docbook.&lt;="" in="" is="" markup="" of="" paragraph.="" pre="" style="margin-left: 20px;&gt;&lt;/div&gt;&lt;/p&gt; &lt;/td&gt; &lt;td&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/tbody&gt; &lt;/table&gt; &lt;/td&gt;&lt;/tr&gt; &lt;tr&gt; &lt;td&gt;Source&lt;/td&gt;&lt;td&gt; &lt;pre&gt;&lt;xs:element name=" subscript="" subscript"="" subset="" text="" this="" type="SubSuperScriptType" xs:annotation&gt;="" xs:documentation&gt;="" xs:element&gt;<=""> </p>								

## Element SubSuperScriptType / superscript

Namespace	No namespace
-----------	--------------

Annotations	A superscript in a text paragraph. This markup is a subset of DocBook.									
Diagram	<pre> classDiagram     class SubSuperScriptType {         mixed true     }     class Attributes {         @ xml:lang     }     class value {         type i18nString     }     class subscript {         type SubSuperScriptType     }     class superscript {         type SubSuperScriptType     }      SubSuperScriptType &lt; -- Attributes     SubSuperScriptType &lt; -- value     SubSuperScriptType &lt; -- subscript     SubSuperScriptType &lt; -- superscript      Attributes &lt; -- xml:lang      value &lt; -- i18nString      subscript &lt; -- SubSuperScriptType     superscript &lt; -- SubSuperScriptType   </pre> <p>The diagram shows the UML class structure for <code>SubSuperScriptType</code>. It includes attributes for <code>xml:lang</code> (with a note about its meaning), <code>value</code> (of type <code>i18nString</code>), <code>subscript</code>, and <code>superscript</code>. The <code>subscript</code> and <code>superscript</code> elements inherit from <code>SubSuperScriptType</code>.</p>									
Type	<code>SubSuperScriptType</code>									
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>mixed:</td> <td>true</td> </tr> </table>	content:	complex	mixed:	true					
content:	complex									
mixed:	true									
Model	<code>value*   subscript   superscript</code>									
Children	<code>subscript, superscript, value</code>									
Instance	<pre> &lt;superscript xml:lang=""&gt;   &lt;value xml:lang=""&gt;{0,unbounded}&lt;/value&gt;   &lt;subscript xml:lang=""&gt;{1,1}&lt;/subscript&gt;   &lt;superscript xml:lang=""&gt;{1,1}&lt;/superscript&gt; &lt;/superscript&gt;   </pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><code>xml:lang</code></td> <td>union of(xs:language, restriction of xs:string)</td> <td>optional</td> </tr> <tr> <td></td> <td> <div> <p><code>&lt;div&gt;</code></p> <p><code>&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</code></p> <p><code>&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</code></p> <p><code>&lt;/div&gt;</code></p> <p><code>&lt;div&gt;</code></p> <p><code>&lt;h4&gt;Notes&lt;/h4&gt;</code></p> <p><code>&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</code></p> <p><code>&lt;p&gt;See BCP 47 at</code></p> <p><code>&lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;</code> and the IANA language subtag registry at</p> <p><code>&lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;</code> for further information.</p> <p><code>&lt;p&gt;The union allows for the 'un-declaration' of <code>xml:lang</code> with the empty string.&lt;/p&gt;</code></p> <p><code>&lt;/div&gt;</code></p> </div></td> <td></td> </tr> </tbody> </table>	QName	Type	Use	<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional		<div> <p><code>&lt;div&gt;</code></p> <p><code>&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</code></p> <p><code>&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</code></p> <p><code>&lt;/div&gt;</code></p> <p><code>&lt;div&gt;</code></p> <p><code>&lt;h4&gt;Notes&lt;/h4&gt;</code></p> <p><code>&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</code></p> <p><code>&lt;p&gt;See BCP 47 at</code></p> <p><code>&lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;</code> and the IANA language subtag registry at</p> <p><code>&lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;</code> for further information.</p> <p><code>&lt;p&gt;The union allows for the 'un-declaration' of <code>xml:lang</code> with the empty string.&lt;/p&gt;</code></p> <p><code>&lt;/div&gt;</code></p> </div>	
QName	Type	Use								
<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional								
	<div> <p><code>&lt;div&gt;</code></p> <p><code>&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</code></p> <p><code>&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</code></p> <p><code>&lt;/div&gt;</code></p> <p><code>&lt;div&gt;</code></p> <p><code>&lt;h4&gt;Notes&lt;/h4&gt;</code></p> <p><code>&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</code></p> <p><code>&lt;p&gt;See BCP 47 at</code></p> <p><code>&lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;</code> and the IANA language subtag registry at</p> <p><code>&lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;</code> for further information.</p> <p><code>&lt;p&gt;The union allows for the 'un-declaration' of <code>xml:lang</code> with the empty string.&lt;/p&gt;</code></p> <p><code>&lt;/div&gt;</code></p> </div>									
Source	<pre> &lt;xs:element name="superscript" type="SubSuperScriptType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A superscript in a text paragraph. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;   </pre>									

## Element ParagraphType / superscript

Namespace	No namespace									
Annotations	A superscript in a text paragraph. This markup is a subset of DocBook.									
Diagram	<p>The diagram illustrates the UML class structure for the <code>SubSuperScriptType</code>. It shows inheritance from <code>ParagraphType</code> (indicated by a line with an open circle) and the declaration of attributes and children. The <code>@xml:lang</code> attribute is described as an attribute whose value is a language code for the natural language of the content. The <code>value</code> child is described as language translation as specified by the <code>xml:lang</code> attribute. The <code>subscript</code> and <code>superscript</code> children are both described as superscripts in a text paragraph, being subsets of DocBook. A general note at the bottom states that this type is used by both <code>subscript</code> and <code>superscript</code> elements.</p>									
Type	SubSuperScriptType									
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>mixed:</td> <td>true</td> </tr> </table>	content:	complex	mixed:	true					
content:	complex									
mixed:	true									
Model	<code>value*   subscript   superscript</code>									
Children	subscript, superscript, value									
Instance	<pre>&lt;superscript xml:lang=""&gt;   &lt;value xml:lang=""&gt;{0,unbounded}&lt;/value&gt;   &lt;subscript xml:lang=""&gt;{1,1}&lt;/subscript&gt;   &lt;superscript xml:lang=""&gt;{1,1}&lt;/superscript&gt; &lt;/superscript&gt;</pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><code>xml:lang</code></td> <td>union of(xs:language, restriction of xs:string)</td> <td>optional</td> </tr> <tr> <td></td> <td> <div style="background-color: #f0f0f0; padding: 10px;"> <p><code>&lt;div&gt;</code></p> <p style="margin-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;/div&gt;</p> <p style="margin-left: 20px;"><code>&lt;div&gt;</code></p> <p style="margin-left: 40px;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="margin-left: 40px;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="margin-left: 40px;">&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 60px;"><a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a> and the IANA language subtag registry at</p> <p style="margin-left: 60px;"><a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a> for further information.&lt;/p&gt;</p> <p style="margin-left: 40px;">&lt;p&gt;The union allows for the 'un-declaration' of <code>xml:lang</code> with the empty string.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;/div&gt;</p> </div> </td> <td></td> </tr> </tbody> </table>	QName	Type	Use	<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional		<div style="background-color: #f0f0f0; padding: 10px;"> <p><code>&lt;div&gt;</code></p> <p style="margin-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;/div&gt;</p> <p style="margin-left: 20px;"><code>&lt;div&gt;</code></p> <p style="margin-left: 40px;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="margin-left: 40px;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="margin-left: 40px;">&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 60px;"><a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a> and the IANA language subtag registry at</p> <p style="margin-left: 60px;"><a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a> for further information.&lt;/p&gt;</p> <p style="margin-left: 40px;">&lt;p&gt;The union allows for the 'un-declaration' of <code>xml:lang</code> with the empty string.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;/div&gt;</p> </div>	
QName	Type	Use								
<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional								
	<div style="background-color: #f0f0f0; padding: 10px;"> <p><code>&lt;div&gt;</code></p> <p style="margin-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;/div&gt;</p> <p style="margin-left: 20px;"><code>&lt;div&gt;</code></p> <p style="margin-left: 40px;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="margin-left: 40px;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="margin-left: 40px;">&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 60px;"><a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a> and the IANA language subtag registry at</p> <p style="margin-left: 60px;"><a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a> for further information.&lt;/p&gt;</p> <p style="margin-left: 40px;">&lt;p&gt;The union allows for the 'un-declaration' of <code>xml:lang</code> with the empty string.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;/div&gt;</p> </div>									
Source	<pre>&lt;xsd:element name="superscript" type="SubSuperScriptType"&gt;   &lt;xsd:annotation&gt;     &lt;xsd:documentation&gt;A superscript in a text paragraph. This markup is a subset of DocBook.&lt;/xsd:documentation&gt;   &lt;/xsd:annotation&gt; &lt;/xsd:element&gt;</pre>									

## Element ParagraphType / literalLayout

Namespace	No namespace
Annotations	This element specifies that the structure of the text within the tag, specifically the whitespace, should not be altered.
Diagram	<pre> classDiagram     class literalLayout {         &lt;&lt;This element specifies that the structure of the text within the tag, specifically the whitespace, should not be...&gt;&gt;     }     class value {         &lt;&lt;Type i18nString&gt;&gt;     }     literalLayout "0..infinity" --&gt; value   </pre>
Properties	<p>content: complex</p> <p>mixed: true</p>
Model	value*
Children	value
Instance	<pre> &lt;literalLayout&gt;   &lt;value xml:lang=""&gt;{0,unbounded}&lt;/value&gt; &lt;/literalLayout&gt;   </pre>
Source	<pre> &lt;xs:element name="literalLayout"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies that the structure of the text within the tag, specifically the whitespace, should not be altered.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType mixed="true"&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="value" type="i18nString" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>

## Element ParagraphType / literalLayout / value

Namespace	No namespace									
Diagram	<pre> classDiagram     class value {         &lt;&lt;Type i18nString&gt;&gt;     }     class i18nString {         &lt;&lt;Base Type NonEmptyStringType&gt;&gt;     }     value "0..infinity" --&gt; i18nString   </pre>									
Type	i18nString									
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:string</li> <li>• NonEmptyStringType</li> <li>• i18nString</li> </ul>									
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>xml:lang</td> <td>union of(xs:language, restriction of xs:string)</td> <td>optional</td> </tr> <tr> <td></td> <td>&lt;div&gt; &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</td> <td></td> </tr> </tbody> </table>	QName	Type	Use	xml:lang	union of(xs:language, restriction of xs:string)	optional		<div> <h3>lang (as an attribute name)</h3>	
QName	Type	Use								
xml:lang	union of(xs:language, restriction of xs:string)	optional								
	<div> <h3>lang (as an attribute name)</h3>									

QName	Type	Use
		<p>&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p style="padding-left: 2em;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="padding-left: 2em;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="padding-left: 2em;">&lt;p&gt;See BCP 47 at  <a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a> and the IANA language subtag registry at  <a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a> for further information.&lt;/p&gt;</p> <p style="padding-left: 2em;">&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p>
Source	<xs:element name="value" type="i18nString" minOccurs="0" maxOccurs="unbounded" />	

## Element ParagraphType / ulink

Namespace	No namespace									
Annotations	This element and its children allow paragraphs to contain urls and titles for anchor tags. This markup is a subset of DocBook.									
Diagram	<pre> classDiagram     class ulink {         @url : i18nString         &lt;&lt;This element and its children allow paragraphs to contain urls and titles for anchor tags. This markup is a subset of DocBook.&gt;&gt;     }     class citetitle {         Type : i18nString     }     ulink "0..oo" --o citetitle     note over ulink: This element and its children allow paragraphs to contain urls and titles for anchor tags. This markup is a subset of DocBook.     note over @url: The url attribute contains the location of the work for a link. This markup is a subset of DocBook.     note over citetitle: The Dublin Core Metadata Initiative   </pre>									
Properties	content: complex minOccurs: 0 mixed: true									
Model	citetitle									
Children	citetitle									
Instance	<pre> &lt;ulink url=""&gt;   &lt;citetitle xml:lang=""&gt;{1,1}&lt;/citetitle&gt; &lt;/ulink&gt;   </pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>url</td> <td></td> <td>optional</td> </tr> <tr> <td></td> <td colspan="2">The url attribute contains the location of the work for a link. This markup is a subset of DocBook.</td></tr> </tbody> </table>	QName	Type	Use	url		optional		The url attribute contains the location of the work for a link. This markup is a subset of DocBook.	
QName	Type	Use								
url		optional								
	The url attribute contains the location of the work for a link. This markup is a subset of DocBook.									
Source	<pre> &lt;xs:element name="ulink" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element and its children allow paragraphs to contain urls and titles for anchor tags. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType mixed="true"&gt;     &lt;xs:sequence minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:element name="citetitle" type="i18nString"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The Dublin Core Metadata Initiative&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="url" use="optional"&gt;       &lt;xs:annotation&gt;     </pre>									

```

<xs:documentation>The url attribute contains the location of the work for a link. This
markup is a subset of DocBook.</xs:documentation>
  </xs:annotation>
  </xs:attribute>
</xs:complexType>
</xs:elements>

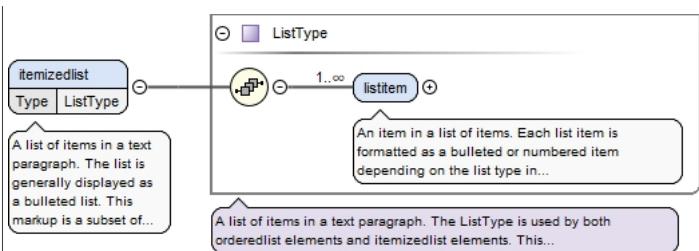
```

## Element ParagraphType / ulink / citetitle

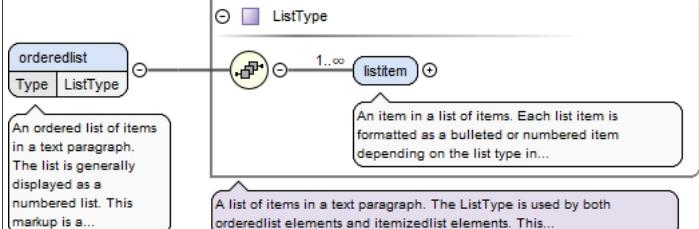
Namespace	No namespace									
Annotations	The Dublin Core Metadata Initiative									
Diagram	<p>The diagram illustrates the inheritance path of the <code>citetitle</code> element. It starts with <code>citetitle</code> (Type: <code>i18nString</code>) which is a specialization of <code>i18nString</code>. <code>i18nString</code> is a base type of <code>NonEmptyStringType</code>, which is itself a specialization of <code>NonEmptyStringType</code>. A callout box provides a detailed description of the <code>NonEmptyStringType</code>: "Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...". Another callout box for the <code>@xml:lang</code> attribute states: "&lt;div&gt; &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt; &lt;p&gt;denotes an attribute whose value is a language code for the natural...".</p>									
Type	<code>i18nString</code>									
Type hierarchy	<ul style="list-style-type: none"> <li>• <code>xs:string</code> <ul style="list-style-type: none"> <li>• <code>NonEmptyStringType</code></li> <li>• <code>i18nString</code></li> </ul> </li> </ul>									
Properties	content: complex									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><code>xml:lang</code></td> <td>union of(<code>xs:language</code>, restriction of <code>xs:string</code>)</td> <td>optional</td> </tr> <tr> <td></td> <td colspan="2"> <p>&lt;div&gt;</p> <p style="margin-left: 2em;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p style="margin-left: 2em;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 3em;">&lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at</p> <p style="margin-left: 3em;">&lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;The union allows for the 'un-declaration' of <code>xml:lang</code> with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> </td> </tr> </tbody> </table>	QName	Type	Use	<code>xml:lang</code>	union of( <code>xs:language</code> , restriction of <code>xs:string</code> )	optional		<p>&lt;div&gt;</p> <p style="margin-left: 2em;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p style="margin-left: 2em;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 3em;">&lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at</p> <p style="margin-left: 3em;">&lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;The union allows for the 'un-declaration' of <code>xml:lang</code> with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p>	
QName	Type	Use								
<code>xml:lang</code>	union of( <code>xs:language</code> , restriction of <code>xs:string</code> )	optional								
	<p>&lt;div&gt;</p> <p style="margin-left: 2em;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p style="margin-left: 2em;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 3em;">&lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at</p> <p style="margin-left: 3em;">&lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;</p> <p style="margin-left: 2em;">&lt;p&gt;The union allows for the 'un-declaration' of <code>xml:lang</code> with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p>									
Source	<pre> &lt;xs:element name="citetitle" type="i18nString"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The Dublin Core Metadata Initiative&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>									

## Element ListType / listitem / itemizedlist

Namespace	No namespace
Annotations	A list of items in a text paragraph. The list is generally displayed as a bulleted list. This markup is a subset of DocBook.

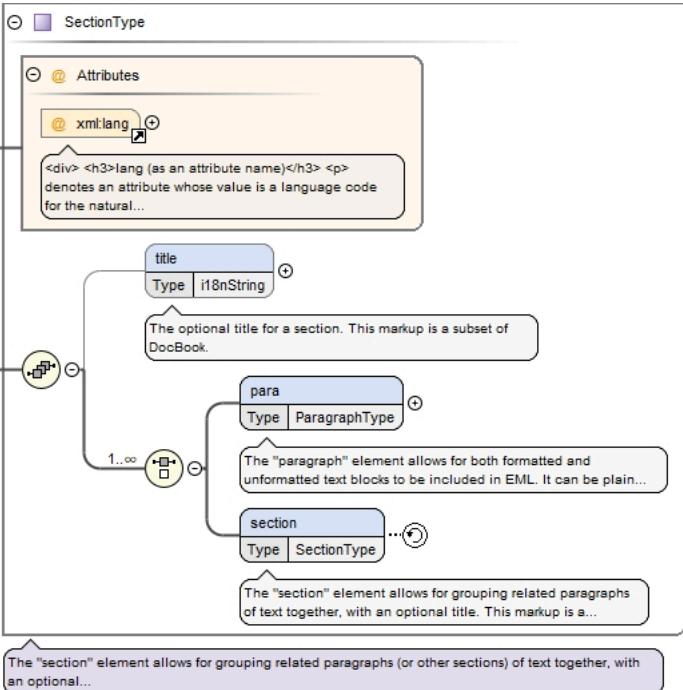
Diagram	
Type	ListType
Properties	content: complex
Model	listitem+
Children	listitem
Instance	<pre>&lt;itemizedlist&gt;   &lt;listitem&gt;{1,unbounded}&lt;/listitem&gt; &lt;/itemizedlist&gt;</pre>
Source	<pre>&lt;xss:element name="itemizedlist" type="ListType"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;A list of items in a text paragraph. The list is generally displayed as a bulleted list. This markup is a subset of DocBook.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>

### Element ListType / listitem / orderedlist

Namespace	No namespace
Annotations	An ordered list of items in a text paragraph. The list is generally displayed as a numbered list. This markup is a subset of DocBook.
Diagram	
Type	ListType
Properties	content: complex
Model	listitem+
Children	listitem
Instance	<pre>&lt;orderedlist&gt;   &lt;listitem&gt;{1,unbounded}&lt;/listitem&gt; &lt;/orderedlist&gt;</pre>
Source	<pre>&lt;xss:element name="orderedlist" type="ListType"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;An ordered list of items in a text paragraph. The list is generally displayed as a numbered list. This markup is a subset of DocBook.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>

### Element SectionType / section

Namespace	No namespace
Annotations	The "section" element allows for grouping related paragraphs of text together, with an optional title. This markup is a subset of DocBook.

Diagram										
Type	SectionType									
Properties	content: complex									
Model	title{0,1} , (para   section)									
Children	para, section, title									
Instance	<pre>&lt;section xml:lang=""&gt;   &lt;title xml:lang=""&gt;{0,1}&lt;/title&gt;   &lt;para xml:lang=""&gt;{1,1}&lt;/para&gt;   &lt;section xml:lang=""&gt;{1,1}&lt;/section&gt; &lt;/section&gt;</pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>xml:lang</td> <td>union of(xs:language, restriction of xs:string)</td> <td>optional</td> </tr> <tr> <td></td> <td> <div> <p>&lt;div&gt;</p> <p style="margin-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p style="margin-left: 20px;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 40px;"><a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a>and the IANA language subtag registry at</p> <p style="margin-left: 40px;"><a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a>for further information.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> </div> </td> <td></td> </tr> </tbody> </table>	QName	Type	Use	xml:lang	union of(xs:language, restriction of xs:string)	optional		<div> <p>&lt;div&gt;</p> <p style="margin-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p style="margin-left: 20px;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 40px;"><a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a>and the IANA language subtag registry at</p> <p style="margin-left: 40px;"><a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a>for further information.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> </div>	
QName	Type	Use								
xml:lang	union of(xs:language, restriction of xs:string)	optional								
	<div> <p>&lt;div&gt;</p> <p style="margin-left: 20px;">&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p style="margin-left: 20px;">&lt;h4&gt;Notes&lt;/h4&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 40px;"><a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt">http://www.rfc-editor.org/rfc/bcp/bcp47.txt</a>and the IANA language subtag registry at</p> <p style="margin-left: 40px;"><a href="http://www.iana.org/assignments/language-subtag-registry">http://www.iana.org/assignments/language-subtag-registry</a>for further information.&lt;/p&gt;</p> <p style="margin-left: 20px;">&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> </div>									
Source	<pre>&lt;xs:element name="section" type="SectionType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The "section" element allows for grouping related paragraphs of text together, with an optional title. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>									

## Element TextType / para

Namespace	No namespace
-----------	--------------

Annotations	The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.								
Diagram	<pre> classDiagram     class para {         &lt;&lt;The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain...&gt;&gt;     }     class ParagraphType {         &lt;&lt;Mixed true&gt;&gt;         &lt;&lt;@ Attributes&lt;/&gt;&gt;         &lt;&lt;@ xml:lang&lt;/&gt;&gt;         &lt;&lt;&lt;div&gt; &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt; &lt;p&gt; denotes an attribute whose value is a language code for the natural...&gt;&gt;         &lt;&lt;0..oo value&lt;/&gt;&gt;         &lt;&lt;Type   I18nString&lt;/&gt;&gt;         &lt;&lt;Language translation as specified by the xml:lang attribute&lt;/&gt;&gt;         &lt;&lt;itemizedlist&lt;/&gt;&gt;         &lt;&lt;Type   ListType&lt;/&gt;&gt;         &lt;&lt;A list of items in a text paragraph. The list is generally displayed as a bulleted list. This markup is a subset of...&gt;&gt;         &lt;&lt;orderedlist&lt;/&gt;&gt;         &lt;&lt;Type   ListType&lt;/&gt;&gt;         &lt;&lt;An ordered list of items in a text paragraph. The list is generally displayed as a numbered list. This markup is a ...&gt;&gt;         &lt;&lt;emphasis&lt;/&gt;&gt;         &lt;&lt;A span of emphasized text in a paragraph. Emphasized text is generally rendered as boldfaced or otherwise distinct from...&gt;&gt;         &lt;&lt;subscript&lt;/&gt;&gt;         &lt;&lt;Type   SubSuperScriptType&lt;/&gt;&gt;         &lt;&lt;A subscript in a text paragraph. This markup is a subset of DocBook.&gt;&gt;         &lt;&lt;superscript&lt;/&gt;&gt;         &lt;&lt;Type   SubSuperScriptType&lt;/&gt;&gt;         &lt;&lt;A superscript in a text paragraph. This markup is a subset of DocBook.&gt;&gt;         &lt;&lt;literalLayout&lt;/&gt;&gt;         &lt;&lt;This element specifies that the structure of the text within the tag, specifically the whitespace, should not be...&gt;&gt;         &lt;&lt;ulink&lt;/&gt;&gt;         &lt;&lt;This element and its children allow paragraphs to contain urls and titles for anchor tags. This markup is a subset of...&gt;&gt;     }     para --&gt; ParagraphType     ParagraphType &lt; -- value     ParagraphType &lt; -- itemizedlist     ParagraphType &lt; -- orderedlist     ParagraphType &lt; -- emphasis     ParagraphType &lt; -- subscript     ParagraphType &lt; -- superscript     ParagraphType &lt; -- literalLayout     ParagraphType &lt; -- ulink   </pre> <p>The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain...</p>								
Type	ParagraphType								
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> <tr> <td>mixed:</td> <td>true</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded	mixed:	true
content:	complex								
minOccurs:	0								
maxOccurs:	unbounded								
mixed:	true								
Model	value*   itemizedlist   orderedlist   emphasis   subscript   superscript   literalLayout   ulink {0,1}								
Children	emphasis, itemizedlist, literalLayout, orderedlist, subscript, superscript, ulink, value								
Instance	<pre> &lt;para xml:lang=""&gt;   &lt;value xml:lang=""&gt;{0,unbounded}&lt;/value&gt;   &lt;itemizedlist&gt;{1,1}&lt;/itemizedlist&gt;   &lt;orderedlist&gt;{1,1}&lt;/orderedlist&gt;   &lt;emphasis xml:lang=""&gt;{1,1}&lt;/emphasis&gt;   &lt;subscript xml:lang=""&gt;{1,1}&lt;/subscript&gt;   &lt;superscript xml:lang=""&gt;{1,1}&lt;/superscript&gt; </pre>								

	<pre>&lt;literalLayout&gt;{1,1}&lt;/literalLayout&gt; &lt;ulink url=""&gt;{0,1}&lt;/ulink&gt; &lt;/para&gt;</pre>		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>xml:lang</b>	union of(xs:language, restriction of xs:string)	optional
		<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt;</pre>	
Source	<pre>&lt;xs:element name="para" type="ParagraphType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

## Element keywordSet

Namespace	No namespace
Annotations	A wrapper element for the keyword and keywordThesaurus elements
Diagram	<pre> classDiagram     class keywordSet     class keyword {         Type NonEmptyStringType     }     class keywordThesaurus {         Type NonEmptyStringType     }      keywordSet "1..oo" -- "1..oo" keyword :      keyword "1..oo" -- "1..oo" keywordSet :    </pre>
Properties	content: complex
Used by	Element dataset
Model	keyword+ , keywordThesaurus
Children	keyword, keywordThesaurus
Instance	<pre>&lt;keywordSet&gt;   &lt;keyword&gt;{1,unbounded}&lt;/keyword&gt;   &lt;keywordThesaurus&gt;{1,1}&lt;/keywordThesaurus&gt; &lt;/keywordSet&gt;</pre>
Source	<pre>&lt;xs:element name="keywordSet"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A wrapper element for the keyword and keywordThesaurus elements&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="keyword" type="NonEmptyStringType" maxOccurs="unbounded"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This field names a keyword or key phrase that concisely describes the resource or is related to the resource. Each keyword field should contain one and only one keyword&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="keywordThesaurus" type="NonEmptyStringType"&gt;</pre>

```

<xs:annotation>
  <xs:documentation>The name of the official keyword thesaurus from which keyword was
derived</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

```

## Element keywordSet / keyword

Namespace	No namespace				
Annotations	This field names a keyword or key phrase that concisely describes the resource or is related to the resource. Each keyword field should contain one and only one keyword				
Diagram	<pre> classDiagram     class keyword {         &lt;&lt;NonEmptyStringType&gt;&gt;     }     NonEmptyStringType "1" -- "1" keyword   </pre> <p>This field names a keyword or key phrase that concisely describes the resource or is related to the resource. Each...</p> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p>				
Type	NonEmptyStringType				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	simple	maxOccurs:	unbounded
content:	simple				
maxOccurs:	unbounded				
Facets	<table> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ]*[ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ]*[ \s \S ]*
minLength	1				
pattern	[ \s ]*[ \S ]*[ \s \S ]*				
Source	<pre> &lt;xs:element name="keyword" type="NonEmptyStringType" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This field names a keyword or key phrase that concisely describes the resource or is related to the resource. Each keyword field should contain one and only one keyword&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>				

## Element keywordSet / keywordThesaurus

Namespace	No namespace				
Annotations	The name of the official keyword thesaurus from which keyword was derived				
Diagram	<pre> classDiagram     class keywordThesaurus {         &lt;&lt;NonEmptyStringType&gt;&gt;     }     NonEmptyStringType "1" -- "1" keywordThesaurus   </pre> <p>The name of the official keyword thesaurus from which keyword was derived</p> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p>				
Type	NonEmptyStringType				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> </table>	content:	simple		
content:	simple				
Facets	<table> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ]*[ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ]*[ \s \S ]*
minLength	1				
pattern	[ \s ]*[ \S ]*[ \s \S ]*				
Source	<pre> &lt;xs:element name="keywordThesaurus" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The name of the official keyword thesaurus from which keyword was derived&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>				

## Element additionalInfo

Namespace	No namespace
Annotations	Any information that is not characterized by the other resource metadata fields
Diagram	<pre> classDiagram     class additionalInfo {         &lt;&lt;para&gt;&gt;     }     para "1" -- "1" additionalInfo   </pre> <p>Any information that is not characterized by the other resource metadata fields</p> <p>The "paragraph" element allows for text blocks to be included in EML.</p>

Properties	content:	complex
Used by	Element	dataset
Model	para	
Children	para	
Instance	<pre>&lt;additionalInfo&gt;   &lt;para&gt;{1,1}&lt;/para&gt; &lt;/additionalInfo&gt;</pre>	
Source	<pre>&lt;x:element name="additionalInfo"&gt;   &lt;x:annotation&gt;     &lt;x:documentation&gt;Any information that is not characterized by the other resource metadata fields&lt;/x:documentation&gt;   &lt;/x:annotation&gt;   &lt;x:complexType&gt;     &lt;x:sequence&gt;       &lt;x:element ref="para"/&gt;     &lt;/x:sequence&gt;   &lt;/x:complexType&gt; &lt;/x:element&gt;</pre>	

## Element para

Namespace	No namespace
Annotations	The "paragraph" element allows for text blocks to be included in EML.
Diagram	<p>The "paragraph" element allows for text blocks to be included in EML.</p> <p>this element and its children allow paragraphs to contain urls and titles for anchor tags.</p>
Properties	<p>content: complex</p> <p>mixed: true</p>
Used by	Elements additionalInfo, description, intellectualRights, methods/sampling/samplingDescription, project-Type/funding
Model	ulink{0,1}
Children	ulink
Instance	<pre>&lt;para&gt;   &lt;ulink url=""&gt;{0,1}&lt;/ulink&gt; &lt;/para&gt;</pre>
Source	<pre>&lt;x:element name="para"&gt;   &lt;x:annotation&gt;     &lt;x:documentation&gt;The "paragraph" element allows for text blocks to be included in EML.&lt;/x:documentation&gt;   &lt;/x:annotation&gt;   &lt;x:complexType mixed="true"&gt;     &lt;x:sequence&gt;       &lt;x:element ref="ulink" minOccurs="0" maxOccurs="1"/&gt;     &lt;/x:sequence&gt;   &lt;/x:complexType&gt; &lt;/x:element&gt;</pre>

## Element ulink

Namespace	No namespace
Annotations	this element and its children allow paragraphs to contain urls and titles for anchor tags.

Diagram	<p>The diagram illustrates the <code>ulink</code> element structure. It has two attributes: <code>url</code> (NonEmptyStringType) which contains the location of the work for a link, and <code>citetitle</code> (NonEmptyStringType) which contains a text title for the url, displayed in an anchor tag.</p>												
Properties	<p>content: complex mixed: true</p>												
Used by	<p>Element para</p>												
Model	<p>citetitle</p>												
Children	<p>citetitle</p>												
Instance	<pre>&lt;ulink url=""&gt;   &lt;citetitle&gt;{1,1}&lt;/citetitle&gt; &lt;/ulink&gt;</pre>												
Attributes	<table border="1"> <thead> <tr> <th data-bbox="306 864 600 893">QName</th><th data-bbox="600 864 917 893">Type</th><th data-bbox="917 864 1076 893">Use</th><th data-bbox="1076 864 1432 893"></th></tr> </thead> <tbody> <tr> <td data-bbox="306 893 600 923"><code>url</code></td><td data-bbox="600 893 917 923">NonEmptyStringType</td><td data-bbox="917 893 1076 923">required</td><td data-bbox="1076 893 1432 923"></td></tr> <tr> <td data-bbox="306 923 600 952"></td><td data-bbox="600 923 917 952"></td><td data-bbox="917 923 1076 952"></td><td data-bbox="1076 923 1432 952">the url attribute contains the location of the work for a link.</td></tr> </tbody> </table>	QName	Type	Use		<code>url</code>	NonEmptyStringType	required					the url attribute contains the location of the work for a link.
QName	Type	Use											
<code>url</code>	NonEmptyStringType	required											
			the url attribute contains the location of the work for a link.										
Source	<pre>&lt;xsd:element name="ulink"&gt;   &lt;xsd:annotation&gt;     &lt;xsd:documentation&gt;this element and its children allow paragraphs to contain urls and titles for anchor tags.&lt;/xsd:documentation&gt;   &lt;/xsd:annotation&gt;   &lt;xsd:complexType mixed="true"&gt;     &lt;xsd:sequence&gt;       &lt;xsd:element name="citetitle" type="NonEmptyStringType"&gt;         &lt;xsd:annotation&gt;           &lt;xsd:documentation&gt;the citetitle element contains a text title for the url. It can be displayed in an anchor tag.&lt;/xsd:documentation&gt;         &lt;/xsd:annotation&gt;       &lt;/xsd:element&gt;     &lt;/xsd:sequence&gt;     &lt;xsd:attribute name="url" use="required" type="NonEmptyStringType"&gt;       &lt;xsd:annotation&gt;         &lt;xsd:documentation&gt;the url attribute contains the location of the work for a link.&lt;/xsd:documentation&gt;       &lt;/xsd:annotation&gt;     &lt;/xsd:attribute&gt;   &lt;/xsd:complexType&gt; &lt;/xsd:element&gt;</pre>												

## Element `ulink / citetitle`

Namespace	No namespace
Annotations	the citetitle element contains a text title for the url. It can be displayed in an anchor tag.
Diagram	<p>The diagram shows the <code>citetitle</code> element with type <code>NonEmptyStringType</code>. A note explains that this type specifies a content pattern for all elements required by EML.</p>
Type	NonEmptyStringType
Properties	<p>content: simple</p>
Facets	<p>minLength 1 pattern <code>[\s]*[\S][\s\S]*</code></p>
Source	<pre>&lt;xsd:element name="citetitle" type="NonEmptyStringType"&gt;   &lt;xsd:annotation&gt;</pre>

```

<xs:documentation>the citetitle element contains a text title for the url. It can be displayed
in an anchor tag.</xs:documentation>
</xs:annotation>
</xs:element>

```

## Element intellectualRights

Namespace	No namespace
Annotations	Contain a rights management statement for the resource, or reference a service providing such information
Diagram	<p>The diagram illustrates the structure of the <code>intellectualRights</code> element. It consists of a main node labeled <code>intellectualRights</code> connected by an arrow to another node labeled <code>para</code>. A callout box associated with the <code>intellectualRights</code> node contains the text: "Contain a rights management statement for the resource, or reference a service providing such information". Another callout box associated with the <code>para</code> node contains the text: "The \"paragraph\" element allows for text blocks to be included in EML."</p>
Properties	content: complex
Used by	Element dataset
Model	para
Children	para
Instance	<pre> &lt;intellectualRights&gt;   &lt;para&gt;{1,1}&lt;/para&gt; &lt;/intellectualRights&gt; </pre>
Source	<pre> &lt;xs:element name="intellectualRights"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Contain a rights management statement for the resource, or reference a service providing such information&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="para" /&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

## Element licensed

Namespace	No namespace
Annotations	This element provides information on how the resource is licensed and what rights may be available to users. GBIF only supports the following licenses CC0, CC BY and CC BY-NC. By default, the license provided here applies to all the dataset records.
Diagram	<p>The diagram shows the <code>licensed</code> element with three child nodes: <code>licenseName</code>, <code>url</code>, and <code>identifier</code>. A callout box for <code>licenseName</code> states: "The official name of a license that applies to the data and metadata described in this metadata record." A callout box for <code>url</code> states: "The persistent URL for the license, typically a SPDX URL, or an official URL from another well-known license vocabulary." A callout box for <code>identifier</code> states: "The official identifier for the license, which should be drawn from the SPDX license vocabulary, or a similar..."</p>
Properties	content: complex
Used by	Element dataset
Model	licenseName , url{0,1} , identifier{0,1}
Children	identifier, licenseName, url
Instance	<pre> &lt;licensed&gt;   &lt;licenseName&gt;{1,1}&lt;/licenseName&gt;   &lt;url&gt;{0,1}&lt;/url&gt;   &lt;identifier&gt;{0,1}&lt;/identifier&gt; </pre>

	</licensed>
Source	<pre> &lt;xs:element name="licensed"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element provides information on how the resource is licensed and what rights may be available to users. GBIF only supports the following licenses CC0, CC BY and CC BY-NC. By default, the license provided here applies to all the dataset records.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="licenseName" type="NonEmptyStringType" minOccurs="1"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The official name of a license that applies to the data and metadata described in this metadata record.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="url" type="xs:anyURI" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The persistent URL for the license, typically a SPDX URL, or an official URL from another well-known license vocabulary.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="identifier" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The official identifier for the license, which should be drawn from the SPDX license vocabulary, or a similar well-known license vocabulary.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

## Element licensed / licenseName

Namespace	No namespace				
Annotations	The official name of a license that applies to the data and metadata described in this metadata record.				
Diagram	<pre> classDiagram     class licenseName {         &lt;&lt;NonEmptyStringType&gt;&gt;     }     licenseName &lt; -- NonEmptyStringType   </pre> <p>The diagram shows a class named "licenseName" with a constraint "NonEmptyStringType". A callout box points to the constraint with the text: "The official name of a license that applies to the data and metadata described in this metadata record." Another callout box points to the constraint with the text: "Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that..."</p>				
Type	NonEmptyStringType				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	1
content:	simple				
minOccurs:	1				
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ]*[ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ]*[ \s \S ]*
minLength	1				
pattern	[ \s ]*[ \S ]*[ \s \S ]*				
Source	<pre> &lt;xs:element name="licenseName" type="NonEmptyStringType" minOccurs="1"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The official name of a license that applies to the data and metadata described in this metadata record.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>				

## Element licensed / url

Namespace	No namespace				
Annotations	The persistent URL for the license, typically a SPDX URL, or an official URL from another well-known license vocabulary.				
Diagram	<pre> classDiagram     class url {         &lt;&lt;xs:anyURI&gt;&gt;     }     url &lt; -- xs:anyURI   </pre> <p>The diagram shows a class named "url" with a constraint "xs:anyURI". A callout box points to the constraint with the text: "The persistent URL for the license, typically a SPDX URL, or an official URL from another well-known license vocabulary." Another callout box points to the constraint with the text: "Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI)."</p>				
Type	xs:anyURI				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				

Source	<pre>&lt;xs:element name="url" type="xs:anyURI" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The persistent URL for the license, typically a SPDX URL, or an official URL from another well-known license vocabulary.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>
--------	--

## Element licensed / identifier

Namespace	No namespace				
Annotations	The official identifier for the license, which should be drawn from the SPDX license vocabulary, or a similar well-known license vocabulary.				
Diagram	<p>The official identifier for the license, which should be drawn from the SPDX license vocabulary, or a similar well-known license vocabulary.</p> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p>				
Type	NonEmptyStringType				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ]*[ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ]*[ \s \S ]*
minLength	1				
pattern	[ \s ]*[ \S ]*[ \s \S ]*				
Source	<pre>&lt;xs:element name="identifier" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The official identifier for the license, which should be drawn from the SPDX license vocabulary, or a similar well-known license vocabulary.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>				

## Element distribution

Namespace	No namespace						
Annotations	This element provides information on how the resource is distributed. When used at the resource level, this element can provide only general information, but elements for describing connections to online systems are provided.						
Diagram	<p>This element provides information on how the resource is distributed. When used at the resource level, this element can...</p> <p>This element contains information for accessing the resource online represented as a URL connection</p>						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> </table>	content:	complex				
content:	complex						
Used by	Elements additionalMetadata/metadata/gbif/physical, dataset						
Model	online						
Children	online						
Instance	<pre>&lt;distribution scope=""&gt;   &lt;online&gt;{1,1}&lt;/online&gt; &lt;/distribution&gt;</pre>						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>scope</td> <td>xs:string</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	scope	xs:string	optional
QName	Type	Use					
scope	xs:string	optional					
Source	<pre>&lt;xs:element name="distribution"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element provides information on how the resource is distributed. When used at the resource level, this element can provide only general information, but elements for describing connections to online systems are provided.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>						

```

<xs:sequence>
  <xs:element ref="online" />
</xs:sequence>
<xs:attribute name="scope" type="xs:string" />
</xs:complexType>
</xs:elements>

```

## Element online

Namespace	No namespace
Annotations	This element contains information for accessing the resource online represented as a URL connection
Diagram	
Properties	content: complex
Used by	Element distribution
Model	url
Children	url
Instance	<online>   <url function="">{1,1}</url> </online>
Source	<xs:element name="online">   <xs:annotation>     <xs:documentation>This element contains information for accessing the resource online represented as a URL connection</xs:documentation>   </xs:annotation>   <xs:complexType>     <xs:sequence>       <xs:element ref="url" />     </xs:sequence>   </xs:complexType> </xs:element>

## Element url

Namespace	No namespace						
Annotations	The URL of the resource that is available online.						
Diagram							
Type	extension of xs:anyURI						
Properties	content: complex						
Used by	Element online						
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>function</td> <td>xs:string</td> <td>required</td> </tr> </tbody> </table>	QName	Type	Use	function	xs:string	required
QName	Type	Use					
function	xs:string	required					
Source	<xs:element name="url">   <xs:annotation>     <xs:documentation>The URL of the resource that is available online.</xs:documentation>   </xs:annotation>   <xs:complexType>     <xs:simpleContent>       <xs:extension base="xs:anyURI">						

```

<xs:attribute name="function" use="required" type="xs:string"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>

```

## Element coverage

Namespace	No namespace
Annotations	Describes the extent of the coverage of the resource in terms of its spatial extent, temporal extent, and taxonomic extent
Diagram	<pre> classDiagram     coverage &lt; -- geographicCoverage     coverage &lt; -- temporalCoverage     coverage &lt; -- taxonomicCoverage     coverage "1..oo" --&gt; coverage     </pre> <p>The coverage element is a container for spatial information about a resource. It allows a bounding box for the overall coverage (in lat long), and also allows description of arbitrary polygons with exclusions.</p>
Properties	content: complex
Used by	Element dataset
Model	geographicCoverage   temporalCoverage   taxonomicCoverage
Children	geographicCoverage, taxonomicCoverage, temporalCoverage
Instance	<pre> &lt;coverage&gt;   &lt;geographicCoverage&gt;{1,1}&lt;/geographicCoverage&gt;   &lt;temporalCoverage&gt;{1,1}&lt;/temporalCoverage&gt;   &lt;taxonomicCoverage&gt;{1,1}&lt;/taxonomicCoverage&gt; &lt;/coverage&gt; </pre>
Source	<pre> &lt;xs:element name="coverage"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Describes the extent of the coverage of the resource in terms of its spatial extent, temporal extent, and taxonomic extent&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:choice maxOccurs="unbounded"&gt;       &lt;xs:element ref="geographicCoverage" maxOccurs="1"/&gt;       &lt;xs:element ref="temporalCoverage" maxOccurs="1"/&gt;       &lt;xs:element ref="taxonomicCoverage" maxOccurs="1"/&gt;     &lt;/xs:choice&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

## Element geographicCoverage

Namespace	No namespace
Annotations	A container for spatial information about a resource; allows a bounding box for the overall coverage (in lat long), and also allows description of arbitrary polygons with exclusions.
Diagram	<pre> classDiagram     geographicCoverage &lt; -- geographicDescription     geographicCoverage &lt; -- boundingCoordinates     geographicCoverage "1..oo" --&gt; geographicCoverage     </pre> <p>The geographicCoverage element is a container for spatial information about a resource; allows a bounding box for the overall coverage (in lat long), and also allows description of arbitrary polygons with exclusions.</p>
Properties	content: complex
Used by	Element coverage
Model	geographicDescription , boundingCoordinates
Children	boundingCoordinates, geographicDescription
Instance	<pre> &lt;geographicCoverage&gt;   &lt;geographicDescription&gt;{1,1}&lt;/geographicDescription&gt; &lt;/geographicCoverage&gt; </pre>

	<pre> &lt;boundingCoordinates&gt;{1,1}&lt;/boundingCoordinates&gt; &lt;/geographicCoverage&gt; </pre>
Source	<pre> &lt;xs:element name="geographicCoverage"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A container for spatial information about a resource; allows a bounding box for the overall coverage (in lat long), and also allows description of arbitrary polygons with exclusions.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="geographicDescription" type="NonEmptyStringType" minOccurs="1"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A short text description of a dataset's geographic areal domain. A text description is especially important to provide a geographic setting when the extent of the dataset cannot be well described by the "boundingCoordinates".&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="boundingCoordinates"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Bounding Coordinates are the four margins (N, S, E, W) of a bounding box, or when considered in lat-lon pairs, the corners of the box.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="westBoundingCoordinate" type="xs:decimal"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The westBoundingCoordinate field defines the longitude of the western-most point of the bounding box that is being described.&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="eastBoundingCoordinate" type="xs:decimal"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The eastBoundingCoordinate field defines the longitude of the eastern-most point of the bounding box that is being described.&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="northBoundingCoordinate" type="xs:decimal"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The northBoundingCoordinate field defines the latitude of the northern-most point of the bounding box that is being described.&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="southBoundingCoordinate" type="xs:decimal"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The southBoundingCoordinate field defines the latitude of the southern-most point of the bounding box that is being described.&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

## Element geographicCoverage / geographicDescription

Namespace	No namespace				
Annotations	A short text description of a dataset's geographic areal domain. A text description is especially important to provide a geographic setting when the extent of the dataset cannot be well described by the "boundingCoordinates".				
Diagram	<pre> classDiagram     class geographicDescription {         &lt;&lt;NonEmptyStringType&gt;&gt;     }     geographicDescription &lt; -- NonEmptyStringType     Note over geographicDescription: A short text description of a dataset's geographic areal domain. A text description is especially important to provide...     Note over NonEmptyStringType: Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...   </pre>				
Type	NonEmptyStringType				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>1</td> </tr> </table>	content:	simple	minOccurs:	1
content:	simple				
minOccurs:	1				
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s]*[\S][\s\S]*</td> </tr> </table>	minLength	1	pattern	[ \s]*[\S][\s\S]*
minLength	1				
pattern	[ \s]*[\S][\s\S]*				
Source	<pre> &lt;xs:element name="geographicDescription" type="NonEmptyStringType" minOccurs="1"&gt;   &lt;xs:annotation&gt; </pre>				

```

<xs:documentation>A short text description of a dataset's geographic areal domain. A text description is especially important to provide a geographic setting when the extent of the dataset cannot be well described by the "boundingCoordinates".</xs:documentation>
</xs:annotation>
</xs:element>

```

## Element geographicCoverage / boundingCoordinates

Namespace	No namespace
Annotations	Bounding Coordinates are the four margins (N, S, E, W) of a bounding box, or when considered in lat-lon pairs, the corners of the box.
Diagram	<pre> graph TD     BC[boundingCoordinates] --&gt; WB[westBoundingCoordinate]     BC --&gt; EB[eastBoundingCoordinate]     BC --&gt; NB[northBoundingCoordinate]     BC --&gt; SB[southBoundingCoordinate]     WB --&gt; WB_desc["The westBoundingCoordinate field defines the longitude of the western-most point of the bounding box that is being..."]     EB --&gt; EB_desc["The eastBoundingCoordinate field defines the longitude of the eastern-most point of the bounding box that is being..."]     NB --&gt; NB_desc["The northBoundingCoordinate field defines the latitude of the northern-most point of the bounding box that is being..."]     SB --&gt; SB_desc["The southBoundingCoordinate field defines the latitude of the southern-most point of the bounding box that is being..."] </pre>
Properties	content: complex
Model	westBoundingCoordinate , eastBoundingCoordinate , northBoundingCoordinate , southBoundingCoordinate
Children	eastBoundingCoordinate, northBoundingCoordinate, southBoundingCoordinate, westBoundingCoordinate
Instance	<pre> &lt;boundingCoordinates&gt;     &lt;westBoundingCoordinate&gt;{1,1}&lt;/westBoundingCoordinate&gt;     &lt;eastBoundingCoordinate&gt;{1,1}&lt;/eastBoundingCoordinate&gt;     &lt;northBoundingCoordinate&gt;{1,1}&lt;/northBoundingCoordinate&gt;     &lt;southBoundingCoordinate&gt;{1,1}&lt;/southBoundingCoordinate&gt; &lt;/boundingCoordinates&gt; </pre>
Source	<pre> &lt;xs:element name="boundingCoordinates"&gt;     &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Bounding Coordinates are the four margins (N, S, E, W) of a bounding box, or when considered in lat-lon pairs, the corners of the box.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;     &lt;xs:complexType&gt;         &lt;xs:sequence&gt;             &lt;xs:element name="westBoundingCoordinate" type="xs:decimal"&gt;                 &lt;xs:annotation&gt;                     &lt;xs:documentation&gt;The westBoundingCoordinate field defines the longitude of the western-most point of the bounding box that is being described.&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="eastBoundingCoordinate" type="xs:decimal"&gt;                 &lt;xs:annotation&gt;                     &lt;xs:documentation&gt;The eastBoundingCoordinate field defines the longitude of the eastern-most point of the bounding box that is being described.&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="northBoundingCoordinate" type="xs:decimal"&gt;                 &lt;xs:annotation&gt;                     &lt;xs:documentation&gt;The northBoundingCoordinate field defines the latitude of the northern-most point of the bounding box that is being described.&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="southBoundingCoordinate" type="xs:decimal"&gt;                 &lt;xs:annotation&gt;                     &lt;xs:documentation&gt;The southBoundingCoordinate field defines the latitude of the southern-most point of the bounding box that is being described.&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;             &lt;/xs:element&gt;         &lt;/xs:sequence&gt;     &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

**Element geographicCoverage / boundingCoordinates / westBoundingCoordinate**

Namespace	No namespace
Annotations	The westBoundingCoordinate field defines the longitude of the western-most point of the bounding box that is being described.
Diagram	<pre> classDiagram     class westBoundingCoordinate {         &lt;&lt;xs:decimal&gt;&gt;     }     class xs_decimal {         &lt;&lt;Built-in primitive type. The decimal datatype represents arbitrary precision decimal numbers.&gt;&gt;     }     westBoundingCoordinate o--&gt; xs_decimal   </pre> <p>The westBoundingCoordinate field defines the longitude of the western-most point of the bounding box that is being described...</p>
Type	xs:decimal
Properties	content: simple
Source	<pre> &lt;xs:element name="westBoundingCoordinate" type="xs:decimal"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The westBoundingCoordinate field defines the longitude of the western-most point of the bounding box that is being described.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;   </pre>

**Element geographicCoverage / boundingCoordinates / eastBoundingCoordinate**

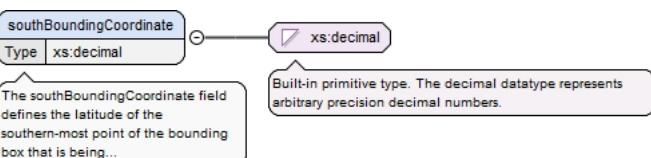
Namespace	No namespace
Annotations	The eastBoundingCoordinate field defines the longitude of the eastern-most point of the bounding box that is being described.
Diagram	<pre> classDiagram     class eastBoundingCoordinate {         &lt;&lt;xs:decimal&gt;&gt;     }     class xs_decimal {         &lt;&lt;Built-in primitive type. The decimal datatype represents arbitrary precision decimal numbers.&gt;&gt;     }     eastBoundingCoordinate o--&gt; xs_decimal   </pre> <p>The eastBoundingCoordinate field defines the longitude of the eastern-most point of the bounding box that is being described...</p>
Type	xs:decimal
Properties	content: simple
Source	<pre> &lt;xs:element name="eastBoundingCoordinate" type="xs:decimal"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The eastBoundingCoordinate field defines the longitude of the eastern-most point of the bounding box that is being described.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;   </pre>

**Element geographicCoverage / boundingCoordinates / northBoundingCoordinate**

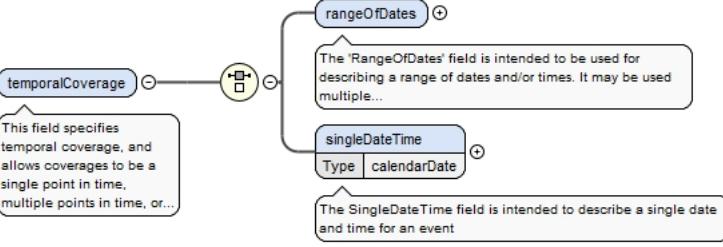
Namespace	No namespace
Annotations	The northBoundingCoordinate field defines the latitude of the northern-most point of the bounding box that is being described.
Diagram	<pre> classDiagram     class northBoundingCoordinate {         &lt;&lt;xs:decimal&gt;&gt;     }     class xs_decimal {         &lt;&lt;Built-in primitive type. The decimal datatype represents arbitrary precision decimal numbers.&gt;&gt;     }     northBoundingCoordinate o--&gt; xs_decimal   </pre> <p>The northBoundingCoordinate field defines the latitude of the northern-most point of the bounding box that is being described...</p>
Type	xs:decimal
Properties	content: simple
Source	<pre> &lt;xs:element name="northBoundingCoordinate" type="xs:decimal"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The northBoundingCoordinate field defines the latitude of the northern-most point of the bounding box that is being described.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;   </pre>

**Element geographicCoverage / boundingCoordinates / southBoundingCoordinate**

Namespace	No namespace
Annotations	The southBoundingCoordinate field defines the latitude of the southern-most point of the bounding box that is being described.

Diagram	
Type	xs:decimal
Properties	content: simple
Source	<pre>&lt;xs:element name="southBoundingCoordinate" type="xs:decimal"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The southBoundingCoordinate field defines the latitude of the southern-most point of the bounding box that is being described.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element temporalCoverage

Namespace	No namespace
Annotations	This field specifies temporal coverage, and allows coverages to be a single point in time, multiple points in time, or a range of dates.
Diagram	
Properties	content: complex
Used by	Element coverage
Model	rangeOfDates   singleDateTime
Children	rangeOfDates, singleDateTime
Instance	<pre>&lt;temporalCoverage&gt;   &lt;rangeOfDates&gt;{1,1}&lt;/rangeOfDates&gt;   &lt;singleDateTime&gt;{1,1}&lt;/singleDateTime&gt; &lt;/temporalCoverage&gt;</pre>
Source	<pre>&lt;xs:element name="temporalCoverage"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This field specifies temporal coverage, and allows coverages to be a single point in time, multiple points in time, or a range of dates.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:choice&gt;       &lt;xs:element name="rangeOfDates"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The 'RangeOfDates' field is intended to be used for describing a range of dates and/or times. It may be used multiple times to document multiple date ranges. It allows for two 'singleDateTime' fields, the first to be used as the beginning dateTime, and the second to be used as the ending dateTime of the range.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="beginDate" type="calendarDate"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;A single time stamp signifying the beginning of some time period&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="endDate" type="calendarDate"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;A single time stamp signifying the end of some time period&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;       &lt;xs:element name="singleDateTime"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A single time stamp signifying a specific point in time&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:choice&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

```

<xs:annotation>
  <xs:documentation>The SingleDateTime field is intended to describe a single date and time
for an event</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>

```

## Element temporalCoverage / rangeOfDates

Namespace	No namespace
Annotations	The 'RangeOfDates' field is intended to be used for describing a range of dates and/or times. It may be used multiple times to document multiple date ranges. It allows for two 'singleDateTime' fields, the first to be used as the beginning dateTime, and the second to be used as the ending dateTime of the range.
Diagram	<pre> classDiagram     class rangeOfDates {         beginDate         endDate     }     beginDate &lt; -- calendarDate     endDate &lt; -- calendarDate </pre> <p>The 'RangeOfDates' field is intended to be used for describing a range of dates and/or times. It may be used multiple times to document multiple date ranges. It allows for two 'singleDateTime' fields, the first to be used as the beginning dateTime, and the second to be used as the ending dateTime of the range.</p>
Properties	content: complex
Model	beginDate , endDate
Children	beginDate, endDate
Instance	<rangeOfDates>   <beginDate>{1,1}</beginDate>   <endDate>{1,1}</endDate> </rangeOfDates>
Source	<xs:element name="rangeOfDates">   <xs:annotation>     <xs:documentation>The 'RangeOfDates' field is intended to be used for describing a range of dates and/or times. It may be used multiple times to document multiple date ranges. It allows for two 'singleDateTime' fields, the first to be used as the beginning dateTime, and the second to be used as the ending dateTime of the range.</xs:documentation>   </xs:annotation>   <xs:complexType>     <xs:sequence>       <xs:element name="beginDate" type="calendarDate">         <xs:annotation>           <xs:documentation>A single time stamp signifying the beginning of some time period</xs:documentation>         </xs:annotation>       </xs:element>       <xs:element name="endDate" type="calendarDate">         <xs:annotation>           <xs:documentation>A single time stamp signifying the end of some time period</xs:documentation>         </xs:annotation>       </xs:element>     </xs:sequence>   </xs:complexType> </xs:element>

## Element temporalCoverage / rangeOfDates / beginDate

Namespace	No namespace
Annotations	A single time stamp signifying the beginning of some time period
Diagram	<pre> classDiagram     class beginDate {         calendarDate     }     calendarDate &lt; -- yearMonth     calendarDate &lt; -- yearDay </pre> <p>A single time stamp signifying the beginning of some time period</p> <p>The calendar date field is used to express a date, giving the year, month, and day. The format should be one that...</p>
Type	calendarDate

Properties	content:	complex
Model	calendarDate	
Children	calendarDate	
Instance	<beginDate> <calendarDate>{1,1}</calendarDate> </beginDate>	
Source	<xs:element name="beginDate" type="calendarDate"> <xs:annotation> <xs:documentation>A single time stamp signifying the beginning of some time period</xs:documentation> </xs:annotation> </xs:element>	

## Element calendarDate

Namespace	No namespace	
Annotations	The calendar date field is used to express a date, giving the year, month, and day. The format should be one that complies with the International Standards Organization's standard 8601. The recommended format for EML is YYYY-MM-DD, where Y is the four digit year, M is the two digit month code (01 - 12, where January = 01), and D is the two digit day of the month (01 - 31). This field can also be used to enter just the year portion of a date.	
Diagram	<pre> classDiagram     class calendarDate {         Type     }     class yearDate {         Type     }     calendarDate "1" --&gt; "2" yearDate     </pre>	
Type	yearDate	
Properties	content:	simple
Used by	Complex Type	calendarDate
Source	<xs:element name="calendarDate" type="yearDate"> <xs:annotation> <xs:documentation>The calendar date field is used to express a date, giving the year, month, and day. The format should be one that complies with the International Standards Organization's standard 8601. The recommended format for EML is YYYY-MM-DD, where Y is the four digit year, M is the two digit month code (01 - 12, where January = 01), and D is the two digit day of the month (01 - 31). This field can also be used to enter just the year portion of a date.</xs:documentation> </xs:annotation> </xs:element>	

## Element temporalCoverage / rangeOfDates / endDate

Namespace	No namespace	
Annotations	A single time stamp signifying the end of some time period	
Diagram	<pre> classDiagram     class endDate {         Type     }     class calendarDate {         Type     }     class yearDate {         Type     }     endDate "1" --&gt; "2" calendarDate     endDate "1" --&gt; "2" yearDate     </pre>	
Type	calendarDate	
Properties	content:	complex
Model	calendarDate	
Children	calendarDate	
Instance	<endDate> <calendarDate>{1,1}</calendarDate> </endDate>	
Source	<xs:element name="endDate" type="calendarDate"> <xs:annotation> <xs:documentation>A single time stamp signifying the end of some time period</xs:documentation> </xs:annotation> </xs:element>	

## Element temporalCoverage / singleDateTime

Namespace	No namespace
Annotations	The SingleDateTime field is intended to describe a single date and time for an event
Diagram	<pre> classDiagram     class singleDateTime {         &lt;&lt;Type   calendarDate&gt;&gt;     }     class calendarDate {         &lt;&lt;Type   yearDate&gt;&gt;     }     singleDateTime --&gt; calendarDate     calendarDate --&gt; yearDate   </pre> <p>The SingleDateTime field is intended to describe a single date and time for an event.</p> <p>The calendar date field is used to express a date, giving the year, month, and day. The format should be one that...</p>
Type	calendarDate
Properties	content: complex
Model	calendarDate
Children	calendarDate
Instance	<pre> &lt;singleDateTime&gt;   &lt;calendarDate&gt;{1,1}&lt;/calendarDate&gt; &lt;/singleDateTime&gt;   </pre>
Source	<pre> &lt;xs:element name="singleDateTime" type="calendarDate"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The SingleDateTime field is intended to describe a single date and time for an event&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;   </pre>

## Element taxonomicCoverage

Namespace	No namespace
Annotations	Taxonomic Coverage is a container for taxonomic information about a resource. It includes a list of species names (or higher level ranks) from one or more classification systems.
Diagram	<pre> classDiagram     class taxonomicCoverage {         &lt;&lt;1..&gt;&gt;     }     class generalTaxonomicCoverage {         &lt;&lt;Type   NonEmptyStringType&gt;&gt;     }     class taxonomicClassification {         &lt;&lt;1..&gt;&gt;     }     taxonomicCoverage --&gt; generalTaxonomicCoverage     generalTaxonomicCoverage --&gt; taxonomicClassification   </pre> <p>Taxonomic Coverage is a container for taxonomic information about a resource. It includes a list of species names (or....</p> <p>A general description of the range of taxa addressed in the data set or collection</p> <p>Information about the range of taxa addressed in the dataset or collection</p>
Properties	content: complex
Used by	Element coverage
Model	generalTaxonomicCoverage{0,1} , taxonomicClassification+
Children	generalTaxonomicCoverage, taxonomicClassification
Instance	<pre> &lt;taxonomicCoverage&gt;   &lt;generalTaxonomicCoverage&gt;{0,1}&lt;/generalTaxonomicCoverage&gt;   &lt;taxonomicClassification&gt;{1,unbounded}&lt;/taxonomicClassification&gt; &lt;/taxonomicCoverage&gt;   </pre>
Source	<pre> &lt;xs:element name="taxonomicCoverage"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Taxonomic Coverage is a container for taxonomic information about a resource. It includes a list of species names (or higher level ranks) from one or more classification systems.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="generalTaxonomicCoverage" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A general description of the range of taxa addressed in the data set or collection&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="taxonomicClassification" maxOccurs="unbounded"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Information about the range of taxa addressed in the dataset or collection&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="taxonRankName" type="NonEmptyStringType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The name of the taxonomic rank for which the Taxon rank value is provided, e.g., Phylum, Class, Genus, Species</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="taxonRankValue" type="NonEmptyStringType">
      <xs:annotation>
        <xs:documentation>The name representing the taxonomic rank of the taxon being described</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="commonName" type="NonEmptyStringType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>Applicable common names; these common names may be general descriptions of a group of organisms if appropriate, e.g., invertebrates, waterfowl</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:elements>

```

## Element taxonomicCoverage / generalTaxonomicCoverage

Namespace	No namespace				
Annotations	A general description of the range of taxa addressed in the data set or collection				
Diagram	<p>generalTaxonomicCoverage</p> <p>Type   NonEmptyStringType</p> <p>A general description of the range of taxa addressed in the data set or collection</p>				
Type	NonEmptyStringType				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[\\s]*[\\S][\\s\\S]*</td> </tr> </table>	minLength	1	pattern	[\\s]*[\\S][\\s\\S]*
minLength	1				
pattern	[\\s]*[\\S][\\s\\S]*				
Source	<pre> &lt;xs:element name="generalTaxonomicCoverage" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A general description of the range of taxa addressed in the data set or collection&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>				

## Element taxonomicCoverage / taxonomicClassification

Namespace	No namespace
Annotations	Information about the range of taxa addressed in the dataset or collection
Diagram	<p>taxonomicClassification</p> <p>taxonRankName Type   NonEmptyStringType</p> <p>The name of the taxonomic rank for which the Taxon rank value is provided, e.g., Phylum, Class, Genus, Species</p> <p>taxonRankValue Type   NonEmptyStringType</p> <p>The name representing the taxonomic rank of the taxon being described</p> <p>commonName Type   NonEmptyStringType</p> <p>Applicable common names; these common names may be general descriptions of a group of organisms if appropriate, e.g.....</p>
Properties	content: complex

	maxOccurs: unbounded
Model	taxonRankName{0,1} , taxonRankValue , commonName{0,1}
Children	commonName, taxonRankName, taxonRankValue
Instance	<pre>&lt;taxonomicClassification&gt;   &lt;taxonRankName&gt;{0,1}&lt;/taxonRankName&gt;   &lt;taxonRankValue&gt;{1,1}&lt;/taxonRankValue&gt;   &lt;commonName&gt;{0,1}&lt;/commonName&gt; &lt;/taxonomicClassification&gt;</pre>
Source	<pre>&lt;xss:element name="taxonomicClassification" maxOccurs="unbounded"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;Information about the range of taxa addressed in the dataset or collection&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element name="taxonRankName" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;The name of the taxonomic rank for which the Taxon rank value is provided, e.g., Phylum, Class, Genus, Species&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;       &lt;xss:element name="taxonRankValue" type="NonEmptyStringType"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;The name representing the taxonomic rank of the taxon being described&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;       &lt;xss:element name="commonName" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;Applicable common names; these common names may be general descriptions of a group of organisms if appropriate, e.g., invertebrates, waterfowl&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;     &lt;/xss:sequence&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>

### Element taxonomicCoverage / taxonomicClassification / taxonRankName

Namespace	No namespace				
Annotations	The name of the taxonomic rank for which the Taxon rank value is provided, e.g., Phylum, Class, Genus, Species				
Diagram					
Type	NonEmptyStringType				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[\\s]*[\\S][\\s\\S]*</td> </tr> </table>	minLength	1	pattern	[\\s]*[\\S][\\s\\S]*
minLength	1				
pattern	[\\s]*[\\S][\\s\\S]*				
Source	<pre>&lt;xss:element name="taxonRankName" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;The name of the taxonomic rank for which the Taxon rank value is provided, e.g., Phylum, Class, Genus, Species&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>				

### Element taxonomicCoverage / taxonomicClassification / taxonRankValue

Namespace	No namespace
Annotations	The name representing the taxonomic rank of the taxon being described
Diagram	

Type	NonEmptyStringType
Properties	content: simple
Facets	minLength 1 pattern [\s]*[\S][\s\S]*
Source	<pre>&lt;xs:element name="taxonRankValue" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The name representing the taxonomic rank of the taxon being described&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element taxonomicCoverage / taxonomicClassification / commonName

Namespace	No namespace
Annotations	Applicable common names; these common names may be general descriptions of a group of organisms if appropriate, e.g., invertebrates, waterfowl
Diagram	<p>commonName</p> <p>Type   NonEmptyStringType</p> <p>Applicable common names; these common names may be general descriptions of a group of organisms if appropriate, e.g....</p> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p>
Type	NonEmptyStringType
Properties	content: simple minOccurs: 0
Facets	minLength 1 pattern [\s]*[\S][\s\S]*
Source	<pre>&lt;xs:element name="commonName" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Applicable common names; these common names may be general descriptions of a group of organisms if appropriate, e.g., invertebrates, waterfowl&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element purpose

Namespace	No namespace
Annotations	A synopsis of the purpose of this dataset. It may include one or more paragraphs, including a summary of key findings if appropriate.
Diagram	<p>purpose</p> <p>Type   TextType</p> <p>A synopsis of the purpose of this dataset. It may include one or more paragraphs, including a summary of key findings...</p> <p>TextType</p> <p>Mixed   true</p> <p>@ Attributes</p> <p>@ xml:lang</p> <p>&lt;div&gt; &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt; &lt;p&gt; denotes an attribute whose value is a language code for the natural...</p> <p>section</p> <p>0..oo</p> <p>para</p> <p>The "section" element allows for grouping related paragraphs of text together, with an optional title. This markup is a... The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain...</p> <p>text</p> <p>The "text" element allows for both formatted and unformatted text blocks to be included in EML. It can contain a number...</p>
Type	TextType
Properties	content: complex

	mixed:	true						
Used by	Element	dataset						
Model	section*   para*							
Children	para, section							
Instance	<pre>&lt;purpose xml:lang=""&gt;   &lt;section xml:lang=""&gt;{0,unbounded}&lt;/section&gt;   &lt;para xml:lang=""&gt;{0,unbounded}&lt;/para&gt; &lt;/purpose&gt;</pre>							
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>xml:lang</td> <td>union of(xs:language, restriction of xs:string)</td> <td>optional</td> </tr> </tbody> </table> <p> <code>&lt;div&gt;</code>  <code>  &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</code>  <code>  &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</code>  <code>&lt;/div&gt;</code>  <code>&lt;div&gt;</code>  <code>  &lt;h4&gt;Notes&lt;/h4&gt;</code>  <code>  &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</code>  <code>  &lt;p&gt;See BCP 47 at</code>  <code>    &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at</code>  <code>    &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for</code>  <code>    further information.&lt;/p&gt;</code>  <code>  &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</code>  <code>&lt;/div&gt;</code> </p>	QName	Type	Use	xml:lang	union of(xs:language, restriction of xs:string)	optional	
QName	Type	Use						
xml:lang	union of(xs:language, restriction of xs:string)	optional						
Source	<pre>&lt;xss:element name="purpose" type="TextType"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;A synopsis of the purpose of this dataset. It may include one or more paragraphs, including a summary of key findings if appropriate.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>							

## Element introduction

Namespace	No namespace
Annotations	One to many paragraphs that provide background and context for the dataset with appropriate figures and references. This is similar to the introduction for a journal article, and would include, for example, project objectives, hypotheses being addressed, what is known about the pattern or process under study, how the data have been used to date (including references), and how they could be used in the future.
Diagram	<pre> classDiagram     class introduction {         Type TextType     }     class TextType {         Mixed true     }     class Attributes {         @ xml:lang xs:string     }     class section {         Type SectionType     }     class para {         Type ParagraphType     }     class text      introduction "1..oo" --&gt; TextType     TextType "*" --&gt; Attributes     TextType "*" --&gt; section     TextType "*" --&gt; para     TextType "*" --&gt; text   </pre>
Type	TextType

Properties	content: mixed:	complex true												
Used by	Element	dataset												
Model	section*   para*													
Children	para, section													
Instance	<pre>&lt;introduction xml:lang=""&gt;   &lt;section xml:lang=""&gt;{0,unbounded}&lt;/section&gt;   &lt;para xml:lang=""&gt;{0,unbounded}&lt;/para&gt; &lt;/introduction&gt;</pre>													
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th><th></th></tr> </thead> <tbody> <tr> <td>xml:lang</td><td>union of(xs:language, restriction of xs:string)</td><td>optional</td><td></td></tr> <tr> <td></td><td> <pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt;</pre> </td><td></td><td></td></tr> </tbody> </table>	QName	Type	Use		xml:lang	union of(xs:language, restriction of xs:string)	optional			<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt;</pre>			
QName	Type	Use												
xml:lang	union of(xs:language, restriction of xs:string)	optional												
	<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt;</pre>													
Source	<pre>&lt;xs:element name="introduction" type="TextType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;One to many paragraphs that provide background and context for the dataset with appropriate figures and references. This is similar to the introduction for a journal article, and would include, for example, project objectives, hypotheses being addressed, what is known about the pattern or process under study, how the data have been used to date (including references), and how they could be used in the future.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>													

## Element gettingStarted

Namespace	No namespace
Annotations	One or more paragraphs describing the dataset's overall interpretation, content and structure. For example, the number and names of data files, the types of measurements that they contain, how those data files fit together in an overall design, and how they relate to the data collection methods, experimental design, and sampling design described in other EML sections. One might describe any specialized software that is available and/or may be necessary for analyzing or interpreting the data, and possibly include a high-level description of data formats if they are unusual.

Diagram	<pre> classDiagram     class TextType {         &lt;&lt;Mixed   true&gt;&gt;         @xml:lang         gettingStarted "1..oo"         section "0..oo"         para "0..oo"         text "0..oo"     }   </pre> <p>The diagram shows the <code>TextType</code> element as a mixed type (<code>Mixed   true</code>). It has an attribute <code>@xml:lang</code>. It contains three children: <code>gettingStarted</code> (multiplicity 1..oo), <code>section</code> (multiplicity 0..oo), <code>para</code> (multiplicity 0..oo), and <code>text</code> (multiplicity 0..oo). A callout box for <code>gettingStarted</code> describes it as one or more paragraphs describing the dataset's overall interpretation, content and structure. A callout box for <code>section</code> describes it as allowing for grouping related paragraphs of text together. A callout box for <code>para</code> describes it as allowing for both formatted and unformatted text blocks. A callout box for <code>text</code> describes it as allowing for both formatted and unformatted text blocks.</p>												
Type	TextType												
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">mixed:</td><td style="padding: 2px;">true</td></tr> </table>	content:	complex	mixed:	true								
content:	complex												
mixed:	true												
Used by	Element dataset												
Model	section*   para*												
Children	para, section												
Instance	<pre> &lt;gettingStarted xml:lang=""&gt;   &lt;section xml:lang=""&gt;{0,unbounded}&lt;/section&gt;   &lt;para xml:lang=""&gt;{0,unbounded}&lt;/para&gt; &lt;/gettingStarted&gt;   </pre>												
Attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">QName</th><th style="width: 30%;">Type</th><th style="width: 30%;">Use</th><th style="width: 10%;"></th></tr> </thead> <tbody> <tr> <td><code>xml:lang</code></td><td>union of(xs:language, restriction of xs:string)</td><td>optional</td><td></td></tr> <tr> <td></td><td></td><td> <div style="font-family: monospace; padding: 10px;"> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </div> </td><td></td></tr> </tbody> </table>	QName	Type	Use		<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional				<div style="font-family: monospace; padding: 10px;"> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </div>	
QName	Type	Use											
<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional											
		<div style="font-family: monospace; padding: 10px;"> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </div>											
Source	<pre> &lt;xs:element name="gettingStarted" type="TextType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;One or more paragraphs describing the dataset's overall interpretation, content and structure. For example, the number and names of data files, the types of measurements that they contain, how those data files fit together in an overall design, and how they relate to the data collection methods, experimental design, and sampling design described in other EML sections. One might describe any specialized software that is available and/or may be necessary for analyzing or interpreting the data, and possibly include a high-level description of data formats if they are unusual.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;   </pre>												

## Element acknowledgements

Namespace	No namespace									
Annotations	<p>One or more sentences that acknowledge funders and other key contributors to the study (excluding the dataset authors listed in the creator field). Note that funding awards are also listed by award number in the award section, which provides a structured list of funders, award numbers, and award URIs for the dataset.</p>									
Diagram	<pre> classDiagram     class acknowledgements {         Type TextType         One or more sentences that acknowledge funders and other key contributors to the study (excluding the dataset authors...)     }     class TextType {         Mixed true     }     class Attributes {         @xml:lang         &lt;div&gt; &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt; &lt;p&gt;denotes an attribute whose value is a language code for the natural...     }     class section {         Type SectionType         0..&gt; section         The "section" element allows for grouping related paragraphs of text together, with an optional title. This markup is a...     }     class para {         Type ParagraphType         0..&gt; para         The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain...     }     class text {         The "text" element allows for both formatted and unformatted text blocks to be included in EML. It can contain a number...     }     acknowledgements --&gt; TextType     TextType --&gt; Attributes     Attributes --&gt; section     section --&gt; para     para --&gt; text   </pre>									
Type	TextType									
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>mixed:</td> <td>true</td> </tr> </table>	content:	complex	mixed:	true					
content:	complex									
mixed:	true									
Used by	Element dataset									
Model	section*   para*									
Children	para, section									
Instance	<pre> &lt;acknowledgements xml:lang=""&gt;   &lt;section xml:lang=""&gt;{0,unbounded}&lt;/section&gt;   &lt;para xml:lang=""&gt;{0,unbounded}&lt;/para&gt; &lt;/acknowledgements&gt;   </pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>xml:lang</td> <td>union of(xs:language, restriction of xs:string)</td> <td>optional</td> </tr> <tr> <td></td> <td> <div> <p>&lt;div&gt;</p> <p>&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p>&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p>&lt;h4&gt;Notes&lt;/h4&gt;</p> <p>&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p>&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 2em;">&lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at</p> <p style="margin-left: 2em;">&lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;</p> <p>&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> </div> </td> <td></td> </tr> </tbody> </table>	QName	Type	Use	xml:lang	union of(xs:language, restriction of xs:string)	optional		<div> <p>&lt;div&gt;</p> <p>&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p>&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p>&lt;h4&gt;Notes&lt;/h4&gt;</p> <p>&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p>&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 2em;">&lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at</p> <p style="margin-left: 2em;">&lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;</p> <p>&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> </div>	
QName	Type	Use								
xml:lang	union of(xs:language, restriction of xs:string)	optional								
	<div> <p>&lt;div&gt;</p> <p>&lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;</p> <p>&lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> <p>&lt;div&gt;</p> <p>&lt;h4&gt;Notes&lt;/h4&gt;</p> <p>&lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;</p> <p>&lt;p&gt;See BCP 47 at</p> <p style="margin-left: 2em;">&lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at</p> <p style="margin-left: 2em;">&lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;</p> <p>&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p> <p>&lt;/div&gt;</p> </div>									
Source	<pre> &lt;xs:element name="acknowledgements" type="TextType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;One or more sentences that acknowledge funders and other key contributors to the study (excluding the dataset authors listed in the creator field). Note that funding awards are also listed by award number in the award section, which provides a structured list of funders, award numbers, and award URIs for the dataset.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;   </pre>									

```

    </xs:annotation>
</xs:element>

```

## Element dataset / maintenance

Namespace	No namespace				
Diagram	<pre> classDiagram     class maintenance {         description         maintenanceUpdateFrequency         changeHistory     }     description --&gt; "The field Description contains general textual descriptions."     maintenanceUpdateFrequency --&gt; "Type MaintUpFreqType"     changeHistory --&gt; "A description of changes made to the data since its release." </pre>				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	description , maintenanceUpdateFrequency , changeHistory*				
Children	changeHistory, description, maintenanceUpdateFrequency				
Instance	<pre> &lt;maintenance&gt;   &lt;description&gt;{1,1}&lt;/description&gt;   &lt;maintenanceUpdateFrequency&gt;{1,1}&lt;/maintenanceUpdateFrequency&gt;   &lt;changeHistory&gt;{0,unbounded}&lt;/changeHistory&gt; &lt;/maintenance&gt; </pre>				
Source	<pre> &lt;xs:element minOccurs="0" name="maintenance"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="description"/&gt;       &lt;xs:element name="maintenanceUpdateFrequency" type="MaintUpFreqType"/&gt;       &lt;xs:element name="changeHistory" minOccurs="0" maxOccurs="unbounded"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A description of changes made to the data since its release.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="changeScope" type="NonEmptyStringType"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The expression should unambiguously identify the entity(s) and attribute(s) that were changed.&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="oldValue" type="NonEmptyStringType"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The previous value or an expression that describes the previous value of the data.&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="changeDate" type="xs:date"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The date the changes were applied.&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element name="comment" type="NonEmptyStringType" minOccurs="0"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;Explanation or justification for the change made to the data.&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;             &lt;/xs:element&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>				

## Element description

Namespace	No namespace
Annotations	The field Description contains general textual descriptions.

Diagram	
Properties	content: complex mixed: true
Used by	Element dataset/maintenance Complex Type description
Model	para*
Children	para
Instance	<description> <para>{0,unbounded}</para> </description>
Source	<pre>&lt;xss:element name="description"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;The field Description contains general textual descriptions.&lt;/ xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:complexType mixed="true"&gt;     &lt;xss:sequence&gt;       &lt;xss:element ref="para" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;/xss:sequence&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>

**Element dataset / maintenance / maintenanceUpdateFrequency**

Namespace	No namespace
Diagram	
Type	MaintUpFreqType
Properties	content: simple
Facets	enumeration annually enumeration asNeeded enumeration biannually enumeration continually enumeration daily enumeration irregular enumeration monthly enumeration notPlanned enumeration weekly enumeration unknown enumeration unkown enumeration otherMaintenancePeriod
Source	<pre>&lt;xss:element name="maintenanceUpdateFrequency" type="MaintUpFreqType" /&gt;</pre>

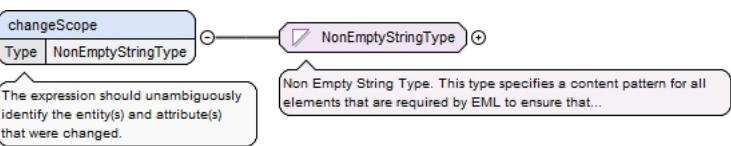
**Element dataset / maintenance / changeHistory**

Namespace	No namespace
Annotations	A description of changes made to the data since its release.

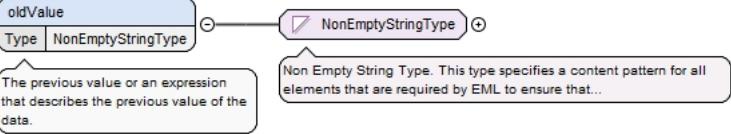
Diagram	<pre> classDiagram     class changeHistory {         changeScope &lt;--&gt; NonEmptyStringType         oldValue &lt;--&gt; NonEmptyStringType         changeDate &lt;--&gt; xs:date         comment &lt;--&gt; NonEmptyStringType     }     class NonEmptyStringType     class xs:date   </pre>						
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> <tr> <td style="padding: 2px;">maxOccurs:</td><td style="padding: 2px;">unbounded</td></tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	changeScope , oldValue , changeDate , comment{0,1}						
Children	changeDate, changeScope, comment, oldValue						
Instance	<pre> &lt;changeHistory&gt;   &lt;changeScope&gt;{1,1}&lt;/changeScope&gt;   &lt;oldValue&gt;{1,1}&lt;/oldValue&gt;   &lt;changeDate&gt;{1,1}&lt;/changeDate&gt;   &lt;comment&gt;{0,1}&lt;/comment&gt; &lt;/changeHistory&gt;   </pre>						
Source	<pre> &lt;xs:element name="changeHistory" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A description of changes made to the data since its release.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="changeScope" type="NonEmptyStringType"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The expression should unambiguously identify the entity(s) and attribute(s) that were changed.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="oldValue" type="NonEmptyStringType"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The previous value or an expression that describes the previous value of the data.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="changeDate" type="xs:date"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The date the changes were applied.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="comment" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Explanation or justification for the change made to the data.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>						

### Element dataset / maintenance / changeHistory / changeScope

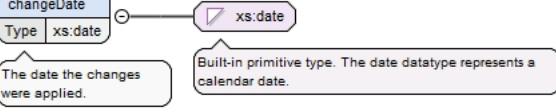
Namespace	No namespace
Annotations	The expression should unambiguously identify the entity(s) and attribute(s) that were changed.

Diagram	
Type	NonEmptyStringType
Properties	content: simple
Facets	minLength 1 pattern [\s]*[\S][\s\S]*
Source	<pre>&lt;xs:element name="changeScope" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The expression should unambiguously identify the entity(s) and attribute(s) that were changed.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element dataset / maintenance / changeHistory / oldValue

Namespace	No namespace
Annotations	The previous value or an expression that describes the previous value of the data.
Diagram	
Type	NonEmptyStringType
Properties	content: simple
Facets	minLength 1 pattern [\s]*[\S][\s\S]*
Source	<pre>&lt;xs:element name="oldValue" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The previous value or an expression that describes the previous value of the data.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element dataset / maintenance / changeHistory / changeDate

Namespace	No namespace
Annotations	The date the changes were applied.
Diagram	
Type	xs:date
Properties	content: simple
Source	<pre>&lt;xs:element name="changeDate" type="xs:date"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The date the changes were applied.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element dataset / maintenance / changeHistory / comment

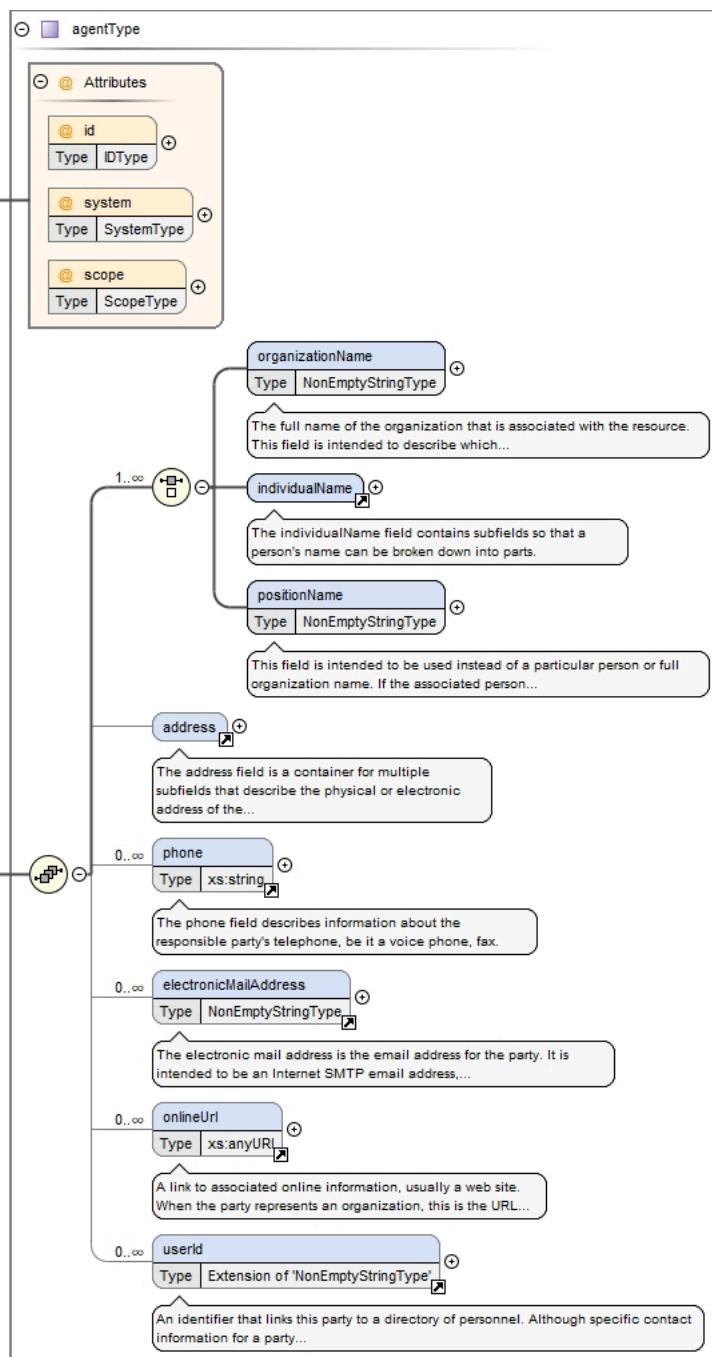
Namespace	No namespace
Annotations	Explanation or justification for the change made to the data.

Diagram	<p>comment Type NonEmptyStringType</p> <p>Explanation or justification for the change made to the data.</p> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p>				
Type	NonEmptyStringType				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ] [ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ] [ \s \S ]*
minLength	1				
pattern	[ \s ]*[ \S ] [ \s \S ]*				
Source	<pre>&lt;xs:element name="comment" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Explanation or justification for the change made to the data.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>				

## Element dataset / contact

Namespace	No namespace
-----------	--------------

## Diagram



Type	agentType
Properties	<p>content: complex</p> <p>maxOccurs: unbounded</p>
Model	( <code>organizationName</code>   <code>individualName</code>   <code>positionName</code> ) , <code>address</code> {0,1} , <code>phone</code> * , <code>electronicMailAddress</code> * , <code>onlineUrl</code> * , <code>userId</code> *
Children	address, <code>electronicMailAddress</code> , <code>individualName</code> , <code>onlineUrl</code> , <code>organizationName</code> , <code>phone</code> , <code>positionName</code> , <code>userId</code>
Instance	<pre>&lt;contact id="" scope="" system=""&gt; &lt;organizationName&gt;{1,1}&lt;/organizationName&gt; &lt;individualName&gt;{1,1}&lt;/individualName&gt; &lt;positionName&gt;{1,1}&lt;/positionName&gt; &lt;address&gt;{0,1}&lt;/address&gt; &lt;phone&gt;{0,unbounded}&lt;/phone&gt; &lt;electronicMailAddress&gt;{0,unbounded}&lt;/electronicMailAddress&gt; &lt;onlineUrl&gt;{0,unbounded}&lt;/onlineUrl&gt; &lt;userId directory=""&gt;{0,unbounded}&lt;/userId&gt; &lt;/contact&gt;</pre>

Attributes	QName	Type	Use
	<b>id</b>	IDType	optional
	<b>scope</b>	ScopeType	optional
	<b>system</b>	SystemType	optional
Source	<xs:element name="contact" type="agentType" maxOccurs="unbounded"/>		

## Element dataset / publisher

Namespace	No namespace				
Annotations	The publisher of this data set. This is typically an institution that is making the data available in a published (ie, citable) format.				
Diagram	<p>The diagram illustrates the structure of the <code>agentType</code> element. It starts with a <code>publisher</code> object (Type: <code>agentType</code>) which has a relationship to the <code>agentType</code> class. The <code>agentType</code> class contains attributes: <code>id</code> (Type: <code>IDType</code>), <code>system</code> (Type: <code>SystemType</code>), and <code>scope</code> (Type: <code>ScopeType</code>). It also has associations with <code>organizationName</code> (Type: <code>NonEmptyStringType</code>), <code>individualName</code> (Type: <code>xs:string</code>), <code>positionName</code> (Type: <code>NonEmptyStringType</code>), <code>address</code> (Type: <code>xs:string</code>), <code>phone</code> (Type: <code>xs:string</code>), <code>electronicMailAddress</code> (Type: <code>NonEmptyStringType</code>), <code>onlineUrl</code> (Type: <code>xs:anyURI</code>), and <code>userid</code> (Type: <code>Extension of 'NonEmptyStringType'</code>). Each association is accompanied by a detailed description box.</p>				
Type	agentType				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				

Model	(organizationName   individualName   positionName) , address{0,1} , phone* , electronicEmailAddress* , onlineUrl* , userId*												
Children	address, electronicEmailAddress, individualName, onlineUrl, organizationName, phone, positionName, userId												
Instance	<pre>&lt;publisher id="" scope="" system=""&gt;   &lt;organizationName&gt;{1,1}&lt;/organizationName&gt;   &lt;individualName&gt;{1,1}&lt;/individualName&gt;   &lt;positionName&gt;{1,1}&lt;/positionName&gt;   &lt;address&gt;{0,1}&lt;/address&gt;   &lt;phone&gt;{0,unbounded}&lt;/phone&gt;   &lt;electronicEmailAddress&gt;{0,unbounded}&lt;/electronicEmailAddress&gt;   &lt;onlineUrl&gt;{0,unbounded}&lt;/onlineUrl&gt;   &lt;userId directory=""&gt;{0,unbounded}&lt;/userId&gt; &lt;/publisher&gt;</pre>												
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>id</b></td> <td>IDType</td> <td>optional</td> </tr> <tr> <td><b>scope</b></td> <td>ScopeType</td> <td>optional</td> </tr> <tr> <td><b>system</b></td> <td>SystemType</td> <td>optional</td> </tr> </tbody> </table>	QName	Type	Use	<b>id</b>	IDType	optional	<b>scope</b>	ScopeType	optional	<b>system</b>	SystemType	optional
QName	Type	Use											
<b>id</b>	IDType	optional											
<b>scope</b>	ScopeType	optional											
<b>system</b>	SystemType	optional											
Source	<pre>&lt;x:element name="publisher" type="agentType" minOccurs="0"&gt;   &lt;x:annotation&gt;     &lt;x:documentation&gt;The publisher of this data set. This is typically an institution that is making the data available in a published (ie, citable) format.&lt;/x:documentation&gt;   &lt;/x:annotation&gt; &lt;/x:element&gt;</pre>												

## Element methods

Namespace	No namespace
Annotations	The methods field documents scientific methods used in the collection of this dataset. It includes information on items such as tools, instrument calibration and software.
Diagram	<pre> classDiagram     class methods {         &lt;&lt;The methods field documents scientific methods used in the collection of this dataset. It includes information on items...&gt;&gt;     }     class methodStep {         &lt;&lt;The methodStep field allows for repeated sets of elements that document a series of procedures followed to produce a...&gt;&gt;     }     class sampling {         &lt;&lt;Description of sampling procedures including the geographic, temporal and taxonomic coverage of the study.&gt;&gt;     }     class qualityControl {         &lt;&lt;The qualityControl field provides a location for the description of actions taken to either control or assess the...&gt;&gt;     }      methods "1..∞" -- "0..∞" methodStep     methods "1..∞" -- "0..∞" sampling     methods "1..∞" -- "0..∞" qualityControl   </pre>
Properties	content: complex
Used by	Element dataset
Model	methodStep , sampling{0,1} , qualityControl*
Children	methodStep, qualityControl, sampling
Instance	<pre>&lt;methods&gt;   &lt;methodStep&gt;{1,1}&lt;/methodStep&gt;   &lt;sampling&gt;{0,1}&lt;/sampling&gt;   &lt;qualityControl&gt;{0,unbounded}&lt;/qualityControl&gt; &lt;/methods&gt;</pre>
Source	<pre>&lt;x:element name="methods"&gt;   &lt;x:annotation&gt;     &lt;x:documentation&gt;The methods field documents scientific methods used in the collection of this dataset. It includes information on items such as tools, instrument calibration and software.&lt;/x:documentation&gt;   &lt;/x:annotation&gt;   &lt;x:complexType&gt;     &lt;x:sequence maxOccurs="unbounded"&gt;       &lt;x:element name="methodStep" type="description"&gt;         &lt;x:annotation&gt;           &lt;x:documentation&gt;The methodStep field allows for repeated sets of elements that document a series of procedures followed to produce a data object. These include text descriptions of the procedures, relevant literature, software, instrumentation, source data and any quality control measures taken.&lt;/x:documentation&gt;         &lt;/x:annotation&gt;       &lt;/x:element&gt;       &lt;x:element name="sampling" minOccurs="0"&gt;</pre>

```

<xs:annotation>
  <xs:documentation>Description of sampling procedures including the geographic, temporal and taxonomic coverage of the study.</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="studyExtent" type="description">
        <xs:annotation>
          <xs:documentation>The field studyExtent represents both a specific sampling area and the sampling frequency (temporal boundaries, frequency of occurrence). The geographic studyExtent is usually a surrogate (representative area of) for the larger area documented in the "studyAreaDescription".</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="samplingDescription">
        <xs:annotation>
          <xs:documentation>The samplingDescription field allows for a text-based/human readable description of the sampling procedures used in the research project. The content of this element would be similar to a description of sampling procedures found in the methods section of a journal article.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="para"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="qualityControl" type="description" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>The qualityControl field provides a location for the description of actions taken to either control or assess the quality of data resulting from the associated method step.</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

## Element methods / methodStep

Namespace	No namespace
Annotations	The methodStep field allows for repeated sets of elements that document a series of procedures followed to produce a data object. These include text descriptions of the procedures, relevant literature, software, instrumentation, source data and any quality control measures taken.
Diagram	<p>The diagram shows a UML class named 'methodStep' with a 'Type' attribute and a 'description' attribute. The 'description' attribute is marked with a multiplicity of *..*, indicating it is a repeating element. A callout box points to this attribute with the text: 'The methodStep field allows for repeated sets of elements that document a series of procedures followed to produce a data object.' Another callout box points to the 'description' element with the text: 'The field Description contains general textual descriptions.'</p>
Type	description
Properties	content: complex
Model	description
Children	description
Instance	<pre>&lt;methodStep&gt;   &lt;description&gt;{1,1}&lt;/description&gt; &lt;/methodStep&gt;</pre>
Source	<pre>&lt;xs:element name="methodStep" type="description"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The methodStep field allows for repeated sets of elements that document a series of procedures followed to produce a data object. These include text descriptions of the procedures, relevant literature, software, instrumentation, source data and any quality control measures taken.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element methods / sampling

Namespace	No namespace
-----------	--------------

Annotations	Description of sampling procedures including the geographic, temporal and taxonomic coverage of the study.				
Diagram	<pre> classDiagram     class sampling {         &lt;&lt;Description of sampling procedures including the geographic, temporal and taxonomic coverage of the study.&gt;&gt;     }     class studyExtent {         &lt;&lt;The field studyExtent represents both a specific sampling area and the sampling frequency (temporal boundaries,...)&gt;&gt;     }     class samplingDescription {         &lt;&lt;The samplingDescription field allows for a text-based/human readable description of the sampling procedures used in the...&gt;&gt;     }      sampling "1..1" --&gt; studyExtent :      sampling "1..1" --&gt; samplingDescription :   </pre>				
Properties	<table> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	studyExtent , samplingDescription				
Children	samplingDescription, studyExtent				
Instance	<sampling>   <studyExtent>{1,1}</studyExtent>   <samplingDescription>{1,1}</samplingDescription> </sampling>				
Source	<pre> &lt;xs:element name="sampling" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Description of sampling procedures including the geographic, temporal and taxonomic coverage of the study.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="studyExtent" type="description"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The field studyExtent represents both a specific sampling area and the sampling frequency (temporal boundaries, frequency of occurrence). The geographic studyExtent is usually a surrogate (representative area of) for the larger area documented in the "studyAreaDescription".&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="samplingDescription"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The samplingDescription field allows for a text-based/human readable description of the sampling procedures used in the research project. The content of this element would be similar to a description of sampling procedures found in the methods section of a journal article.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>				

### Element methods / sampling / studyExtent

Namespace	No namespace		
Annotations	The field studyExtent represents both a specific sampling area and the sampling frequency (temporal boundaries, frequency of occurrence). The geographic studyExtent is usually a surrogate (representative area of) for the larger area documented in the "studyAreaDescription".		
Diagram	<pre> classDiagram     class studyExtent {         &lt;&lt;The field studyExtent represents both a specific sampling area and the sampling frequency (temporal boundaries,...)&gt;&gt;     }     class description {         &lt;&lt;The field Description contains general textual descriptions.&gt;&gt;     }      studyExtent "1..1" --&gt; description :   </pre>		
Type	description		
Properties	<table> <tr> <td>content:</td> <td>complex</td> </tr> </table>	content:	complex
content:	complex		

Model	description
Children	description
Instance	<pre>&lt;studyExtent&gt;   &lt;description&gt;{1,1}&lt;/description&gt; &lt;/studyExtent&gt;</pre>
Source	<pre>&lt;xs:element name="studyExtent" type="description"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The field studyExtent represents both a specific sampling area and the sampling frequency (temporal boundaries, frequency of occurrence). The geographic studyExtent is usually a surrogate (representative area of) for the larger area documented in the "studyAreaDescription".&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element methods / sampling / samplingDescription

Namespace	No namespace
Annotations	The samplingDescription field allows for a text-based/human readable description of the sampling procedures used in the research project. The content of this element would be similar to a description of sampling procedures found in the methods section of a journal article.
Diagram	<p>The diagram illustrates the structure of the samplingDescription element. It consists of two main components: 'samplingDescription' and 'para'. An association line connects them. A callout box points to the 'samplingDescription' node with the text: 'The samplingDescription field allows for a text-based/human readable description of the sampling procedures used in the...'. Another callout box points to the 'para' node with the text: 'The "paragraph" element allows for text blocks to be included in EML.'</p>
Properties	content: complex
Model	para
Children	para
Instance	<pre>&lt;samplingDescription&gt;   &lt;para&gt;{1,1}&lt;/para&gt; &lt;/samplingDescription&gt;</pre>
Source	<pre>&lt;xs:element name="samplingDescription"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The samplingDescription field allows for a text-based/human readable description of the sampling procedures used in the research project. The content of this element would be similar to a description of sampling procedures found in the methods section of a journal article.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="para" /&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

## Element methods / qualityControl

Namespace	No namespace						
Annotations	The qualityControl field provides a location for the description of actions taken to either control or assess the quality of data resulting from the associated method step.						
Diagram	<p>The diagram illustrates the structure of the qualityControl element. It consists of two main components: 'qualityControl' and 'description'. An association line connects them. A callout box points to the 'qualityControl' node with the text: 'The qualityControl field provides a location for the description of actions taken to either control or assess the...'. Another callout box points to the 'description' node with the text: 'The field Description contains general textual descriptions.'</p>						
Type	description						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	description						

Children	description
Instance	<pre>&lt;qualityControl&gt;   &lt;description&gt;{1,1}&lt;/description&gt; &lt;/qualityControl&gt;</pre>
Source	<pre>&lt;xs:element name="qualityControl" type="description" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The qualityControl field provides a location for the description of actions taken to either control or assess the quality of data resulting from the associated method step.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element project

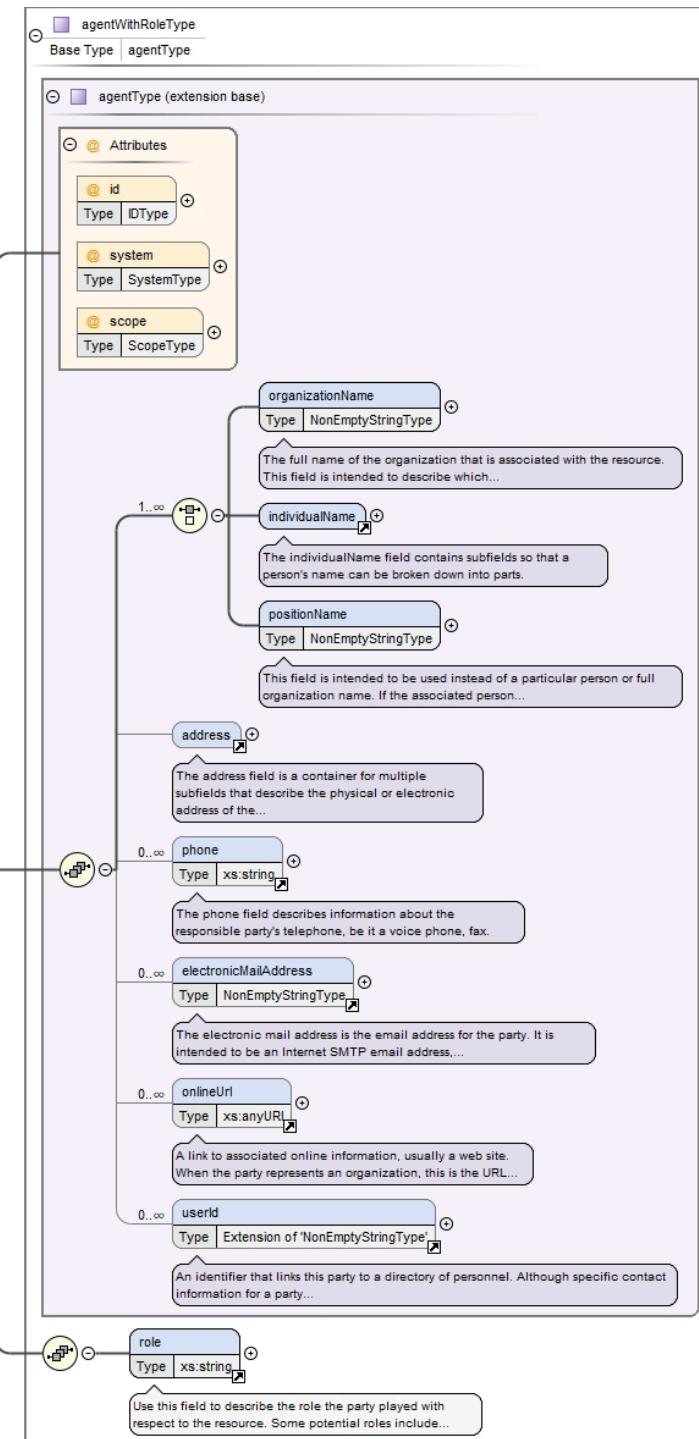
Namespace	No namespace
Annotations	The project field contains information on the project in which this dataset was collected. It includes information such as project personnel, funding, study area, project design and related projects.
Diagram	<pre> classDiagram     class project {         &lt;&lt;The project field contains information on the project in which this dataset was collected. It includes information such...&gt;&gt;         &lt;&lt;The project field contains information on the project in which this dataset was collected. It includes information such...&gt;&gt;         &lt;&lt;A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with...&gt;&gt;         &lt;&lt;The 'title' field provides a description of the resource that is being documented that is long enough to differentiate...&gt;&gt;         &lt;&lt;The Personnel field extends ResponsibleParty with role information and is used to document people involved in a...&gt;&gt;         &lt;&lt;A brief overview describing the dataset&gt;&gt;         &lt;&lt;The funding field is used to provide information about funding sources for the project such as: grant and contract...&gt;&gt;         &lt;&lt;The award field is used to provide specific information about the funding awards for a project in a structured format...&gt;&gt;         &lt;&lt;The studyAreaDescription field documents the physical area associated with the research project. It can include...&gt;&gt;         &lt;&lt;The field designDescription contains general textual descriptions of research design. It can include detailed accounts...&gt;&gt;         &lt;&lt;This field is a recursive link to another project. This allows projects to be nested under one another in the case...&gt;&gt;     }     class projectType {         @id xs:string         title i18nString         personnel agentWithRoleType         abstract TextType         funding         award awardType         studyAreaDescription         designDescription description         relatedProject relatedProjectType     }     project "3" --&gt; "3" : project     project "3" --&gt; projectType : project   </pre>
Type	projectType
Properties	content: complex

Used by	Element dataset		
Model	(title , personnel+ , abstract{0,1} , funding{0,1} , award* , studyAreaDescription{0,1} , designDescription{0,1} , relatedProject*)		
Children	abstract, award, designDescription, funding, personnel, relatedProject, studyAreaDescription, title		
Instance	<pre>&lt;project id=""&gt;   &lt;title xml:lang=""&gt;{1,1}&lt;/title&gt;   &lt;personnel id="" scope="" system=""&gt;{1,unbounded}&lt;/personnel&gt;   &lt;abstract xml:lang=""&gt;{0,1}&lt;/abstract&gt;   &lt;funding&gt;{0,1}&lt;/funding&gt;   &lt;award&gt;{0,unbounded}&lt;/award&gt;   &lt;studyAreaDescription&gt;{0,1}&lt;/studyAreaDescription&gt;   &lt;designDescription&gt;{0,1}&lt;/designDescription&gt;   &lt;relatedProject id=""&gt;{0,unbounded}&lt;/relatedProject&gt; &lt;/project&gt;</pre>		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>id</b>	xs:string	optional
		A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.	
Source	<pre>&lt;xs:element name="project" type="projectType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The project field contains information on the project in which this dataset was collected. It includes information such as project personnel, funding, study area, project design and related projects.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>		

## Element **projectType** / **personnel**

Namespace	No namespace
Annotations	The Personnel field extends ResponsibleParty with role information and is used to document people involved in a research project by providing contact information and their role in the project.

## Diagram



Type	<code>agentWithRoleType</code>
Type hierarchy	<ul style="list-style-type: none"> <li>• <code>agentType</code></li> <li>• <code>agentWithRoleType</code></li> </ul>
Properties	<p>content: complex</p> <p>maxOccurs: unbounded</p>
Model	<code>(organizationName   individualName   positionName) , address{0,1} , phone* , electronicMailAddress* , onlineUrl* , userId* , role</code>
Children	address, electronicMailAddress, individualName, onlineUrl, organizationName, phone, positionName, role, userId
Instance	<pre>&lt;personnel id="" scope="" system=""&gt;   &lt;organizationName&gt;{1,1}&lt;/organizationName&gt;   &lt;individualName&gt;{1,1}&lt;/individualName&gt;   &lt;positionName&gt;{1,1}&lt;/positionName&gt;</pre>

	<pre> &lt;address&gt;{0,1}&lt;/address&gt; &lt;phone&gt;{0,unbounded}&lt;/phone&gt; &lt;electronicEmailAddress&gt;{0,unbounded}&lt;/electronicEmailAddress&gt; &lt;onlineUrl&gt;{0,unbounded}&lt;/onlineUrl&gt; &lt;userId directory=""&gt;{0,unbounded}&lt;/userId&gt; &lt;role&gt;{1,1}&lt;/role&gt; &lt;/personnel&gt; </pre>
Attributes	<b>QName</b>
	<b>id</b>
	<b>scope</b>
Source	<b>system</b>
	<pre> &lt;xs:element name="personnel" type="agentWithRoleType" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The Personnel field extends ResponsibleParty with role information and is used to document people involved in a research project by providing contact information and their role in the project.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>
	The Personnel field extends ResponsibleParty with role information and is used to document people involved in a research project by providing contact information and their role in the project.

## Element projectType / funding

Namespace	No namespace				
Annotations	The funding field is used to provide information about funding sources for the project such as: grant and contract numbers; names and addresses of funding sources.				
Diagram					
Properties	<table> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	para				
Children	para				
Instance	<pre> &lt;funding&gt;   &lt;para&gt;{1,1}&lt;/para&gt; &lt;/funding&gt; </pre>				
Source	<pre> &lt;xs:element name="funding" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The funding field is used to provide information about funding sources for the project such as: grant and contract numbers; names and addresses of funding sources.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="para"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>				

## Element projectType / award

Namespace	No namespace
Annotations	The award field is used to provide specific information about the funding awards for a project in a structured format. Sub-fields are provided for the name of the funding agency, the Open Funder Registry identifiers for the agency and program that made the award, the award number assigned, the title of the award, and the URL to the award page describing the award. In general, the funding agency should be listed with a cross-reference to the appropriate identifier from the Open Funder Registry (included in the EML distribution but updated periodically from the Open Funder Registry).

Diagram	<p>The award field is used to provide specific information about the funding awards for a project in a structured format....</p>						
Type	awardType						
Properties	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">content:</td><td style="padding: 2px;">complex</td></tr> <tr> <td style="padding: 2px;">minOccurs:</td><td style="padding: 2px;">0</td></tr> <tr> <td style="padding: 2px;">maxOccurs:</td><td style="padding: 2px;">unbounded</td></tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	funderName , funderIdentifier* , awardNumber{0,1} , title , awardUrl{0,1}						
Children	awardNumber, awardUrl, funderIdentifier, funderName, title						
Instance	<pre>&lt;award&gt;   &lt;funderName&gt;{1,1}&lt;/funderName&gt;   &lt;funderIdentifier&gt;{0,unbounded}&lt;/funderIdentifier&gt;   &lt;awardNumber&gt;{0,1}&lt;/awardNumber&gt;   &lt;title&gt;{1,1}&lt;/title&gt;   &lt;awardUrl&gt;{0,1}&lt;/awardUrl&gt; &lt;/award&gt;</pre>						
Source	<pre>&lt;xs:element name="award" type="awardType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The award field is used to provide specific information about the funding awards for a project in a structured format. Sub-fields are provided for the name of the funding agency, the Open Funder Registry identifiers for the agency and program that made the award, the award number assigned, the title of the award, and the URL to the award page describing the award. In general, the funding agency should be listed with a cross-reference to the appropriate identifier from the Open Funder Registry (included in the EML distribution but updated periodically from the Open Funder Registry).&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>						

### Element awardType / funderName

Namespace	No namespace
Annotations	The name of the funding institution, with fully expanded acronyms to show the full, official name of the funding agency. In general, this should match the official name of the funder as listed in an Authority such as the Open Funder Registry. The Open Funder Registry and other organizational authorities may provide a list of other alternative names for the funding agency.
Diagram	<p>The name of the funding institution, with fully expanded acronyms to show the full, official name of the funding...</p>

Type	NonEmptyStringType
Properties	content: simple minOccurs: 1 maxOccurs: 1
Facets	minLength 1 pattern [\s]*[\S][\s\S]*
Source	<pre>&lt;xs:element name="funderName" type="NonEmptyStringType" minOccurs="1" maxOccurs="1"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The name of the funding institution, with fully expanded acronyms to show the full, official name of the funding agency. In general, this should match the official name of the funder as listed in an Authority such as the Open Funder Registry. The Open Funder Registry and other organizational authorities may provide a list of other alternative names for the funding agency.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element awardType / funderIdentifier

Namespace	No namespace
Annotations	The funder identifier is used to provide one or more canonical identifiers that reference the funder. These identifiers should be globally unique. The most common form of a funder identifier is a DOI identifier of an institution or program drawn from the CrossRef Open Funder Registry ( <a href="https://gitlab.com/crossref/open_funder_registry">https://gitlab.com/crossref/open_funder_registry</a> ), which assigns DOIs to each funding agency and to their programs, and links these together in a navigable hierarchy. A copy of the current Funder Registry is included as an RDF file with EML for reference, but as the list is constantly growing, users can retrieve new copies of the RDF file to get updates and current metadata about funders.
Diagram	<p>The diagram shows a UML class named "funderIdentifier" with a constraint "Type NonEmptyStringType". An association line connects "funderIdentifier" to a class represented by a purple rounded rectangle labeled "NonEmptyStringType". There are two callouts: one pointing to "funderIdentifier" stating "The funder identifier is used to provide one or more canonical identifiers that reference the funder. These identifiers...", and another pointing to "NonEmptyStringType" stating "Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...".</p>
Type	NonEmptyStringType
Properties	content: simple minOccurs: 0 maxOccurs: unbounded
Facets	minLength 1 pattern [\s]*[\S][\s\S]*
Source	<pre>&lt;xs:element name="funderIdentifier" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The funder identifier is used to provide one or more canonical identifiers that reference the funder. These identifiers should be globally unique. The most common form of a funder identifier is a DOI identifier of an institution or program drawn from the CrossRef Open Funder Registry (<a href="https://gitlab.com/crossref/open_funder_registry">https://gitlab.com/crossref/open_funder_registry</a>), which assigns DOIs to each funding agency and to their programs, and links these together in a navigable hierarchy. A copy of the current Funder Registry is included as an RDF file with EML for reference, but as the list is constantly growing, users can retrieve new copies of the RDF file to get updates and current metadata about funders.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element awardType / awardNumber

Namespace	No namespace
Annotations	The awardNumber field provides the unique identifier used by the funder to uniquely identify an award. These are typically alphanumeric values that are unique within the system used by a given funder. The number should be listed using the canonical form that each funder uses to express its award numbers, and not be prefixed or postfixed with extra text such as the acronym of the funder or the name of the funder, which is available instead in the funderName field.

Diagram	A UML class diagram showing 'awardNumber' as a type of 'NonEmptyStringType'. A callout box next to 'awardNumber' states: 'The awardNumber field provides the unique identifier used by the funder to uniquely identify an award. These are...'. A callout box next to 'NonEmptyStringType' states: 'Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...'.
Type	NonEmptyStringType
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: 1</p>
Facets	<p>minLength 1</p> <p>pattern <math>[\s]*[\S][\s\S]*</math></p>
Source	<pre>&lt;xs:element name="awardNumber" type="NonEmptyStringType" minOccurs="0" maxOccurs="1"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The awardNumber field provides the unique identifier used by the funder to uniquely identify an award. These are typically alphanumeric values that are unique within the system used by a given funder. The number should be listed using the canonical form that each funder uses to express its award numbers, and not be prefixed or postfixed with extra text such as the acronym of the funder or the name of the funder, which is available instead in the funderName field.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element awardType / title

Namespace	No namespace
Annotations	The title field is used for the title of the award or grant being described.
Diagram	A UML class diagram showing 'title' as a type of 'NonEmptyStringType'. A callout box next to 'title' states: 'The title field is used for the title of the award or grant being described.'. A callout box next to 'NonEmptyStringType' states: 'Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...'.
Type	NonEmptyStringType
Properties	<p>content: simple</p> <p>minOccurs: 1</p> <p>maxOccurs: 1</p>
Facets	<p>minLength 1</p> <p>pattern <math>[\s]*[\S][\s\S]*</math></p>
Source	<pre>&lt;xs:element name="title" type="NonEmptyStringType" minOccurs="1" maxOccurs="1"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The title field is used for the title of the award or grant being described.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element awardType / awardUrl

Namespace	No namespace
Annotations	Typically, the awardUrl is use to find and locate the award, and generally addresses the internet location to find out more information about the award. This should point to a funder site for the award, rather than a project site.
Diagram	A UML class diagram showing 'awardUrl' as a type of 'NonEmptyStringType'. A callout box next to 'awardUrl' states: 'Typically, the awardUrl is use to find and locate the award, and generally addresses the internet location to find out...'. A callout box next to 'NonEmptyStringType' states: 'Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...'.
Type	NonEmptyStringType
Properties	<p>content: simple</p> <p>minOccurs: 0</p>

	maxOccurs:	1
Facets	minLength	1
	pattern	[ \s]*[\S][\s\S]*

Source	<pre>&lt;xs:element name="awardUrl" type="NonEmptyStringType" minOccurs="0" maxOccurs="1"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Typically, the awardUrl is used to find and locate the award, and generally addresses the internet location to find out more information about the award. This should point to a funder site for the award, rather than a project site.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>
--------	---

## Element projectType / studyAreaDescription

Namespace	No namespace				
Annotations	The studyAreaDescription field documents the physical area associated with the research project. It can include descriptions of the geographic, temporal, and taxonomic coverage of the research location and descriptions of domains (themes) of interest such as climate, geology, soils or disturbances.				
Diagram	<p>The diagram shows a class named 'studyAreaDescription' connected to a class named 'descriptor' via a line with a multiplicity of '0..1' at the 'studyAreaDescription' end and '1' at the 'descriptor' end. A note below the 'studyAreaDescription' class states: 'The studyAreaDescription field documents the physical area associated with the research project. It can include...'. A note below the 'descriptor' class states: 'The descriptor field is used to document domains (themes) of interest such as climate, geology, soils or disturbances.'</p>				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				
Model	descriptor				
Children	descriptor				
Instance	<pre>&lt;studyAreaDescription&gt;   &lt;descriptor citableClassificationSystem="" name=""&gt;{1,1}&lt;/descriptor&gt; &lt;/studyAreaDescription&gt;</pre>				
Source	<pre>&lt;xs:element name="studyAreaDescription" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The studyAreaDescription field documents the physical area associated with the research project. It can include descriptions of the geographic, temporal, and taxonomic coverage of the research location and descriptions of domains (themes) of interest such as climate, geology, soils or disturbances.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="descriptor"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>				

## Element descriptor

Namespace	No namespace		
Annotations	The descriptor field is used to document domains (themes) of interest such as climate, geology, soils or disturbances.		
Diagram	<p>The diagram shows a class named 'descriptor' connected to a class named 'descriptorValue' via a line with a multiplicity of '0..1' at both ends. Both classes are connected to a box labeled 'Attributes' which contains two entries: '@ citableClassificationSystem' with type 'xs:boolean' and '@ name' with type 'descriptorEnum'. A note below the 'descriptor' class states: 'The descriptor field is used to document domains (themes) of interest such as climate, geology, soils or disturbances.'. A note below the 'descriptorValue' class states: 'The descriptorValue field contains a general description, either thematic or geographic, of the study area.'</p>		
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> </table>	content:	complex
content:	complex		

Used by	Element projectType/studyAreaDescription		
Model	descriptorValue		
Children	descriptorValue		
Instance	<descriptor citableClassificationSystem="" name=""> <descriptorValue>{1,1}</descriptorValue> </descriptor>		
Attributes	QName	Type	Use
	citableClassificationSystem	xs:boolean	optional
	name	descriptorEnum	optional
Source	<pre>&lt;xs:element name="descriptor"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The descriptor field is used to document domains (themes) of interest such as climate, geology, soils or disturbances.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="descriptorValue"/&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="citableClassificationSystem" use="optional" type="xs:boolean"/&gt;     &lt;xs:attribute name="name" use="optional" type="descriptorEnum"/&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>		

## Element descriptorValue

Namespace	No namespace
Annotations	The descriptorValue field contains a general description, either thematic or geographic, of the study area.
Diagram	<p>The diagram shows a class named 'descriptorValue' with a single attribute 'Type' of type 'xs:string'. A callout box indicates that the 'descriptorValue' field contains a general description, either thematic or geographic, of the study area. Another callout box shows a purple rectangle labeled 'xs:string' with the text 'Built-in primitive type. The string datatype represents character strings in XML.'</p>
Type	xs:string
Properties	content: simple
Used by	Element descriptor
Source	<pre>&lt;xs:element name="descriptorValue" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The descriptorValue field contains a general description, either thematic or geographic, of the study area.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element projectType / designDescription

Namespace	No namespace				
Annotations	The field designDescription contains general textual descriptions of research design. It can include detailed accounts of goals, motivations, theory, hypotheses, strategy, statistical design, and actual work.				
Diagram	<p>The diagram shows a class named 'designDescription' with a single attribute 'Type' of type 'description'. This 'description' type is shown as a composite structure containing multiple 'description' elements, each with a '+' sign indicating multiplicity. A callout box states: 'The field designDescription contains general textual descriptions of research design. It can include detailed accounts..'. Another callout box shows a purple rectangle labeled 'description' with the text 'The field Description contains general textual descriptions.'</p>				
Type	description				
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0
content:	complex				
minOccurs:	0				

Model	description
Children	description
Instance	<designDescription> <description>{1,1}</description> </designDescription>
Source	<pre>&lt;xss:element name="designDescription" type="description" minOccurs="0"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;The field designDescription contains general textual descriptions of research design. It can include detailed accounts of goals, motivations, theory, hypotheses, strategy, statistical design, and actual work.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>

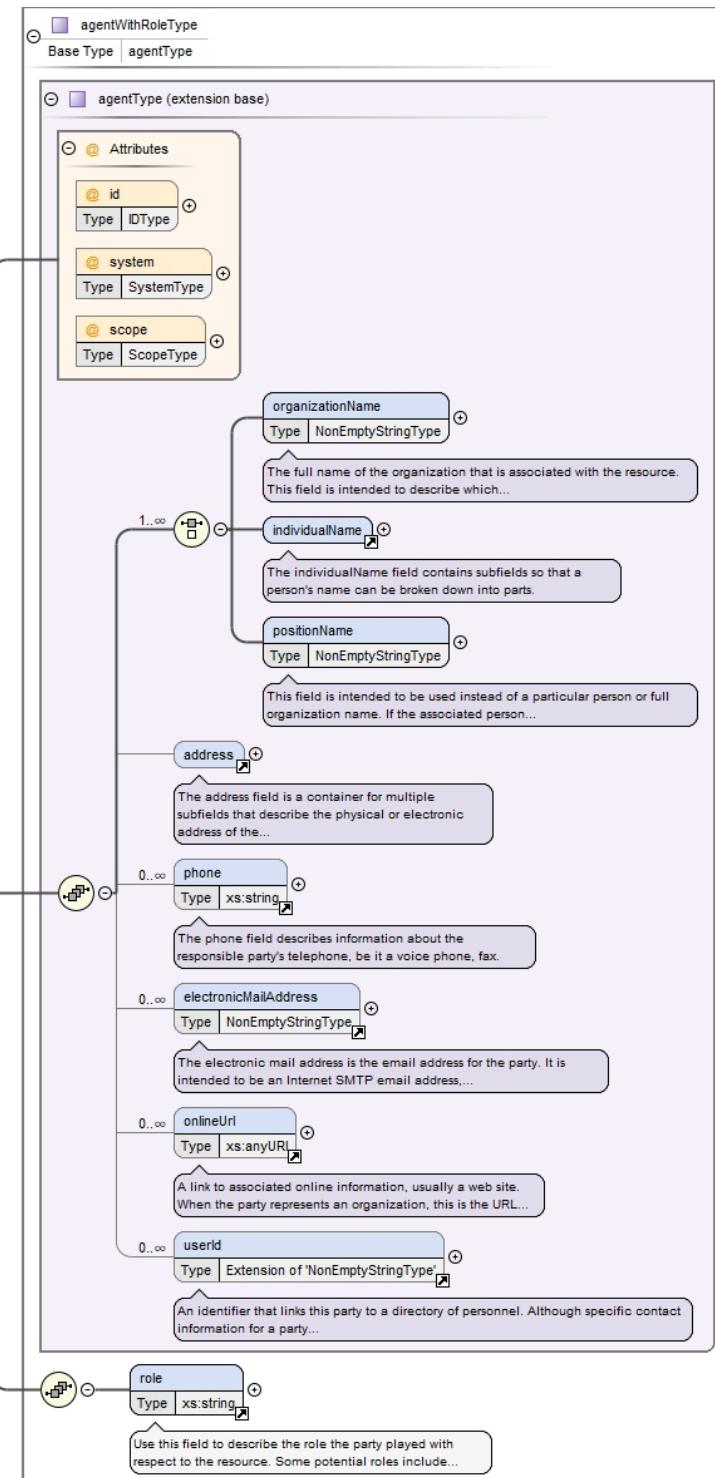
## Element projectType / relatedProject

Namespace	No namespace									
Annotations	This field is a recursive link to another project. This allows projects to be nested under one another in the case where one project spawns another.									
Diagram	<p>The diagram illustrates the structure of the <code>relatedProjectType</code> element. It features a central class box labeled <code>relatedProjectType</code> containing attributes <code>@id</code> (xs:string) and <code>title</code> (i18nString), and associations with <code>relatedProject</code>, <code>personnel</code>, and <code>abstract</code>. A callout box provides a detailed description of the <code>id</code> attribute as a unique identifier for the project. Another callout box describes the <code>title</code> attribute as providing a description of the resource being documented. The <code>personnel</code> association is marked with a multiplicity of <code>1..oo</code>, and the <code>abstract</code> association is marked with a multiplicity of <code>{0,1}</code>.</p>									
Type	relatedProjectType									
Properties	<table> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded			
content:	complex									
minOccurs:	0									
maxOccurs:	unbounded									
Model	(title , personnel+, abstract{0,1})									
Children	abstract, personnel, title									
Instance	<pre>&lt;relatedProject id=""&gt;   &lt;title xml:lang=""&gt;{1,1}&lt;/title&gt;   &lt;personnel id="" scope="" system=""&gt;{1,unbounded}&lt;/personnel&gt;   &lt;abstract xml:lang=""&gt;{0,1}&lt;/abstract&gt; &lt;/relatedProject&gt;</pre>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>id</b></td> <td>xs:string</td> <td>optional</td> </tr> <tr> <td></td> <td></td> <td>A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.</td> </tr> </tbody> </table>	QName	Type	Use	<b>id</b>	xs:string	optional			A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.
QName	Type	Use								
<b>id</b>	xs:string	optional								
		A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.								
Source	<pre>&lt;xss:element name="relatedProject" type="relatedProjectType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;This field is a recursive link to another project. This allows projects to be nested under one another in the case where one project spawns another.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;</pre>									

## Element relatedProjectType / personnel

Namespace	No namespace
-----------	--------------

## Diagram



Type	<code>agentWithRoleType</code>
Type hierarchy	<ul style="list-style-type: none"> <li>• <code>agentType</code></li> <li>• <code>agentWithRoleType</code></li> </ul>
Properties	content: complex maxOccurs: unbounded
Model	<code>(organizationName   individualName   positionName) , address{0,1} , phone* , electronicMailAddress* , onlineUrl* , userId* , role</code>
Children	address, electronicMailAddress, individualName, onlineUrl, organizationName, phone, positionName, role, userId
Instance	<pre>&lt;personnel id="" scope="" system=""&gt;   &lt;organizationName&gt;{1,1}&lt;/organizationName&gt;   &lt;individualName&gt;{1,1}&lt;/individualName&gt;</pre>

	<pre> &lt;positionName&gt;{1,1}&lt;/positionName&gt; &lt;address&gt;{0,1}&lt;/address&gt; &lt;phone&gt;{0,unbounded}&lt;/phone&gt; &lt;electronicMailAddress&gt;{0,unbounded}&lt;/electronicMailAddress&gt; &lt;onlineUrl&gt;{0,unbounded}&lt;/onlineUrl&gt; &lt;userId directory=""&gt;{0,unbounded}&lt;/userId&gt; &lt;role&gt;{1,1}&lt;/role&gt; &lt;/personnel&gt; </pre>													
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td><b>id</b></td><td>IDType</td><td>optional</td></tr> <tr> <td><b>scope</b></td><td>ScopeType</td><td>optional</td></tr> <tr> <td><b>system</b></td><td>SystemType</td><td>optional</td></tr> </tbody> </table>	QName	Type	Use	<b>id</b>	IDType	optional	<b>scope</b>	ScopeType	optional	<b>system</b>	SystemType	optional	
QName	Type	Use												
<b>id</b>	IDType	optional												
<b>scope</b>	ScopeType	optional												
<b>system</b>	SystemType	optional												
Source	<code>&lt;xss:element name="personnel" type="agentWithRoleType" maxOccurs="unbounded"/&gt;</code>													

## Element dataset / literatureCited

Namespace	No namespace						
Annotations	A citation to articles or products which were referenced in the dataset or its associated metadata. The list represents the bibliography of works related to the dataset, whether for reference, comparison, or others purposes.						
Diagram	<pre> classDiagram     class literatureCited {         &lt;&lt;CitationListType&gt;&gt;     }     class bibtex {         &lt;&lt;xs:string&gt;&gt;     }     literatureCited "1..oo" --&gt; bibtex </pre> <p>A citation to articles or products which were referenced in the dataset or its associated metadata. The list represents the bibliography of works related to the dataset, whether for reference, comparison, or others purposes.</p> <p>The bibtex field provides a parseable list of citations formatted according to the Bibtex formatting conventions. Each...</p>						
Type	CitationListType						
Properties	<table> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	bibtex						
Children	bibtex						
Instance	<pre> &lt;literatureCited&gt;   &lt;bibtex&gt;{1,1}&lt;/bibtex&gt; &lt;/literatureCited&gt; </pre>						
Source	<code>&lt;xss:element name="literatureCited" type="CitationListType" minOccurs="0" maxOccurs="unbounded"&gt;</code> <code>&lt;xss:annotation&gt;</code> <code>&lt;xss:documentation&gt;A citation to articles or products which were referenced in the dataset or its associated metadata. The list represents the bibliography of works related to the dataset, whether for reference, comparison, or others purposes.&lt;/xss:documentation&gt;</code> <code>&lt;/xss:annotation&gt;</code> <code>&lt;/xss:element&gt;</code>						

## Element CitationListType / bibtex

Namespace	No namespace
Annotations	The bibtex field provides a parseable list of citations formatted according to the Bibtex formatting conventions. Each citation entry is assigned a unique key that must be unique across all bibtex fields in the EML document. The citation key can be used in markdown sections of the text to refer to this citation using the pandoc-style of inline citation keys. See the markdown element for more details. The record is delimited using curly braces. Most reference software can both import and export citations in Bibtex format, so this is a simpler representation to produce and consume than native EML citation representations.
Diagram	<pre> classDiagram     class bibtex {         &lt;&lt;xs:string&gt;&gt;     }     xs:string     bibtex --&gt; xs:string </pre> <p>The bibtex field provides a parseable list of citations formatted according to the Bibtex formatting conventions. Each...</p> <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string

Properties	content: simple
Source	<pre>&lt;xs:element name="bibtex" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The bibtex field provides a parseable list of citations formatted according to the Bibtex formatting conventions. Each citation entry is assigned a unique key that must be unique across all bibtex fields in the EML document. The citation key can be used in markdown sections of the text to refer to this citation using the pandoc-style of inline citation keys. See the markdown element for more details. The record is delimited using curly braces. Most reference software can both import and export citations in Bibtex format, so this is a simpler representation to produce and consume than native EML citation representations.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element additionalMetadata

Namespace	No namespace
Annotations	A flexible field for including any other relevant metadata that pertains to the resource being described. This field allows EML to be extensible in that any XML-based metadata can be included in this element.
Diagram	<p>This element contains the additional metadata to be included in the document. This element should be used for extending...</p>
Properties	content: complex
Used by	Element eml:eml
Model	metadata
Children	metadata
Instance	<pre>&lt;additionalMetadata&gt;   &lt;metadata&gt;{1,1}&lt;/metadata&gt; &lt;/additionalMetadata&gt;</pre>
Source	<pre>&lt;xs:element name="additionalMetadata"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A flexible field for including any other relevant metadata that pertains to the resource being described. This field allows EML to be extensible in that any XML-based metadata can be included in this element.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="metadata"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This element contains the additional metadata to be included in the document. This element should be used for extending EML to include metadata that is not already available in another part of the EML specification.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="gbif"&gt;               &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;A block of additional metadata used for some special GBIF purposes and crossmapping to other schemas like the TDWG Natural Collection Data (NCD) schema&lt;/xs:documentation&gt;               &lt;/xs:annotation&gt;               &lt;xs:complexType&gt;                 &lt;xs:sequence&gt;                   &lt;!-- additional general metadata --&gt;                   &lt;xs:element name="dateStamp" type="xs:dateTime"&gt;                     &lt;xs:annotation&gt;                       &lt;xs:documentation&gt;The date the metadata document was created or modified.&lt;/xs:documentation&gt;                     &lt;/xs:annotation&gt;                   &lt;/xs:element&gt;                   &lt;xs:element name="hierarchyLevel" type="NonEmptyStringType" minOccurs="0"&gt;                     &lt;xs:annotation&gt;                       &lt;xs:documentation/&gt;                     &lt;/xs:annotation&gt;                   &lt;/xs:element&gt;                   &lt;xs:element name="citation" type="citationType" minOccurs="0"&gt;                     &lt;xs:annotation&gt;                       &lt;xs:documentation&gt;A single citation for use when citing the dataset&lt;/xs:documentation&gt;                     &lt;/xs:annotation&gt;                   &lt;/xs:element&gt;                 &lt;/xs:sequence&gt;               &lt;/xs:complexType&gt;             &lt;/xs:element&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

```

        </xs:annotation>
    </xs:element>
<xs:element name="bibliography" minOccurs="0">
    <xs:annotation>
        <xs:documentation>A list of citations that form a bibliography on literature related / used in the dataset</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="citation" type="citationType" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="physical" maxOccurs="unbounded" minOccurs="0">
    <xs:annotation>
        <xs:documentation>A container element for all of the elements that let you describe the internal/external characteristics and distribution of a data object (e.g., dataObject, dataFormat, distribution) .</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="objectName"/>
            <xs:element ref="characterEncoding" minOccurs="0"/>
            <xs:element ref="dataFormat"/>
            <xs:element ref="distribution"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="resourceLogoUrl" type="xs:anyURI" minOccurs="0">
    <xs:annotation>
        <xs:documentation>URL of the logo associated with a resource</xs:documentation>
    </xs:annotation>
</xs:element>
<!-- additional NCD collection metadata -->
<xs:element name="collection" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>A container element for other elements associated with collections (e.g., collectionIdentifier, collectionName).</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="parentCollectionIdentifier" type="NonEmptyStringType" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Identifier for the parent collection for this sub-collection. Enables a hierarchy of collections and sub collections to be built.</xs:documentation>
                </xs:annotation>
                <xs:element name="collectionIdentifier" type="NonEmptyStringType" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>The URI (LSID or URL) of the collection. In RDF, used as URI of the collection resource.</xs:documentation>
                    </xs:annotation>
                    <xs:element name="collectionName" type="NonEmptyStringType">
                        <xs:annotation>
                            <xs:documentation>Official name of the Collection in the local language</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
</xs:element>
<xs:element name="formationPeriod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>Text description of the time period during which the collection was assembled e.g. "Victorian", or "1922 - 1932", or "c. 1750".</xs:documentation>
    </xs:annotation>
    <xs:element name="specimenPreservationMethod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation>Picklist keyword indicating the process or technique used to prevent physical deterioration of non-living collections. Expected to contain a value from the GBIF Specimen Preservation Method vocabulary</xs:documentation>
        </xs:annotation>
    </xs:element>
<xs:element name="livingTimePeriod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>

```

```

        <xs:documentation>Time period during which biological material was alive. (for
palaeontological collections)</xs:documentation>
        </xs:annotation>
        </xs:element>
        <xs:element ref="jgtiCuratorialUnit" maxOccurs="unbounded" minOccurs="0">
        <xs:annotation>
            <xs:documentation>Pointer to previous version of the document</
        xs:documentation>
        </xs:annotation>
        </xs:element>
        </xs:sequence>
        </xs:complexType>
        </xs:sequence>
        </xs:complexType>
        </xs:element>
        </xs:sequence>
        </xs:complexType>
    </xs:element>

```

## Element additionalMetadata / metadata

Namespace	No namespace
Annotations	This element contains the additional metadata to be included in the document. This element should be used for extending EML to include metadata that is not already available in another part of the EML specification.
Diagram	
Properties	content: complex
Model	gbif
Children	gbif
Instance	<metadata>     <gbif>{1,1}</gbif> </metadata>
Source	<pre> &lt;xs:element name="metadata"&gt;     &lt;xs:annotation&gt;         &lt;xs:documentation&gt;This element contains the additional metadata to be included in the document. This element should be used for extending EML to include metadata that is not already available in another part of the EML specification.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;     &lt;xs:complexType&gt;         &lt;xs:sequence&gt;             &lt;xs:element name="gbif"&gt;                 &lt;xs:annotation&gt;                     &lt;xs:documentation&gt;A block of additional metadata used for some special GBIF purposes and crossmapping to other schemas like the TDWG Natural Collection Data (NCD) schema&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;             &lt;xs:complexType&gt;                 &lt;xs:sequence&gt;                     &lt;!-- additional general metadata --&gt;                     &lt;xs:element name="dateStamp" type="xs:dateTime"&gt;                         &lt;xs:annotation&gt;                             &lt;xs:documentation&gt;The date the metadata document was created or modified.&lt;/                     xs:documentation&gt;                         &lt;/xs:annotation&gt;                     &lt;/xs:element&gt;                     &lt;xs:element name="hierarchyLevel" type="NonEmptyStringType" minOccurs="0"&gt;                         &lt;xs:annotation&gt;                             &lt;xs:documentation/&gt;                         &lt;/xs:annotation&gt;                     &lt;/xs:element&gt;                     &lt;xs:element name="citation" type="citationType" minOccurs="0"&gt;                         &lt;xs:annotation&gt;                             &lt;xs:documentation&gt;A single citation for use when citing the dataset&lt;/                     xs:documentation&gt;                         &lt;/xs:annotation&gt;                     &lt;/xs:element&gt;                     &lt;xs:element name="bibliography" minOccurs="0"&gt; </pre>

```

<xs:annotation>
    <xs:documentation>A list of citations that form a bibliography on literature related / used in the dataset</xs:documentation>
</xs:annotation>
<xs:complexType>
    <xs:sequence>
        <xs:element name="citation" type="citationType" maxOccurs="unbounded" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="physical" maxOccurs="unbounded" minOccurs="0">
    <xs:annotation>
        <xs:documentation>A container element for all of the elements that let you describe the internal/external characteristics and distribution of a data object (e.g., dataObject, dataFormat, distribution) .</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="objectName" />
            <xs:element ref="characterEncoding" minOccurs="0" />
            <xs:element ref="dataFormat" />
            <xs:element ref="distribution" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="resourceLogoUrl" type="xs:anyURI" minOccurs="0">
    <xs:annotation>
        <xs:documentation>URL of the logo associated with a resource</xs:documentation>
    </xs:annotation>
</xs:element>
<!-- additional NCD collection metadata -->
<xs:element name="collection" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>A container element for other elements associated with collections (e.g., collectionIdentifier, collectionName).</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="parentCollectionIdentifier" type="NonEmptyStringType" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Identifier for the parent collection for this sub-collection. Enables a hierarchy of collections and sub collections to be built.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="collectionIdentifier" type="NonEmptyStringType" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>The URI (LSID or URL) of the collection. In RDF, used as URI of the collection resource.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="collectionName" type="NonEmptyStringType">
                <xs:annotation>
                    <xs:documentation>Official name of the Collection in the local language</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="formationPeriod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>Text description of the time period during which the collection was assembled e.g. "Victorian", or "1922 - 1932", or "c. 1750".</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="specimenPreservationMethod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>Picklist keyword indicating the process or technique used to prevent physical deterioration of non-living collections. Expected to contain a value from the GBIF Specimen Preservation Method vocabulary</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="livingTimePeriod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>Time period during which biological material was alive. (for palaeontological collections)</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element ref="jgtiCuratorialUnit" maxOccurs="unbounded" minOccurs="0" />

```

```

<xs:element ref="dc:replaces" minOccurs="0">
  <xs:annotation>
    <xs:documentation>Pointer to previous version of the document</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:elements>

```

## Element additionalMetadata / metadata / gbif

Namespace	No namespace
Annotations	A block of additional metadata used for some special GBIF purposes and crossmapping to other schemas like the TDWG Natural Collection Data (NCD) schema
Diagram	<pre> classDiagram     class gbif {         dateStamp : xs:dateTime         hierarchyLevel : NonEmptyStringType         citation : citationType         bibliography         physical         resourceLogoUrl : xs:anyURI         collection         gbf         dc:replaces     } </pre> <p>The diagram illustrates the structure of the <code>gbif</code> element. It contains several attributes:</p> <ul style="list-style-type: none"> <li><code>dateStamp</code>: Type <code>xs:dateTime</code>. Description: The date the metadata document was created or modified.</li> <li><code>hierarchyLevel</code>: Type <code>NonEmptyStringType</code>.</li> <li><code>citation</code>: Type <code>citationType</code>. Description: A single citation for use when citing the dataset.</li> <li><code>bibliography</code>: Description: A list of citations that form a bibliography on literature related / used in the dataset.</li> <li><code>physical</code>: Multiplicity <code>0..∞</code>. Description: A container element for all of the elements that let you describe the internal/external characteristics and...</li> <li><code>resourceLogoUrl</code>: Type <code>xs:anyURI</code>. Description: URL of the logo associated with a resource.</li> <li><code>collection</code>: Multiplicity <code>0..∞</code>. Description: A container element for other elements associated with collections (e.g., <code>collectionIdentifier</code>, <code>collectionName</code>).</li> <li><code>formationPeriod</code>: Multiplicity <code>0..∞</code>. Type <code>NonEmptyStringType</code>. Description: Text description of the time period during which the collection was assembled e.g. "Victorian", or "1922 - 1932", or...</li> <li><code>specimenPreservationMethod</code>: Multiplicity <code>0..∞</code>. Type <code>NonEmptyStringType</code>. Description: Picklist keyword indicating the process or technique used to prevent physical deterioration of non-living collections....</li> <li><code>livingTimePeriod</code>: Multiplicity <code>0..∞</code>. Type <code>NonEmptyStringType</code>. Description: Time period during which biological material was alive. (for palaeontological collections)</li> <li><code>jgtiCuratorialUnit</code>: Multiplicity <code>0..∞</code>. Description: A quantitative descriptor (number of specimens, samples or batches).</li> <li><code>dc:replaces</code>: Description: Pointer to previous version of the document.</li> </ul>
Properties	content: complex

Model	dateStamp , hierarchyLevel{0,1} , citation{0,1} , bibliography{0,1} , physical* , resourceLogoUrl{0,1} , collection* , formationPeriod* , specimenPreservationMethod* , livingTimePeriod* , jgtiCuratorialUnit* , replaces{0,1}
Children	bibliography, citation, collection, dateStamp, formationPeriod, hierarchyLevel, jgtiCuratorialUnit, livingTimePeriod, physical, replaces, resourceLogoUrl, specimenPreservationMethod
Instance	<pre> &lt;gbif xmlns:dc="http://purl.org/dc/terms/"&gt;   &lt;dateStamp&gt;{1,1}&lt;/dateStamp&gt;   &lt;hierarchyLevel&gt;{0,1}&lt;/hierarchyLevel&gt;   &lt;citation identifiers=""&gt;{0,1}&lt;/citation&gt;   &lt;bibliography&gt;{0,1}&lt;/bibliography&gt;   &lt;physical&gt;{0,unbounded}&lt;/physical&gt;   &lt;resourceLogoUrl&gt;{0,1}&lt;/resourceLogoUrl&gt;   &lt;collection&gt;{0,unbounded}&lt;/collection&gt;   &lt;formationPeriod&gt;{0,unbounded}&lt;/formationPeriod&gt;   &lt;specimenPreservationMethod&gt;{0,unbounded}&lt;/specimenPreservationMethod&gt;   &lt;livingTimePeriod&gt;{0,unbounded}&lt;/livingTimePeriod&gt;   &lt;jgtiCuratorialUnit&gt;{0,unbounded}&lt;/jgtiCuratorialUnit&gt;   &lt;dc:replaces&gt;{0,1}&lt;/dc:replaces&gt; &lt;/gbif&gt;</pre>
Source	<pre> &lt;xs:element name="gbif"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A block of additional metadata used for some special GBIF purposes and crossmapping to other schemas like the TDWG Natural Collection Data (NCD) schema&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;!-- additional general metadata --&gt;       &lt;xs:element name="dateStamp" type="xs:dateTime"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The date the metadata document was created or modified.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="hierarchyLevel" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation/&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="citation" type="citationType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A single citation for use when citing the dataset&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="bibliography" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A list of citations that form a bibliography on literature related / used in the dataset&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element name="citation" type="citationType" maxOccurs="unbounded"/&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;       &lt;xs:element name="physical" maxOccurs="unbounded" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A container element for all of the elements that let you describe the internal/external characteristics and distribution of a data object (e.g., dataObject, dataFormat, distribution) .&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:complexType&gt;           &lt;xs:sequence&gt;             &lt;xs:element ref="objectName"/&gt;             &lt;xs:element ref="characterEncoding" minOccurs="0"/&gt;             &lt;xs:element ref="dataFormat"/&gt;             &lt;xs:element ref="distribution"/&gt;           &lt;/xs:sequence&gt;         &lt;/xs:complexType&gt;       &lt;/xs:element&gt;       &lt;xs:element name="resourceLogoUrl" type="xs:anyURI" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;URL of the logo associated with a resource&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;!-- additional NCD collection metadata --&gt;     &lt;xs:element name="collection" minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;A container element for other elements associated with collections (e.g., collectionIdentifier, collectionName).&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>

```

<xs:sequence>
    <xs:element name="parentCollectionIdentifier" type="NonEmptyStringType" minOccurs="0">
        <xs:annotation>
            <xs:documentation>Identifier for the parent collection for this sub-collection. Enables a hierarchy of collections and sub collections to be built.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="collectionIdentifier" type="NonEmptyStringType" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The URI (LSID or URL) of the collection. In RDF, used as URI of the collection resource.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="collectionName" type="NonEmptyStringType">
        <xs:annotation>
            <xs:documentation>Official name of the Collection in the local language</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:sequence>
        <xs:complexType>
            </xs:element>
            <xs:element name="formationPeriod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>Text description of the time period during which the collection was assembled e.g. "Victorian", or "1922 - 1932", or "c. 1750".</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="specimenPreservationMethod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>Picklist keyword indicating the process or technique used to prevent physical deterioration of non-living collections. Expected to contain a value from the GBIF Specimen Preservation Method vocabulary</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="livingTimePeriod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>Time period during which biological material was alive. (for palaeontological collections)</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element ref="jgtiCuratorialUnit" maxOccurs="unbounded" minOccurs="0"/>
            <xs:element ref="dc:replaces" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Pointer to previous version of the document</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

### **Element additionalMetadata / metadata / gbif / dateStamp**

Namespace	No namespace
Annotations	The date the metadata document was created or modified.
Diagram	<pre> classDiagram     dateStamp {         Type xs:dateTime     }     dateStamp --o xs:dateTime     note over dateStamp: The date the metadata document was created or modified.     note over xs:dateTime: Built-in primitive type. The dateTime datatype represents a specific instant of time. </pre>
Type	xs:dateTime
Properties	content: simple
Source	<pre> &lt;xs:element name="dateStamp" type="xs:dateTime"&gt;     &lt;xs:annotation&gt;         &lt;xs:documentation&gt;The date the metadata document was created or modified.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>

### **Element additionalMetadata / metadata / gbif / hierarchyLevel**

Namespace	No namespace
Annotations	

Diagram	A UML class diagram showing the 'hierarchyLevel' element. It has a 'Type' association with 'NonEmptyStringType'. A callout box indicates that NonEmptyStringType specifies a content pattern for all elements.				
Type	NonEmptyStringType				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td><code>[ \s]*[ \s][\s\S]*</code></td> </tr> </table>	minLength	1	pattern	<code>[ \s]*[ \s][\s\S]*</code>
minLength	1				
pattern	<code>[ \s]*[ \s][\s\S]*</code>				
Source	<pre>&lt;xs:element name="hierarchyLevel" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation/&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>				

### Element additionalMetadata / metadata / gbif / citation

Namespace	No namespace									
Annotations	A single citation for use when citing the dataset									
Diagram	A UML class diagram showing the 'citation' element. It has a 'Type' association with 'citationType'. A callout box indicates that citation is a single citation for use when citing the dataset. Another callout box shows the 'citationType' class with its base type 'NonEmptyStringType'. An attribute 'identifier' is shown with type 'xs:string' and a callout box indicating it is a URI, DOI or other persistent identifier.									
Type	citationType									
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:string</li> <li>• NonEmptyStringType</li> <li>• citationType</li> </ul>									
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	complex	minOccurs:	0					
content:	complex									
minOccurs:	0									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>identifier</td> <td>xs:string</td> <td>optional</td> </tr> <tr> <td colspan="3">A URI, DOI or other persistent identifier for the citation</td></tr> </tbody> </table>	QName	Type	Use	identifier	xs:string	optional	A URI, DOI or other persistent identifier for the citation		
QName	Type	Use								
identifier	xs:string	optional								
A URI, DOI or other persistent identifier for the citation										
Source	<pre>&lt;xs:element name="citation" type="citationType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A single citation for use when citing the dataset&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>									

### Element additionalMetadata / metadata / gbif / bibliography

Namespace	No namespace
Annotations	A list of citations that form a bibliography on literature related / used in the dataset
Diagram	A UML class diagram showing the 'bibliography' element. It has a '1..>' multiplicity association with the 'citation' element, which has a 'Type' association with 'citationType'. A callout box indicates that bibliography is a list of citations forming a bibliography on literature related / used in the dataset.

Properties	content: complex minOccurs: 0
Model	citation+
Children	citation
Instance	<bibliography> <citation identifier="">{1,unbounded}</citation> </bibliography>

Source

```
<x:element name="bibliography" minOccurs="0">
  <x:annotation>
    <x:documentation>A list of citations that form a bibliography on literature related / used in the dataset</x:documentation>
  </x:annotation>
  <x:complexType>
    <x:sequence>
      <x:element name="citation" type="citationType" maxOccurs="unbounded"/>
    </x:sequence>
  </x:complexType>
</x:element>
```

### Element additionalMetadata / metadata / gbif / bibliography / citation

Namespace	No namespace									
Diagram	<p>A single literature citation with an optional identifier</p>									
Type	citationType									
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:string</li> <li>• NonEmptyStringType</li> <li>• citationType</li> </ul>									
Properties	content: complex maxOccurs: unbounded									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>identifier</td> <td>xs:string</td> <td>optional</td> </tr> <tr> <td></td> <td></td> <td>A URI, DOI or other persistent identifier for the citation</td> </tr> </tbody> </table>	QName	Type	Use	identifier	xs:string	optional			A URI, DOI or other persistent identifier for the citation
QName	Type	Use								
identifier	xs:string	optional								
		A URI, DOI or other persistent identifier for the citation								
Source	<x:element name="citation" type="citationType" maxOccurs="unbounded"/>									

### Element additionalMetadata / metadata / gbif / physical

Namespace	No namespace
Annotations	A container element for all of the elements that let you describe the internal/external characteristics and distribution of a data object (e.g., dataObject, dataFormat, distribution) .

Diagram	<pre> classDiagram     class physical {         objectName         characterEncoding         dataFormat         distribution     }     objectName &lt; -- NonEmptyStringType     characterEncoding &lt; -- NonEmptyStringType     dataFormat &lt; -- NonEmptyStringType     distribution &lt; -- NonEmptyStringType     note over physical: A container element for all of the elements that let you describe the internal/external characteristics and...   </pre>
Properties	<p>content: complex</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Model	objectName , characterEncoding{0,1} , dataFormat , distribution
Children	characterEncoding, dataFormat, distribution, objectName
Instance	<pre> &lt;physical&gt;   &lt;objectName&gt;{1,1}&lt;/objectName&gt;   &lt;characterEncoding&gt;{0,1}&lt;/characterEncoding&gt;   &lt;dataFormat&gt;{1,1}&lt;/dataFormat&gt;   &lt;distribution scope=""&gt;{1,1}&lt;/distribution&gt; &lt;/physical&gt;   </pre>
Source	<pre> &lt;xs:element name="physical" maxOccurs="unbounded" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A container element for all of the elements that let you describe the internal/external characteristics and distribution of a data object (e.g., dataObject, dataFormat, distribution) .&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="objectName"/&gt;       &lt;xs:element ref="characterEncoding" minOccurs="0"/&gt;       &lt;xs:element ref="dataFormat"/&gt;       &lt;xs:element ref="distribution"/&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>

## Element **objectName**

Namespace	No namespace
Annotations	The name of the data object. This often is the filename of a file in a file system or that is accessible on the network.
Diagram	<pre> classDiagram     class objectName {         Type NonEmptyStringType     }     note over objectName: The name of the data object. This often is the filename of a file in a file system or that is accessible on the network.     note over NonEmptyStringType: Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...   </pre>
Type	NonEmptyStringType
Properties	<p>content: simple</p>
Facets	<p>minLength 1</p>
	pattern $[\s]*[\S][\s\S]*$
Used by	Element additionalMetadata/metadata/gbif/physical
Source	<pre> &lt;xs:element name="objectName" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;   </pre>

```

<xs:documentation>The name of the data object. This often is the filename of a file in a file system or that is accessible on the network.</xs:documentation>
</xs:annotation>
</xs:element>

```

## Element characterEncoding

Namespace	No namespace				
Annotations	This element contains the name of the character encoding. This is typically ASCII or UTF-8, or one of the other common encodings.				
Diagram	<p>The diagram shows a class named 'characterEncoding' with a multiplicity of 0..1. It has a directed association labeled 'Type' pointing to a class named 'NonEmptyStringType' with a multiplicity of 1..1. A callout box points to the 'characterEncoding' class with the text: 'This element contains the name of the character encoding. This is typically ASCII or UTF-8, or one of the other common...'. Another callout box points to the 'NonEmptyStringType' class with the text: 'Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...'.</p>				
Type	NonEmptyStringType				
Properties	content: simple				
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ]*[ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ]*[ \s \S ]*
minLength	1				
pattern	[ \s ]*[ \S ]*[ \s \S ]*				
Used by	Element additionalMetadata/metadata/gbif/physical				
Source	<pre> &lt;xs:element name="characterEncoding" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element contains the name of the character encoding. This is typically ASCII or UTF-8, or one of the other common encodings.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>				

## Element dataFormat

Namespace	No namespace
Annotations	This is a container element for other elements which describe the internal physical characteristics of the data object.
Diagram	<p>The diagram shows a class named 'dataFormat' with a multiplicity of 0..1. It has two directed associations labeled 'textFormat' and 'externallyDefinedFormat', both with a multiplicity of 0..1. A callout box points to the 'dataFormat' class with the text: 'This is a container element for other elements which describe the internal physical characteristics of the data object.'. Another callout box points to the 'textFormat' class with the text: 'Description of a text formatted object. The description includes detailed parsing instructions for extracting...'. A third callout box points to the 'externallyDefinedFormat' class with the text: 'Information about a non-text or proprietary formatted object.'.</p>
Properties	content: complex
Used by	Element additionalMetadata/metadata/gbif/physical
Model	textFormat   externallyDefinedFormat
Children	externallyDefinedFormat, textFormat
Instance	<pre> &lt;dataFormat&gt;   &lt;textFormat&gt;{1,1}&lt;/textFormat&gt;   &lt;externallyDefinedFormat&gt;{1,1}&lt;/externallyDefinedFormat&gt; &lt;/dataFormat&gt; </pre>
Source	<pre> &lt;xs:element name="dataFormat"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is a container element for other elements which describe the internal physical characteristics of the data object.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:choice&gt;       &lt;xs:element name="textFormat"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Description of a text formatted object. The description includes detailed parsing instructions for extracting attributes from the bytestream for simple delimited file formats (e.g., CSV), fixed format files that use fixed columns for attribute locations, and mixtures of the two. It also supports records that span multiple lines.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:choice&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

```

<xs:element name="numHeaderLines" type="xs:int" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Number of header lines preceding data. Lines are determined by the physicalLineDelimiter, or if it is absent, by the recordDelimiter. This value indicated the number of header lines that should be skipped before starting to parse the data.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="numFooterLines" type="xs:int" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Number of footer lines following data. Lines are determined by the physicalLineDelimiter, or if it is absent, by the recordDelimiter. This value indicated the number of footer lines that should be skipped after parsing the data. If this value is omitted, parsers should assume the data continues to the end of the data stream.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="recordDelimiter" type="xs:string" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>This element specifies the record delimiter character when the format is text. The record delimiter is usually a linefeed (\n) on UNIX, a carriage return (\r) on MacOS, or both (\r\n) on Windows/DOS. Multiline records are usually delimited with two line ending characters, for example on UNIX it would be two linefeed characters (\n\n). As record delimiters are often non-printing characters, one can use either the special value "\n" to represent a linefeed (ASCII 0x0a) and "\r" to represent a carriage return (ASCII 0x0d). Alternatively, one can use the hex value to represent character values (e.g., 0x0a).</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="physicalLineDelimiter" type="xs:string" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>This element specifies the physical line delimiter character when the format is text. The line delimiter is usually a linefeed (\n) on UNIX, a carriage return (\r) on MacOS, or both (\r\n) on Windows/DOS. Multiline records are usually delimited with two line ending characters, for example on UNIX it would be two linefeed characters (\n\n). As line delimiters are often non-printing characters, one can use either the special value "\n" to represent a linefeed (ASCII 0x0a) and "\r" to represent a carriage return (ASCII 0x0d). Alternatively, one can use the hex value to represent character values (e.g., 0x0a). If this value is not provided, processors should assume that the physical line delimiter is the same as the record delimiter.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="numPhysicalLinesPerRecord" type="xs:unsignedInt" minOccurs="0">
    <xs:annotation>
        <xs:documentation>A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, it is necessary to know the number of lines per record in order to correctly read them. If this value is not provided, processors should assume that records are wholly contained on one physical line. If the value is greater than 1, then processors should examine the lineNumber field for each attribute to determine which line of the record contains the information.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="maxRecordLength" type="xs:unsignedLong" minOccurs="0">
    <xs:annotation>
        <xs:documentation>The maximum number of characters in any record in the physical file. For delimited files, the record length varies and this is not particularly useful. However, for fixed format files that do not contain record delimiters, this field is critical to tell processors when one record stops and another begins.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="attributeOrientation">
    <xs:annotation>
        <xs:documentation>Specifies whether the attributes described in the physical stream are found in columns or rows. The valid values are column or row. If set to 'column', then the attributes are in columns. If set to 'row', then the attributes are in rows. Row orientation is rare, but some systems such as SPlus and R utilize it. For example, some data with column orientation: DATE PLOT SPECIES 2002-01-15 hfr5 acer rubrum 2002-01-15 hfr5 acer xxxx The same data in a rowMajor table: DATE 2002-01-15 PLOT hfr5 SPECIES acer rubrum acer xxxx</xs:documentation>
    </xs:annotation>
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:enumeration value="column"/>
        <xs:enumeration value="row"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:choice>
    <xs:element name="simpleDelimited">
        <xs:annotation>
            <xs:documentation>A simple delimited format that uses one of a series of delimiters to indicate the ends of fields in the data stream. More complex formats such as fixed format or mixed delimited and fixed formats can be described using the "complex" element.</xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:choice>
<xs:complexType>

```

```

<xs:sequence>
    <xs:element name="fieldDelimiter" type="xs:string" maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation>This element specifies a character to be used in the object for indicating the ending column for an attribute. The delimiter character itself is not part of the attribute value, but rather is present in the column following the last character of the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote characters can be provided as their hex values, and for tab characters by its ASCII string "\t". Processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="collapseDelimiters" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or multiple delimiters. An example is when a space delimiter is used; often there may be several repeated spaces that should be treated as a single delimiter, but not always. The valid values are yes or no. If it is set to yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive delimiters will be treated as separate delimiters, by default.</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="yes"/>
                <xs:enumeration value="no"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="quoteCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation>This element specifies a character to be used in the object for quoting values so that field delimiters can be used within the value. This basically allows delimiter "escaping". The quoteCharacter is typically a " or '. When a processor encounters a quote character, it should not interpret any following characters as a delimiter until a matching quote character has been encountered (i.e., quotes come in pairs). It is an error to not provide a closing quote before the record ends. Non-printable quote characters can be provided as their hex values.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="literalCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation>This element specifies a character to be used for escaping special character values so that they are treated as literal values. This allows "escaping" for special characters like quotes, commas, and spaces when they are intended to be used in an attribute value rather than being intended as a delimiter. The literalCharacter is typically a \.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:sequence>
        <xs:complexType>
            <xs:element name="complex">
                <xs:annotation>
                    <xs:documentation>A complex text format that can describe delimited fields, fixed width fields, and mixtures of the two. This supports multiline records (where one record is distributed across multiple physical lines). When using the complex format, the number of textFixed and textDelimited elements should exactly equal the number of attributes that have been described for the entity, and the order of the textFixed and textDelimited elements should correspond to the order of the attributes as described in the entity. Thus, for a delimited file with fourteen attributes, one should provide exactly fourteen textDelimited elements.</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:complexType>
        <xs:choice maxOccurs="unbounded">
            <xs:element name="textFixed">
                <xs:annotation>
                    <xs:documentation>Describes the physical format of data sequences that use a fixed number of characters in a specified position in the stream to locate attribute values. This method is common in sensor-derived data and in legacy database systems. To parse it, one must know the number of characters for each attribute and the starting column and line to begin reading the value.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="fieldWidth" type="xs:unsignedLong">
                        <xs:annotation>

```

```

<xs:documentation>Fixed width fields have a set length, thus the end
of the field can always be determined by adding the fieldWidth to the starting column number.</
xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="lineNumber" type="xs:unsignedLong" minOccurs="0">
    <xs:annotation>
        <xs:documentation>The line on which the data field is found, when the
data record is written over more than one physical line in the file. A single logical data record
may be written over several physical lines in a file, with no special marker to indicate the end of
a record. In such cases, the relative location of a data field must be indicated by both relative
row and column number. The lineNumber should never greater than the number of physical lines per
record.</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="fieldStartColumn" type="xs:long" minOccurs="0">
        <xs:annotation>
            <xs:documentation>The starting column number for a fixed format
attribute. Fixed width fields have a set length, thus the end of the field can always be determined
by adding the fieldWidth to the starting column number. If the starting column is not provided,
processors should assume that the field starts in the column following the previous field if the
previous field was fixed, or in the column following the delimiter from the previous field if the
previous field was delimited.</xs:documentation>
            </xs:annotation>
        </xs:element>
        </xs:sequence>
    </xs:complexType>
    </xs:elements>
    <xs:element name="textDelimited">
        <xs:annotation>
            <xs:documentation>Describes the physical format of data sequences that use
delimiters in the stream to locate attribute values. This method is common in data exported from
spreadsheets and database systems. To parse it, one must know the character that indicates the end
of each attribute and the line to begin reading the value.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="fieldDelimiter" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>This element specifies a character to be used in the
object for indicating the ending column for an attribute. The delimiter character itself is not
part of the attribute value, but rather is present in the column following the last character of
the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time
the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted
string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote
characters can be provided as their hex values, and for tab characters by its ASCII string "\t".
Processors should assume that the field starts in the column following the previous field if the
previous field was fixed, or in the column following the delimiter from the previous field if the
previous field was delimited.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="collapseDelimiters" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>The collapseDelimiters element specifies whether
sequential delimiters should be treated as a single delimiter or multiple delimiters. An example
is when a space delimiter is used; often there may be several repeated spaces that should be
treated as a single delimiter, but not always. The valid values are yes or no. If it is set to
yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive
delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive
delimiters will be treated as separate delimiters, by default.</xs:documentation>
                    </xs:annotation>
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="yes"/>
                        <xs:enumeration value="no"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="lineNumber" type="xs:unsignedLong" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>A single logical data record may be written over
several physical lines in a file, with no special marker to indicate the end of a record. In such
cases, the relative location of a data field must be indicated by both relative row and column
number. The lineNumber should never be greater than the number of physical lines per record. When
parsing the first field on a physical line as a delimited field, they should assume that the field
data starts in the first column. Otherwise, follow the rules indicated under fieldDelimiter.</
xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="quoteCharacter" type="NonEmptyStringType" minOccurs="0"
maxOccurs="unbounded">
                <xs:annotation>

```

```

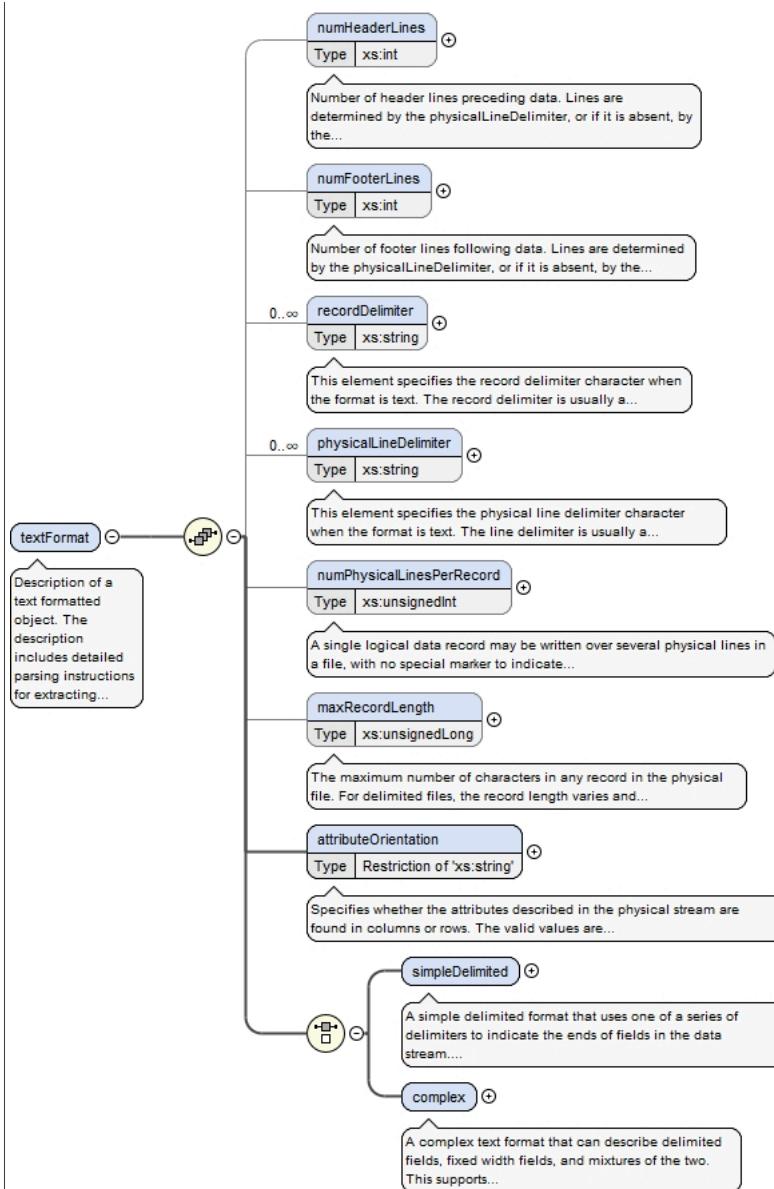
<xs:documentation>This element specifies a character to be used in the object for quoting values so that field delimiters can be used within the value. This basically allows delimiter "escaping". The quoteCharacter is typically a " or '. When a processor encounters a quote character, it should not interpret any following characters as a delimiter until a matching quote character has been encountered (i.e., quotes come in pairs). It is an error to not provide a closing quote before the record ends. Non-printable quote characters can be provided as their hex values.</xs:documentation>
    </xs:annotation>
    </xs:element>
    <xs:element name="literalCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation>This element specifies a character to be used for escaping special character values so that they are treated as literal values. This allows "escaping" for special characters like quotes, commas, and spaces when they are intended to be used in an attribute value rather than being intended as a delimiter. The literalCharacter is typically a \.</xs:documentation>
                </xs:annotation>
                </xs:element>
                </xs:sequence>
                </xs:complexType>
            </xs:elements>
            </xs:choice>
            </xs:complexType>
            </xs:element>
            </xs:choice>
            </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="externallyDefinedFormat">
            <xs:annotation>
                <xs:documentation>Information about a non-text or proprietary formatted object.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="formatName" type="xs:string">
                        <xs:annotation>
                            <xs:documentation>Name of the format of the data object, e.g., ESRI Shapefile.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    <xs:element name="formatVersion" type="NonEmptyStringType" minOccurs="0">
                        <xs:annotation>
                            <xs:documentation>Version of the format of the data object</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                    </xs:sequence>
                    </xs:complexType>
                </xs:element>
                </xs:choice>
            </xs:complexType>
        </xs:element>
    </xs:annotation>
</xs:element>

```

## Element dataFormat / textFormat

Namespace	No namespace
Annotations	Description of a text formatted object. The description includes detailed parsing instructions for extracting attributes from the bytestream for simple delimited file formats (e.g., CSV), fixed format files that use fixed columns for attribute locations, and mixtures of the two. It also supports records that span multiple lines.

## Diagram



Properties	content: complex
Model	numHeaderLines{0,1} , numFooterLines{0,1} , recordDelimiter* , physicalLineDelimiter* , numPhysicalLinesPerRecord{0,1} , maxRecordLength{0,1} , attributeOrientation , (simpleDelimited   complex)
Children	attributeOrientation, complex, maxRecordLength, numFooterLines, numHeaderLines, numPhysicalLinesPerRecord, physicalLineDelimiter, recordDelimiter, simpleDelimited
Instance	<pre> &lt;textFormat&gt;     &lt;numHeaderLines&gt;{0,1}&lt;/numHeaderLines&gt;     &lt;numFooterLines&gt;{0,1}&lt;/numFooterLines&gt;     &lt;recordDelimiter&gt;{0,unbounded}&lt;/recordDelimiter&gt;     &lt;physicalLineDelimiter&gt;{0,unbounded}&lt;/physicalLineDelimiter&gt;     &lt;numPhysicalLinesPerRecord&gt;{0,1}&lt;/numPhysicalLinesPerRecord&gt;     &lt;maxRecordLength&gt;{0,1}&lt;/maxRecordLength&gt;     &lt;attributeOrientation&gt;{1,1}&lt;/attributeOrientation&gt;     &lt;simpleDelimited&gt;{1,1}&lt;/simpleDelimited&gt;     &lt;complex&gt;{1,1}&lt;/complex&gt; &lt;/textFormat&gt; </pre>
Source	<pre> &lt;xss:element name="textFormat"&gt;     &lt;xss:annotation&gt;         &lt;xss:documentation&gt;Description of a text formatted object. The description includes detailed parsing instructions for extracting attributes from the bytestream for simple delimited file formats (e.g., CSV), fixed format files that use fixed columns for attribute locations, and mixtures of the two. It also supports records that span multiple lines.&lt;/xss:documentation&gt;     &lt;/xss:annotation&gt;     &lt;xss:complexType&gt;         &lt;xss:sequence&gt; </pre>

```

<xs:element name="numHeaderLines" type="xs:int" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Number of header lines preceding data. Lines are determined by the physicalLineDelimiter, or if it is absent, by the recordDelimiter. This value indicated the number of header lines that should be skipped before starting to parse the data.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="numFooterLines" type="xs:int" minOccurs="0">
    <xs:annotation>
        <xs:documentation>Number of footer lines following data. Lines are determined by the physicalLineDelimiter, or if it is absent, by the recordDelimiter. This value indicated the number of footer lines that should be skipped after parsing the data. If this value is omitted, parsers should assume the data continues to the end of the data stream.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="recordDelimiter" type="xs:string" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>This element specifies the record delimiter character when the format is text. The record delimiter is usually a linefeed (\n) on UNIX, a carriage return (\r) on MacOS, or both (\r\n) on Windows/DOS. Multiline records are usually delimited with two line ending characters, for example on UNIX it would be two linefeed characters (\n\r). As record delimiters are often non-printing characters, one can use either the special value "\n" to represent a linefeed (ASCII 0x0a) and "\r" to represent a carriage return (ASCII 0x0d). Alternatively, one can use the hex value to represent character values (e.g., 0x0a).</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="physicalLineDelimiter" type="xs:string" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>This element specifies the physical line delimiter character when the format is text. The line delimiter is usually a linefeed (\n) on UNIX, a carriage return (\r) on MacOS, or both (\r\n) on Windows/DOS. Multiline records are usually delimited with two line ending characters, for example on UNIX it would be two linefeed characters (\n\r). As line delimiters are often non-printing characters, one can use either the special value "\n" to represent a linefeed (ASCII 0x0a) and "\r" to represent a carriage return (ASCII 0x0d). Alternatively, one can use the hex value to represent character values (e.g., 0x0a). If this value is not provided, processors should assume that the physical line delimiter is the same as the record delimiter.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="numPhysicalLinesPerRecord" type="xs:unsignedInt" minOccurs="0">
    <xs:annotation>
        <xs:documentation>A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, it is necessary to know the number of lines per record in order to correctly read them. If this value is not provided, processors should assume that records are wholly contained on one physical line. If the value is greater than 1, then processors should examine the lineNumber field for each attribute to determine which line of the record contains the information.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="maxRecordLength" type="xs:unsignedLong" minOccurs="0">
    <xs:annotation>
        <xs:documentation>The maximum number of characters in any record in the physical file. For delimited files, the record length varies and this is not particularly useful. However, for fixed format files that do not contain record delimiters, this field is critical to tell processors when one record stops and another begins.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="attributeOrientation">
    <xs:annotation>
        <xs:documentation>Specifies whether the attributes described in the physical stream are found in columns or rows. The valid values are column or row. If set to 'column', then the attributes are in columns. If set to 'row', then the attributes are in rows. Row orientation is rare, but some systems such as SPlus and R utilize it. For example, some data with column orientation: DATE PLOT SPECIES 2002-01-15 hfr5 acer rubrum 2002-01-15 hfr5 acer xxxx The same data in a rowMajor table: DATE 2002-01-15 PLOT hfr5 SPECIES acer rubrum acer xxxx</xs:documentation>
    </xs:annotation>
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:enumeration value="column"/>
        <xs:enumeration value="row"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:choice>
    <xs:element name="simpleDelimited">
        <xs:annotation>
            <xs:documentation>A simple delimited format that uses one of a series of delimiters to indicate the ends of fields in the data stream. More complex formats such as fixed format or mixed delimited and fixed formats can be described using the "complex" element.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="fieldDelimiter" type="xs:string" maxOccurs="unbounded">

```

```

<xs:annotation>
    <xs:documentation>This element specifies a character to be used in the object for indicating the ending column for an attribute. The delimiter character itself is not part of the attribute value, but rather is present in the column following the last character of the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote characters can be provided as their hex values, and for tab characters by its ASCII string "\t". Processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="collapseDelimiters" minOccurs="0">
    <xs:annotation>
        <xs:documentation>The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or multiple delimiters. An example is when a space delimiter is used; often there may be several repeated spaces that should be treated as a single delimiter, but not always. The valid values are yes or no. If it is set to yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive delimiters will be treated as separate delimiters, by default.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="yes"/>
            <xs:enumeration value="no"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="quoteCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>This element specifies a character to be used in the object for quoting values so that field delimiters can be used within the value. This basically allows delimiter "escaping". The quoteCharacter is typically a " or '. When a processor encounters a quote character, it should not interpret any following characters as a delimiter until a matching quote character has been encountered (i.e., quotes come in pairs). It is an error to not provide a closing quote before the record ends. Non-printable quote characters can be provided as their hex values.</xs:documentation>
    </xs:annotation>
    <xs:element name="literalCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
            <xs:documentation>This element specifies a character to be used for escaping special character values so that they are treated as literal values. This allows "escaping" for special characters like quotes, commas, and spaces when they are intended to be used in an attribute value rather than being intended as a delimiter. The literalCharacter is typically a \.</xs:documentation>
        </xs:annotation>
        <xs:element>
            <xs:sequence>
                <xs:complexType>
                    <xs:element name="complex">
                        <xs:annotation>
                            <xs:documentation>A complex text format that can describe delimited fields, fixed width fields, and mixtures of the two. This supports multiline records (where one record is distributed across multiple physical lines). When using the complex format, the number of textFixed and textDelimited elements should exactly equal the number of attributes that have been described for the entity, and the order of the textFixed and textDelimited elements should correspond to the order of the attributes as described in the entity. Thus, for a delimited file with fourteen attributes, one should provide exactly fourteen textDelimited elements.</xs:documentation>
                        </xs:annotation>
                    </xs:element>
                <xs:choice maxOccurs="unbounded">
                    <xs:element name="textFixed">
                        <xs:annotation>
                            <xs:documentation>Describes the physical format of data sequences that use a fixed number of characters in a specified position in the stream to locate attribute values. This method is common in sensor-derived data and in legacy database systems. To parse it, one must know the number of characters for each attribute and the starting column and line to begin reading the value.</xs:documentation>
                        </xs:annotation>
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="fieldWidth" type="xs:unsignedLong">
                                <xs:annotation>
                                    <xs:documentation>Fixed width fields have a set length, thus the end of the field can always be determined by adding the fieldWidth to the starting column number.</xs:documentation>
                                </xs:annotation>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:choice>
            </xs:sequence>
        </xs:element>
    </xs:annotation>
</xs:element>

```

```

        </xs:element>
        <xs:element name="lineNumber" type="xs:unsignedLong" minOccurs="0">
            <xs:annotation>
                <xs:documentation>The line on which the data field is found, when the data record is written over more than one physical line in the file. A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, the relative location of a data field must be indicated by both relative row and column number. The lineNumber should never greater than the number of physical lines per record.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="fieldStartColumn" type="xs:long" minOccurs="0">
            <xs:annotation>
                <xs:documentation>The starting column number for a fixed format attribute. Fixed width fields have a set length, thus the end of the field can always be determined by adding the fieldWidth to the starting column number. If the starting column is not provided, processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.</xs:documentation>
            </xs:annotation>
        </xs:element>
        </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="textDelimited">
        <xs:annotation>
            <xs:documentation>Describes the physical format of data sequences that use delimiters in the stream to locate attribute values. This method is common in data exported from spreadsheets and database systems. To parse it, one must know the character that indicates the end of each attribute and the line to begin reading the value.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="fieldDelimiter" type="xs:string">
                    <xs:annotation>
                        <xs:documentation>This element specifies a character to be used in the object for indicating the ending column for an attribute. The delimiter character itself is not part of the attribute value, but rather is present in the column following the last character of the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote characters can be provided as their hex values, and for tab characters by its ASCII string "\t". Processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:element name="collapseDelimiters" minOccurs="0">
                    <xs:annotation>
                        <xs:documentation>The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or multiple delimiters. An example is when a space delimiter is used; often there may be several repeated spaces that should be treated as a single delimiter, but not always. The valid values are yes or no. If it is set to yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive delimiters will be treated as separate delimiters, by default.</xs:documentation>
                    </xs:annotation>
                </xs:element>
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="yes"/>
                        <xs:enumeration value="no"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:sequence>
        </xs:complexType>
        <xs:element name="lineNumber" type="xs:unsignedLong" minOccurs="0">
            <xs:annotation>
                <xs:documentation>A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, the relative location of a data field must be indicated by both relative row and column number. The lineNumber should never be greater than the number of physical lines per record. When parsing the first field on a physical line as a delimited field, they should assume that the field data starts in the first column. Otherwise, follow the rules indicated under fieldDelimiter.</xs:documentation>
            </xs:annotation>
        </xs:element>
        <xs:element name="quoteCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>This element specifies a character to be used in the object for quoting values so that field delimiters can be used within the value. This basically allows delimiter "escaping". The quoteCharacter is typically a " or '. When a processor encounters a quote character, it should not interpret any following characters as a delimiter until a matching quote character has been encountered (i.e., quotes come in pairs). It is an error to not provide a
            </xs:annotation>
        </xs:element>
    </xs:sequence>
    </xs:complexType>

```

```

closing quote before the record ends. Non-printable quote characters can be provided as their hex
values.</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="literalCharacter" type="NonEmptyStringType" minOccurs="0"
maxOccurs="unbounded">
    <xs:annotation>
        <xs:documentation>This element specifies a character to be used for escaping
special character values so that they are treated as literal values. This allows "escaping"
for special characters like quotes, commas, and spaces when they are intended to be used in an
attribute value rather than being intended as a delimiter. The literalCharacter is typically a \.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:sequence>
</xs:complexType>
</xs:element>

```

### Element dataFormat / textFormat / numHeaderLines

Namespace	No namespace				
Annotations	Number of header lines preceding data. Lines are determined by the physicalLineDelimiter, or if it is absent, by the recordDelimiter. This value indicated the number of header lines that should be skipped before starting to parse the data.				
Diagram	<p>Diagram illustrating the type xs:int for numHeaderLines:</p> <pre> classDiagram     class numHeaderLines {         &lt;&lt;xs:int&gt;&gt;         Type xs:int     }     numHeaderLines --o xs:int     xs:int     note over numHeaderLines: Number of header lines preceding data. Lines are determined by the physicalLineDelimiter, or if it is absent, by the recordDelimiter.     note over xs:int: Built-in derived type. The int datatype is derived from long by setting the value of maxInclusive to be 2147483647 and...   </pre>				
Type	xs:int				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<pre> &lt;xs:element name="numHeaderLines" type="xs:int" minOccurs="0"&gt;     &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Number of header lines preceding data. Lines are determined by the physicalLineDelimiter, or if it is absent, by the recordDelimiter. This value indicated the number of header lines that should be skipped before starting to parse the data.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>				

### Element dataFormat / textFormat / numFooterLines

Namespace	No namespace
Annotations	Number of footer lines following data. Lines are determined by the physicalLineDelimiter, or if it is absent, by the recordDelimiter. This value indicated the number of footer lines that should be skipped after parsing the data. If this value is omitted, parsers should assume the data continues to the end of the data stream.
Diagram	<p>Diagram illustrating the type xs:int for numFooterLines:</p> <pre> classDiagram     class numFooterLines {         &lt;&lt;xs:int&gt;&gt;         Type xs:int     }     numFooterLines --o xs:int     xs:int     note over numFooterLines: Number of footer lines following data. Lines are determined by the physicalLineDelimiter, or if it is absent, by the recordDelimiter.     note over xs:int: Built-in derived type. The int datatype is derived from long by setting the value of maxInclusive to be 2147483647 and...   </pre>
Type	xs:int

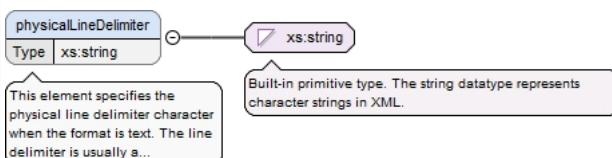
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<pre>&lt;xs:element name="numFooterLines" type="xs:int" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Number of footer lines following data. Lines are determined by the physicalLineDelimiter, or if it is absent, by the recordDelimiter. This value indicated the number of footer lines that should be skipped after parsing the data. If this value is omitted, parsers should assume the data continues to the end of the data stream.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element dataFormat / textFormat / recordDelimiter

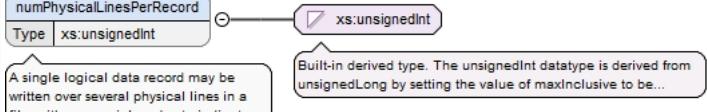
Namespace	No namespace
Annotations	<p>This element specifies the record delimiter character when the format is text. The record delimiter is usually a linefeed (\n) on UNIX, a carriage return (\r) on MacOS, or both (\r\n) on Windows/DOS. Multiline records are usually delimited with two line ending characters, for example on UNIX it would be two linefeed characters (\n\n). As record delimiters are often non-printing characters, one can use either the special value "\n" to represent a linefeed (ASCII 0x0a) and "\r" to represent a carriage return (ASCII 0x0d). Alternatively, one can use the hex value to represent character values (e.g., 0x0a).</p>
Diagram	<pre> classDiagram     class recordDelimiter {         Type xs:string     }     xs:string     recordDelimiter "1" -- "1" xs:string     note over recordDelimiter: This element specifies the record delimiter character when the format is text. The record delimiter is usually a...     note over xs:string: Built-in primitive type. The string datatype represents character strings in XML.   </pre>
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Source	<pre>&lt;xs:element name="recordDelimiter" type="xs:string" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies the record delimiter character when the format is text. The record delimiter is usually a linefeed (\n) on UNIX, a carriage return (\r) on MacOS, or both (\r\n) on Windows/DOS. Multiline records are usually delimited with two line ending characters, for example on UNIX it would be two linefeed characters (\n\n). As record delimiters are often non-printing characters, one can use either the special value "\n" to represent a linefeed (ASCII 0x0a) and "\r" to represent a carriage return (ASCII 0x0d). Alternatively, one can use the hex value to represent character values (e.g., 0x0a).&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element dataFormat / textFormat / physicalLineDelimiter

Namespace	No namespace
Annotations	<p>This element specifies the physical line delimiter character when the format is text. The line delimiter is usually a linefeed (\n) on UNIX, a carriage return (\r) on MacOS, or both (\r\n) on Windows/DOS. Multiline records are usually delimited with two line ending characters, for example on UNIX it would be two linefeed characters (\n\n). As line delimiters are often non-printing characters, one can use either the special value "\n" to represent a linefeed (ASCII 0x0a) and "\r" to represent a carriage return (ASCII 0x0d). Alternatively, one can use the hex value to represent character values (e.g., 0x0a). If this value is not provided, processors should assume that the physical line delimiter is the same as the record delimiter.</p>

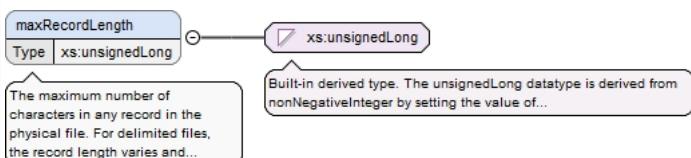
Diagram	
Type	xs:string
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Source	<pre>&lt;xs:element name="physicalLineDelimiter" type="xs:string" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies the physical line delimiter character when the format is text. The line delimiter is usually a linefeed (\n) on UNIX, a carriage return (\r) on MacOS, or both (\r\n) on Windows/DOS. Multiline records are usually delimited with two line ending characters, for example on UNIX it would be two linefeed characters (\n\n). As line delimiters are often non-printing characters, one can use either the special value "\n" to represent a linefeed (ASCII 0x0a) and "\r" to represent a carriage return (ASCII 0x0d). Alternatively, one can use the hex value to represent character values (e.g., 0x0a). If this value is not provided, processors should assume that the physical line delimiter is the same as the record delimiter.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element dataFormat / textFormat / numPhysicalLinesPerRecord

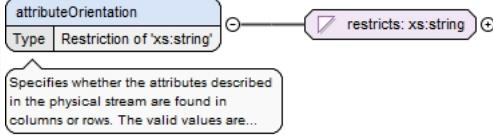
Namespace	No namespace
Annotations	<p>A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, it is necessary to know the number of lines per record in order to correctly read them. If this value is not provided, processors should assume that records are wholly contained on one physical line. If the value is greater than 1, then processors should examine the lineNumber field for each attribute to determine which line of the record contains the information.</p>
Diagram	
Type	xs:unsignedInt
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<pre>&lt;xs:element name="numPhysicalLinesPerRecord" type="xs:unsignedInt" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, it is necessary to know the number of lines per record in order to correctly read them. If this value is not provided, processors should assume that records are wholly contained on one physical line. If the value is greater than 1, then processors should examine the lineNumber field for each attribute to determine which line of the record contains the information.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element dataFormat / textFormat / maxRecordLength

Namespace	No namespace
Annotations	<p>The maximum number of characters in any record in the physical file. For delimited files, the record length varies and this is not particularly useful. However, for fixed format files that do not contain record delimiters, this field is critical to tell processors when one record stops and another begins.</p>

Diagram	
Type	xs:unsignedLong
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<pre>&lt;xs:element name="maxRecordLength" type="xs:unsignedLong" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The maximum number of characters in any record in the physical file. For delimited files, the record length varies and this is not particularly useful. However, for fixed format files that do not contain record delimiters, this field is critical to tell processors when one record stops and another begins.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

**Element dataFormat / textFormat / attributeOrientation**

Namespace	No namespace															
Annotations	<p>Specifies whether the attributes described in the physical stream are found in columns or rows. The valid values are column or row. If set to 'column', then the attributes are in columns. If set to 'row', then the attributes are in rows. Row orientation is rare, but some systems such as SPlus and R utilize it.</p> <p>For example, some data with column orientation:</p> <table> <tr><td>DATE</td><td>PLOT</td><td>SPECIES</td></tr> <tr><td>2002-01-15</td><td>hfr5</td><td>acer rubrum</td></tr> <tr><td>2002-01-15</td><td>hfr5</td><td>acer xxxx</td></tr> </table> <p>The same data in a rowMajor table:</p> <table> <tr><td>DATE</td><td>2002-01-15</td></tr> <tr><td>PLOT</td><td>hfr5</td></tr> <tr><td>SPECIES</td><td>acer rubrum acer xxxx</td></tr> </table>	DATE	PLOT	SPECIES	2002-01-15	hfr5	acer rubrum	2002-01-15	hfr5	acer xxxx	DATE	2002-01-15	PLOT	hfr5	SPECIES	acer rubrum acer xxxx
DATE	PLOT	SPECIES														
2002-01-15	hfr5	acer rubrum														
2002-01-15	hfr5	acer xxxx														
DATE	2002-01-15															
PLOT	hfr5															
SPECIES	acer rubrum acer xxxx															
Diagram																
Type	restriction of xs:string															
Properties	<p>content: simple</p>															
Facets	<p>enumeration column</p> <p>enumeration row</p>															
Source	<pre>&lt;xs:element name="attributeOrientation"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Specifies whether the attributes described in the physical stream are found in columns or rows. The valid values are column or row. If set to 'column', then the attributes are in columns. If set to 'row', then the attributes are in rows. Row orientation is rare, but some systems such as SPlus and R utilize it. For example, some data with column orientation: DATE PLOT SPECIES 2002-01-15 hfr5 acer rubrum 2002-01-15 hfr5 acer xxxx The same data in a rowMajor table: DATE 2002-01-15 PLOT hfr5 SPECIES acer rubrum acer xxxx&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:string"&gt;       &lt;xs:enumeration value="column"/&gt;       &lt;xs:enumeration value="row"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>															

**Element dataFormat / textFormat / simpleDelimited**

Namespace	No namespace
Annotations	A simple delimited format that uses one of a series of delimiters to indicate

	<p>the ends of fields in the data stream. More complex formats such as fixed format or mixed delimited and fixed formats can be described using the "complex" element.</p>
Diagram	<pre> classDiagram     class simpleDelimited {         &lt;&lt;A simple delimited format that uses one of a series of delimiters to indicate the ends of fields in the data stream....&gt;&gt;     }     class fieldDelimiter {         &lt;&lt;This element specifies a character to be used in the object for indicating the ending column for an attribute. The delimiter character itself is not part of the attribute value, but rather is present in the column following the last character of the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote characters can be provided as their hex values, and for tab characters by its ASCII string "\t". Processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.&gt;&gt;     }     class collapseDelimiters {         &lt;&lt;The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or multiple delimiters. An example is when a space delimiter is used; often there may be several repeated spaces that should be treated as a single delimiter, but not always. The valid values are yes or no. If it is set to yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive delimiters will be treated as separate delimiters, by default.&gt;&gt;     }     class quoteCharacter {         &lt;&lt;This element specifies a character to be used in the object for quoting values so that field delimiters can be used...&gt;&gt;     }     class literalCharacter {         &lt;&lt;This element specifies a character to be used for escaping special character values so that they are treated as literal...&gt;&gt;     }      simpleDelimited "0..1" --&gt; "1..oo" simpleDelimited : simpleDelimited     simpleDelimited "*" --&gt; "1..oo" fieldDelimiter : fieldDelimiter     simpleDelimited "*" --&gt; "0..1" collapseDelimiters : collapseDelimiters     simpleDelimited "*" --&gt; "0..oo" quoteCharacter : quoteCharacter     simpleDelimited "*" --&gt; "0..oo" literalCharacter : literalCharacter   </pre>
Properties	content: complex
Model	fieldDelimiter+, collapseDelimiters{0,1}, quoteCharacter*, literalCharacter*
Children	collapseDelimiters, fieldDelimiter, literalCharacter, quoteCharacter
Instance	<pre> &lt;simpleDelimited&gt;   &lt;fieldDelimiter&gt;{1,unbounded}&lt;/fieldDelimiter&gt;   &lt;collapseDelimiters&gt;{0,1}&lt;/collapseDelimiters&gt;   &lt;quoteCharacter&gt;{0,unbounded}&lt;/quoteCharacter&gt;   &lt;literalCharacter&gt;{0,unbounded}&lt;/literalCharacter&gt; &lt;/simpleDelimited&gt;   </pre>
Source	<pre> &lt;xs:element name="simpleDelimited"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A simple delimited format that uses one of a series of delimiters to indicate the ends of fields in the data stream. More complex formats such as fixed format or mixed delimited and fixed formats can be described using the "complex" element.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="fieldDelimiter" type="xs:string" maxOccurs="unbounded"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This element specifies a character to be used in the object for indicating the ending column for an attribute. The delimiter character itself is not part of the attribute value, but rather is present in the column following the last character of the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote characters can be provided as their hex values, and for tab characters by its ASCII string "\t". Processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="collapseDelimiters" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or multiple delimiters. An example is when a space delimiter is used; often there may be several repeated spaces that should be treated as a single delimiter, but not always. The valid values are yes or no. If it is set to yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive delimiters will be treated as separate delimiters, by default.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="quoteCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;         &lt;xs:annotation&gt;   </pre>

```

<xs:documentation>This element specifies a character to be used in the object for quoting values so that field delimiters can be used within the value. This basically allows delimiter "escaping". The quoteCharacter is typically a " or '. When a processor encounters a quote character, it should not interpret any following characters as a delimiter until a matching quote character has been encountered (i.e., quotes come in pairs). It is an error to not provide a closing quote before the record ends. Non-printable quote characters can be provided as their hex values.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="literalCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
<xs:annotation>
<xs:documentation>This element specifies a character to be used for escaping special character values so that they are treated as literal values. This allows "escaping" for special characters like quotes, commas, and spaces when they are intended to be used in an attribute value rather than being intended as a delimiter. The literalCharacter is typically a \.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:elements>

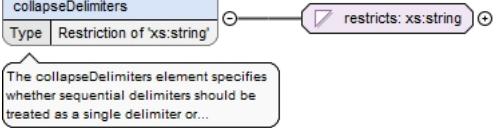
```

## Element dataFormat / textFormat / simpleDelimited / fieldDelimiter

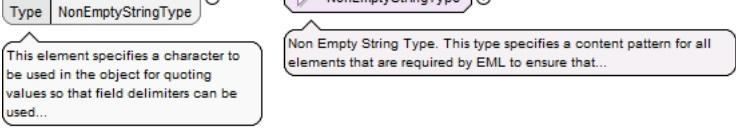
Namespace	No namespace				
Annotations	<p>This element specifies a character to be used in the object for indicating the ending column for an attribute. The delimiter character itself is not part of the attribute value, but rather is present in the column following the last character of the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote characters can be provided as their hex values, and for tab characters by its ASCII string "\t". Processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.</p>				
Diagram	<p>The diagram shows a UML class named 'fieldDelimiter' with a multiplicity of 0..1. It has a directed association to another class named 'xs:string' with a multiplicity of 1..1. A callout box points to the 'fieldDelimiter' class with the text: 'This element specifies a character to be used in the object for indicating the ending column for an attribute. The...'. Another callout box points to the 'xs:string' class with the text: 'Built-in primitive type. The string datatype represents character strings in XML.'</p>				
Type	xs:string				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	simple	maxOccurs:	unbounded
content:	simple				
maxOccurs:	unbounded				
Source	<pre> &lt;xs:element name="fieldDelimiter" type="xs:string" maxOccurs="unbounded"&gt; &lt;xs:annotation&gt; &lt;xs:documentation&gt;This element specifies a character to be used in the object for indicating the ending column for an attribute. The delimiter character itself is not part of the attribute value, but rather is present in the column following the last character of the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote characters can be provided as their hex values, and for tab characters by its ASCII string "\t". Processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>				

## Element dataFormat / textFormat / simpleDelimited / collapseDelimiters

Namespace	No namespace
-----------	--------------

Annotations	The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or multiple delimiters. An example is when a space delimiter is used; often there may be several repeated spaces that should be treated as a single delimiter, but not always. The valid values are yes or no. If it is set to yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive delimiters will be treated as separate delimiters, by default.				
Diagram					
Type	restriction of xs:string				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table> <tr> <td>enumeration</td> <td>yes</td> </tr> <tr> <td>enumeration</td> <td>no</td> </tr> </table>	enumeration	yes	enumeration	no
enumeration	yes				
enumeration	no				
Source	<pre>&lt;xs:element name="collapseDelimiters" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or multiple delimiters. An example is when a space delimiter is used; often there may be several repeated spaces that should be treated as a single delimiter, but not always. The valid values are yes or no. If it is set to yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive delimiters will be treated as separate delimiters, by default.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:string"&gt;       &lt;xs:enumeration value="yes"/&gt;       &lt;xs:enumeration value="no"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>				

### Element dataFormat / textFormat / simpleDelimited / quoteCharacter

Namespace	No namespace				
Annotations	This element specifies a character to be used in the object for quoting values so that field delimiters can be used within the value. This basically allows delimiter "escaping". The quoteCharacter is typically a " or '. When a processor encounters a quote character, it should not interpret any following characters as a delimiter until a matching quote character has been encountered (i.e., quotes come in pairs). It is an error to not provide a closing quote before the record ends. Non-printable quote characters can be provided as their hex values.				
Diagram					
Type	NonEmptyStringType				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				

	maxOccurs:	unbounded
Facets	minLength	1
	pattern	[ \s ]*[ \S ]*[ \s \S ]*
Source	<pre>&lt;xs:element name="quoteCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies a character to be used in the object for quoting values so that field delimiters can be used within the value. This basically allows delimiter "escaping". The quoteCharacter is typically a " or '. When a processor encounters a quote character, it should not interpret any following characters as a delimiter until a matching quote character has been encountered (i.e., quotes come in pairs). It is an error to not provide a closing quote before the record ends. Non-printable quote characters can be provided as their hex values.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>	

### Element **dataFormat / textFormat / simpleDelimited / literalCharacter**

Namespace	No namespace						
Annotations	<p>This element specifies a character to be used for escaping special character values so that they are treated as literal values.</p> <p>This allows "escaping" for special characters like quotes, commas, and spaces when they are intended to be used in an attribute value rather than being intended as a delimiter. The literalCharacter is typically a \.</p>						
Diagram	<pre> classDiagram     class literalCharacter {         Type NonEmptyStringType     }     class NonEmptyStringType     literalCharacter o-- NonEmptyStringType     </pre> <p>The diagram shows a UML class named 'literalCharacter' with a multiplicity of 0..1 at its end of the association. The other class is 'NonEmptyStringType'. A callout box points to the 'literalCharacter' class with the text: 'This element specifies a character to be used for escaping special character values so that they are treated as literal...'. Another callout box points to the 'NonEmptyStringType' class with the text: 'Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...'. Ellipses (...) are shown at the bottom right of the 'NonEmptyStringType' callout box.</p>						
Type	NonEmptyStringType						
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	unbounded
content:	simple						
minOccurs:	0						
maxOccurs:	unbounded						
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ]*[ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ]*[ \s \S ]*		
minLength	1						
pattern	[ \s ]*[ \S ]*[ \s \S ]*						
Source	<pre>&lt;xs:element name="literalCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies a character to be used for escaping special character values so that they are treated as literal values. This allows "escaping" for special characters like quotes, commas, and spaces when they are intended to be used in an attribute value rather than being intended as a delimiter. The literalCharacter is typically a \.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>						

### Element **dataFormat / textFormat / complex**

Namespace	No namespace
Annotations	<p>A complex text format that can describe delimited fields, fixed width fields, and mixtures of the two. This supports multiline records (where one record is distributed across multiple physical lines). When using the complex format, the number of textFixed and textDelimited elements should exactly equal the number of attributes that have been described for the entity, and the order of the textFixed and textDelimited elements should correspond to the order of the attributes as described in the entity. Thus, for a delimited file with fourteen attributes, one should provide exactly fourteen textDelimited elements.</p>

Diagram	<pre> classDiagram     class complex {         &lt;&lt;A complex text format that can describe delimited fields, fixed width fields, and mixtures of the two. This supports...&gt;&gt;     }     class textFixed {         &lt;&lt;Describes the physical format of data sequences that use a fixed number of characters in a specified position in the...&gt;&gt;     }     class textDelimited {         &lt;&lt;Describes the physical format of data sequences that use delimiters in the stream to locate attribute values. This...&gt;&gt;     }      complex &lt; -- textFixed     complex &lt; -- textDelimited   </pre>
Properties	content: complex
Model	textFixed   textDelimited
Children	textDelimited, textFixed
Instance	<pre> &lt;complex&gt;   &lt;textFixed&gt;{1,1}&lt;/textFixed&gt;   &lt;textDelimited&gt;{1,1}&lt;/textDelimited&gt; &lt;/complex&gt;   </pre>
Source	<pre> &lt;xss:element name="complex"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;A complex text format that can describe delimited fields, fixed width fields, and mixtures of the two. This supports multiline records (where one record is distributed across multiple physical lines). When using the complex format, the number of textFixed and textDelimited elements should exactly equal the number of attributes that have been described for the entity, and the order of the textFixed and textDelimited elements should correspond to the order of the attributes as described in the entity. Thus, for a delimited file with fourteen attributes, one should provide exactly fourteen textDelimited elements.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:complexType&gt;     &lt;xss:choice maxOccurs="unbounded"&gt;       &lt;xss:element name="textFixed"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;Describes the physical format of data sequences that use a fixed number of characters in a specified position in the stream to locate attribute values. This method is common in sensor-derived data and in legacy database systems. To parse it, one must know the number of characters for each attribute and the starting column and line to begin reading the value.&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;       &lt;xss:element name="fieldWidth" type="xs:unsignedLong"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;Fixed width fields have a set length, thus the end of the field can always be determined by adding the fieldWidth to the starting column number.&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;       &lt;xss:element name="lineNumber" type="xs:unsignedLong" minOccurs="0"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;The line on which the data field is found, when the data record is written over more than one physical line in the file. A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, the relative location of a data field must be indicated by both relative row and column number. The lineNumber should never greater than the number of physical lines per record.&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;       &lt;xss:element name="fieldStartColumn" type="xs:long" minOccurs="0"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;The starting column number for a fixed format attribute. Fixed width fields have a set length, thus the end of the field can always be determined by adding the fieldWidth to the starting column number. If the starting column is not provided, processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;       &lt;/xss:sequence&gt;     &lt;/xss:complexType&gt;   &lt;/xss:element&gt;   &lt;xss:element name="textDelimited"&gt;     &lt;xss:annotation&gt;       &lt;xss:documentation&gt;Describes the physical format of data sequences that use delimiters in the stream to locate attribute values. This method is common in data exported from spreadsheets and database systems, To parse it, one must know the character that indicates the end of each attribute and the line to begin reading the value.&lt;/xss:documentation&gt;     &lt;/xss:annotation&gt;   &lt;/xss:element&gt; </pre>

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="fieldDelimiter" type="xs:string">
      <xs:annotation>
        <xs:documentation>This element specifies a character to be used in the object for indicating the ending column for an attribute. The delimiter character itself is not part of the attribute value, but rather is present in the column following the last character of the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote characters can be provided as their hex values, and for tab characters by its ASCII string "\t". Processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="collapseDelimiters" minOccurs="0">
      <xs:annotation>
        <xs:documentation>The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or multiple delimiters. An example is when a space delimiter is used; often there may be several repeated spaces that should be treated as a single delimiter, but not always. The valid values are yes or no. If it is set to yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive delimiters will be treated as separate delimiters, by default.</xs:documentation>
      </xs:annotation>
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="yes"/>
        <xs:enumeration value="no"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:sequence>
  <xs:element name="lineNumber" type="xs:unsignedLong" minOccurs="0">
    <xs:annotation>
      <xs:documentation>A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, the relative location of a data field must be indicated by both relative row and column number. The lineNumber should never be greater than the number of physical lines per record. When parsing the first field on a physical line as a delimited field, they should assume that the field data starts in the first column. Otherwise, follow the rules indicated under fieldDelimiter.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="quoteCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
      <xs:documentation>This element specifies a character to be used in the object for quoting values so that field delimiters can be used within the value. This basically allows delimiter "escaping". The quoteCharacter is typically a " or '. When a processor encounters a quote character, it should not interpret any following characters as a delimiter until a matching quote character has been encountered (i.e., quotes come in pairs). It is an error to not provide a closing quote before the record ends. Non-printable quote characters can be provided as their hex values.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="literalCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
      <xs:documentation>This element specifies a character to be used for escaping special character values so that they are treated as literal values. This allows "escaping" for special characters like quotes, commas, and spaces when they are intended to be used in an attribute value rather than being intended as a delimiter. The literalCharacter is typically a \.</xs:documentation>
    </xs:annotation>
  </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>

```

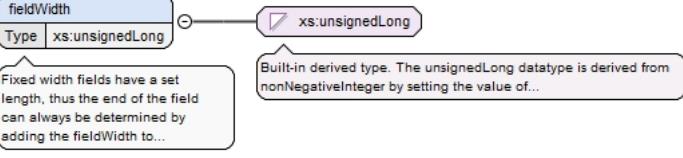
## Element dataFormat / textFormat / complex / textFixed

Namespace	No namespace
Annotations	Describes the physical format of data sequences that use a fixed number of characters in a specified position in the stream to locate attribute values. This method is common in sensor-derived data and in legacy database systems. To

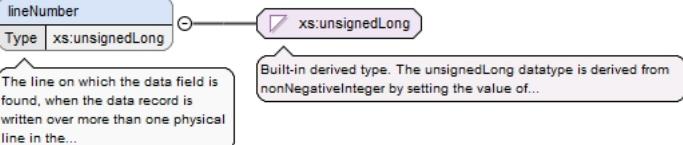
	<p>parse it, one must know the number of characters for each attribute and the starting column and line to begin reading the value.</p>
Diagram	
Properties	content: complex
Model	fieldWidth , lineNumber{0,1} , fieldStartColumn{0,1}
Children	fieldStartColumn, fieldWidth, lineNumber
Instance	<pre>&lt;textFixed&gt;   &lt;fieldWidth&gt;{1,1}&lt;/fieldWidth&gt;   &lt;lineNumber&gt;{0,1}&lt;/lineNumber&gt;   &lt;fieldStartColumn&gt;{0,1}&lt;/fieldStartColumn&gt; &lt;/textFixed&gt;</pre>
Source	<pre>&lt;xss:element name="textFixed"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;Describes the physical format of data sequences that use a fixed number of characters in a specified position in the stream to locate attribute values. This method is common in sensor-derived data and in legacy database systems. To parse it, one must know the number of characters for each attribute and the starting column and line to begin reading the value.&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt;   &lt;xss:complexType&gt;     &lt;xss:sequence&gt;       &lt;xss:element name="fieldWidth" type="xs:unsignedLong"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;Fixed width fields have a set length, thus the end of the field can always be determined by adding the fieldWidth to the starting column number.&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;       &lt;xss:element name="lineNumber" type="xs:unsignedLong" minOccurs="0"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;The line on which the data field is found, when the data record is written over more than one physical line in the file. A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, the relative location of a data field must be indicated by both relative row and column number. The lineNumber should never greater than the number of physical lines per record.&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;       &lt;xss:element name="fieldStartColumn" type="xs:long" minOccurs="0"&gt;         &lt;xss:annotation&gt;           &lt;xss:documentation&gt;The starting column number for a fixed format attribute. Fixed width fields have a set length, thus the end of the field can always be determined by adding the fieldWidth to the starting column number. If the starting column is not provided, processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.&lt;/xss:documentation&gt;         &lt;/xss:annotation&gt;       &lt;/xss:element&gt;     &lt;/xss:sequence&gt;   &lt;/xss:complexType&gt; &lt;/xss:element&gt;</pre>

## Element dataFormat / textFormat / complex / textFixed / fieldWidth

Namespace	No namespace
Annotations	Fixed width fields have a set length, thus the end of the field can always be determined by adding the fieldWidth to the starting

	column number.
Diagram	 <p>fieldWidth Type xs:unsignedLong</p> <p>Fixed width fields have a set length, thus the end of the field can always be determined by adding the fieldWidth to...</p> <p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>
Type	xs:unsignedLong
Properties	content: simple
Source	<pre>&lt;xs:element name="fieldWidth" type="xs:unsignedLong"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Fixed width fields have a set length, thus the end of the field can always be determined by adding the fieldWidth to the starting column number.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element dataFormat / textFormat / complex / textFixed / lineNumber

Namespace	No namespace
Annotations	<p>The line on which the data field is found, when the data record is written over more than one physical line in the file. A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, the relative location of a data field must be indicated by both relative row and column number. The lineNumber should never greater than the number of physical lines per record.</p>
Diagram	 <p>lineNumber Type xs:unsignedLong</p> <p>The line on which the data field is found, when the data record is written over more than one physical line in the...</p> <p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>
Type	xs:unsignedLong
Properties	<p>content: simple</p> <p>minOccurs: 0</p>
Source	<pre>&lt;xs:element name="lineNumber" type="xs:unsignedLong" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The line on which the data field is found, when the data record is written over more than one physical line in the file. A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, the relative location of a data field must be indicated by both relative row and column number. The lineNumber should never greater than the number of physical lines per record.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element dataFormat / textFormat / complex / textFixed / fieldStartColumn

Namespace	No namespace
Annotations	<p>The starting column number for a fixed format attribute. Fixed width fields have a set length, thus the end of the field can always be determined by adding the fieldWidth to the starting column number. If the starting column is not provided, processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the</p>

	delimiter from the previous field if the previous field was delimited.				
Diagram	<p>The starting column number for a fixed format attribute. Fixed width fields have a set length, thus the end of the...</p> <p>Built-in derived type. The long datatype is derived from integer by setting the value of maxInclusive to be...</p>				
Type	xs:long				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<pre>&lt;xs:element name="fieldStartColumn" type="xs:long" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The starting column number for a fixed format attribute. Fixed width fields have a set length, thus the end of the field can always be determined by adding the fieldWidth to the starting column number. If the starting column is not provided, processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>				

## Element dataFormat / textFormat / complex / textDelimited

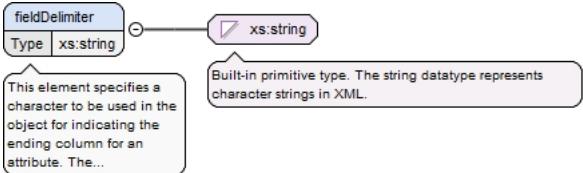
Namespace	No namespace		
Annotations	<p>Describes the physical format of data sequences that use delimiters in the stream to locate attribute values. This method is common in data exported from spreadsheets and database systems, To parse it, one must know the character that indicates the end of each attribute and the line to begin reading the value.</p>		
Diagram	<p>This element specifies a character to be used in the object for indicating the ending column for an attribute. The...</p> <p>The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or...</p> <p>A single logical data record may be written over several physical lines in a file, with no special marker to indicate...</p> <p>This element specifies a character to be used in the object for quoting values so that field delimiters can be used...</p> <p>This element specifies a character to be used for escaping special character values so that they are treated as literal...</p> <p>Describes the physical format of data sequences that use delimiters in the stream to locate attribute values. This...</p>		
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> </table>	content:	complex
content:	complex		
Model	fieldDelimiter , collapseDelimiters{0,1} , lineNumber{0,1} , quoteCharacter* , literalCharacter*		
Children	collapseDelimiters, fieldDelimiter, lineNumber, literalCharacter, quoteCharacter		
Instance	<pre>&lt;textDelimited&gt;   &lt;fieldDelimiter&gt;{1,1}&lt;/fieldDelimiter&gt;   &lt;collapseDelimiters&gt;{0,1}&lt;/collapseDelimiters&gt;   &lt;lineNumber&gt;{0,1}&lt;/lineNumber&gt;   &lt;quoteCharacter&gt;{0,unbounded}&lt;/quoteCharacter&gt;   &lt;literalCharacter&gt;{0,unbounded}&lt;/literalCharacter&gt;</pre>		

	</textDelimited>
Source	<pre> &lt;xs:element name="textDelimited"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Describes the physical format of data sequences that use delimiters in the stream to locate attribute values. This method is common in data exported from spreadsheets and database systems, To parse it, one must know the character that indicates the end of each attribute and the line to begin reading the value.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="fieldDelimiter" type="xs:string"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This element specifies a character to be used in the object for indicating the ending column for an attribute. The delimiter character itself is not part of the attribute value, but rather is present in the column following the last character of the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote characters can be provided as their hex values, and for tab characters by its ASCII string "\t". Processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="collapseDelimiters" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or multiple delimiters. An example is when a space delimiter is used; often there may be several repeated spaces that should be treated as a single delimiter, but not always. The valid values are yes or no. If it is set to yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive delimiters will be treated as separate delimiters, by default.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;         &lt;xs:simpleType&gt;           &lt;xs:restriction base="xs:string"&gt;             &lt;xs:enumeration value="yes"/&gt;             &lt;xs:enumeration value="no"/&gt;           &lt;/xs:restriction&gt;         &lt;/xs:simpleType&gt;       &lt;/xs:element&gt;       &lt;xs:element name="lineNumber" type="xs:unsignedLong" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, the relative location of a data field must be indicated by both relative row and column number. The lineNumber should never be greater than the number of physical lines per record. When parsing the first field on a physical line as a delimited field, they should assume that the field data starts in the first column. Otherwise, follow the rules indicated under fieldDelimiter.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="quoteCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This element specifies a character to be used in the object for quoting values so that field delimiters can be used within the value. This basically allows delimiter "escaping". The quoteCharacter is typically a " or '. When a processor encounters a quote character, it should not interpret any following characters as a delimiter until a matching quote character has been encountered (i.e., quotes come in pairs). It is an error to not provide a closing quote before the record ends. Non-printable quote characters can be provided as their hex values.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="literalCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;This element specifies a character to be used for escaping special character values so that they are treated as literal values. This allows "escaping" for special characters like quotes, commas, and spaces when they are intended to be used in an attribute value rather than being intended as a delimiter. The literalCharacter is typically a \.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

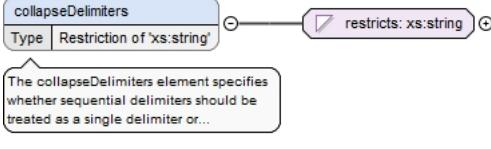
### Element dataFormat / textFormat / complex / textDelimited / fieldDelimiter

Namespace	No namespace
Annotations	This element specifies a character to be used

in the object for indicating the ending column for an attribute. The delimiter character itself is not part of the attribute value, but rather is present in the column following the last character of the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote characters can be provided as their hex values, and for tab characters by its ASCII string "\t". Processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.

Diagram	
Type	xs:string
Properties	content: simple
Source	<pre>&lt;xs:element name="fieldDelimiter" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This element specifies a character to be used in the object for indicating the ending column for an attribute. The delimiter character itself is not part of the attribute value, but rather is present in the column following the last character of the value. Typical delimiter characters include commas, tabs, spaces, and semicolons. The only time the fieldDelimiter character is not interpreted as a delimiter is if it is contained in a quoted string (see quoteCharacter) or is immediately preceded by a literalCharacter. Non-printable quote characters can be provided as their hex values, and for tab characters by its ASCII string "\t". Processors should assume that the field starts in the column following the previous field if the previous field was fixed, or in the column following the delimiter from the previous field if the previous field was delimited.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element dataFormat / textFormat / complex / textDelimited / collapseDelimiters

Namespace	No namespace
Annotations	<p>The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or multiple delimiters. An example is when a space delimiter is used; often there may be several repeated spaces that should be treated as a single delimiter, but not always. The valid values are yes or no. If it is set to yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive delimiters will be treated as separate delimiters, by default.</p>
Diagram	
Type	restriction of xs:string

Properties	content: simple minOccurs: 0
Facets	enumeration yes enumeration no
Source	<pre>&lt;xs:element name="collapseDelimiters" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The collapseDelimiters element specifies whether sequential delimiters should be treated as a single delimiter or multiple delimiters. An example is when a space delimiter is used; often there may be several repeated spaces that should be treated as a single delimiter, but not always. The valid values are yes or no. If it is set to yes, then consecutive delimiters will be collapsed to one. If set to no or absent, then consecutive delimiters will be treated as separate delimiters. Default behaviour is no; hence, consecutive delimiters will be treated as separate delimiters, by default.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:string"&gt;       &lt;xs:enumeration value="yes"/&gt;       &lt;xs:enumeration value="no"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:element&gt;</pre>

**Element dataFormat / textFormat / complex / textDelimited / lineNumber**

Namespace	No namespace
Annotations	<p>A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, the relative location of a data field must be indicated by both relative row and column number.</p> <p>The lineNumber should never be greater than the number of physical lines per record. When parsing the first field on a physical line as a delimited field, they should assume that the field data starts in the first column. Otherwise, follow the rules indicated under fieldDelimiter.</p>
Diagram	<pre> classDiagram     class lineNumber {         Type xs:unsignedLong     }     xs:unsignedLong &lt; --&gt; nonNegativeInteger     nonNegativeInteger &lt; --&gt; integer     integer &lt; --&gt; long     long &lt; --&gt; unsignedLong     unsignedLong &lt; --&gt; unsignedInt     unsignedInt &lt; --&gt; unsignedShort     unsignedShort &lt; --&gt; unsignedByte     unsignedByte &lt; --&gt; byte     byte &lt; --&gt; short     short &lt; --&gt; int     int &lt; --&gt; long     long &lt; --&gt; unsignedLong   </pre> <p>A single logical data record may be written over several physical lines in a file, with no special marker to indicate...</p> <p>Built-in derived type. The unsignedLong datatype is derived from nonNegativeInteger by setting the value of...</p>
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	<pre>&lt;xs:element name="lineNumber" type="xs:unsignedLong" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A single logical data record may be written over several physical lines in a file, with no special marker to indicate the end of a record. In such cases, the relative location of a data field must be indicated by both relative row and column number. The lineNumber should never be greater than the number of physical lines per record. When parsing the first field on a physical line as a delimited field, they should assume that the field data starts in the first column. Otherwise, follow the rules indicated under fieldDelimiter.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

**Element dataFormat / textFormat / complex / textDelimited / quoteCharacter**

Namespace	No namespace
Annotations	This element specifies a character to be used in the object for quoting values so that field delimiters can be used within the value. This basically allows delimiter "escaping". The

	<p>quoteCharacter is typically a " or '. When a processor encounters a quote character, it should not interpret any following characters as a delimiter until a matching quote character has been encountered (i.e., quotes come in pairs). It is an error to not provide a closing quote before the record ends. Non-printable quote characters can be provided as their hex values.</p>						
Diagram	<pre> classDiagram     class quoteCharacter {         &lt;&lt;NonEmptyStringType&gt;&gt;     }     class NonEmptyStringType     quoteCharacter --o NonEmptyStringType   </pre> <p>This element specifies a character to be used in the object for quoting values so that field delimiters can be used...</p> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p>						
Type	NonEmptyStringType						
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	unbounded
content:	simple						
minOccurs:	0						
maxOccurs:	unbounded						
Facets	<table> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ] [ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ] [ \s \S ]*		
minLength	1						
pattern	[ \s ]*[ \S ] [ \s \S ]*						
Source	<pre> &lt;x:element name="quoteCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;x:annotation&gt;     &lt;x:documentation&gt;This element specifies a character to be used in the object for quoting values so that field delimiters can be used within the value. This basically allows delimiter "escaping". The quoteCharacter is typically a " or '. When a processor encounters a quote character, it should not interpret any following characters as a delimiter until a matching quote character has been encountered (i.e., quotes come in pairs). It is an error to not provide a closing quote before the record ends. Non-printable quote characters can be provided as their hex values.&lt;/x:documentation&gt;   &lt;/x:annotation&gt; &lt;/x:element&gt;   </pre>						

## Element dataFormat / textFormat / complex / textDelimited / literalCharacter

Namespace	No namespace						
Annotations	<p>This element specifies a character to be used for escaping special character values so that they are treated as literal values. This allows "escaping" for special characters like quotes, commas, and spaces when they are intended to be used in an attribute value rather than being intended as a delimiter. The literalCharacter is typically a \.</p>						
Diagram	<pre> classDiagram     class literalCharacter {         &lt;&lt;NonEmptyStringType&gt;&gt;     }     class NonEmptyStringType     literalCharacter --o NonEmptyStringType   </pre> <p>This element specifies a character to be used for escaping special character values so that they are treated as literal...</p> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p>						
Type	NonEmptyStringType						
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	simple	minOccurs:	0	maxOccurs:	unbounded
content:	simple						
minOccurs:	0						
maxOccurs:	unbounded						
Facets	<table> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ] [ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ] [ \s \S ]*		
minLength	1						
pattern	[ \s ]*[ \S ] [ \s \S ]*						
Source	<pre> &lt;x:element name="literalCharacter" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;x:annotation&gt;     &lt;x:documentation&gt;This element specifies a character to be used for escaping special character values so that they are treated as literal values. This allows "escaping" for special characters   &lt;/x:documentation&gt; &lt;/x:element&gt;   </pre>						

<pre>like quotes, commas, and spaces when they are intended to be used in an attribute value rather than being intended as a delimiter. The literalCharacter is typically a \.&lt;/xs:documentation&gt; &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>
--

## Element dataFormat / externallyDefinedFormat

Namespace	No namespace
Annotations	Information about a non-text or proprietary formatted object.
Diagram	<pre> classDiagram     class externallyDefinedFormat {         formatName : xs:string         formatVersion : NonEmptyStringType     }     externallyDefinedFormat &lt; --&gt; Information     formatName &lt; --&gt; Name     formatVersion &lt; --&gt; Version   </pre> <p>The diagram illustrates the structure of the <code>externallyDefinedFormat</code> element. It is represented by a rounded rectangle with a hollow center. Two arrows point from it to two separate boxes: one for <code>formatName</code> and one for <code>formatVersion</code>. Each box contains a label (<code>formatName</code> or <code>formatVersion</code>) and a "Type" field (<code>xs:string</code> or <code>NonEmptyStringType</code>). Below each box is a descriptive text box: <code>formatName</code> is associated with "Name of the format of the data object, e.g., ESRI Shapefile.", and <code>formatVersion</code> is associated with "Version of the format of the data object".</p>
Properties	content: complex
Model	formatName , formatVersion{0,1}
Children	formatName, formatVersion
Instance	<pre> &lt;externallyDefinedFormat&gt;   &lt;formatName&gt;{1,1}&lt;/formatName&gt;   &lt;formatVersion&gt;{0,1}&lt;/formatVersion&gt; &lt;/externallyDefinedFormat&gt;   </pre>
Source	<pre> &lt;xs:element name="externallyDefinedFormat"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Information about a non-text or proprietary formatted object.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="formatName" type="xs:string"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Name of the format of the data object, e.g., ESRI Shapefile.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="formatVersion" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Version of the format of the data object&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;   </pre>

## Element dataFormat / externallyDefinedFormat / formatName

Namespace	No namespace
Annotations	Name of the format of the data object, e.g., ESRI Shapefile.
Diagram	<pre> classDiagram     class externallyDefinedFormat {         formatName : xs:string     }     externallyDefinedFormat &lt; --&gt; Information     formatName &lt; --&gt; xsString   </pre> <p>The diagram shows the <code>formatName</code> attribute of the <code>externallyDefinedFormat</code> element. It is represented by a rounded rectangle with a hollow center. An arrow points from it to another rounded rectangle labeled <code>xs:string</code>. Below the <code>formatName</code> box is a descriptive text box: "Name of the format of the data object, e.g., ESRI Shapefile.". Below the <code>xs:string</code> box is another descriptive text box: "Built-in primitive type. The string datatype represents character strings in XML."</p>
Type	xs:string
Properties	content: simple
Source	<pre> &lt;xs:element name="formatName" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Name of the format of the data object, e.g., ESRI Shapefile.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;   </pre>

## Element dataFormat / externallyDefinedFormat / formatVersion

Namespace	No namespace
-----------	--------------

Annotations	Version of the format of the data object				
Diagram	<pre> classDiagram     class formatVersion {         &lt;&lt;NonEmptyStringType&gt;&gt;     }     formatVersion &lt; -- NonEmptyStringType   </pre> <p>Version of the format of the data object</p> <p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...</p>				
Type	NonEmptyStringType				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ] [ \s \S ]*</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ] [ \s \S ]*
minLength	1				
pattern	[ \s ]*[ \S ] [ \s \S ]*				
Source	<pre> &lt;xss:element name="formatVersion" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;Version of the format of the data object&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;   </pre>				

### Element additionalMetadata / metadata / gbif / resourceLogoUrl

Namespace	No namespace				
Annotations	URL of the logo associated with a resource				
Diagram	<pre> classDiagram     class resourceLogoUrl {         &lt;&lt;xs:anyURI&gt;&gt;     }     resourceLogoUrl &lt; -- xs:anyURI   </pre> <p>URL of the logo associated with a resource</p> <p>Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).</p>				
Type	xs:anyURI				
Properties	<table border="1"> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Source	<pre> &lt;xss:element name="resourceLogoUrl" type="xs:anyURI" minOccurs="0"&gt;   &lt;xss:annotation&gt;     &lt;xss:documentation&gt;URL of the logo associated with a resource&lt;/xss:documentation&gt;   &lt;/xss:annotation&gt; &lt;/xss:element&gt;   </pre>				

### Element additionalMetadata / metadata / gbif / collection

Namespace	No namespace						
Annotations	A container element for other elements associated with collections (e.g., collectionIdentifier, collectionName).						
Diagram	<pre> classDiagram     class collection {         &lt;&lt;NonEmptyStringType&gt;&gt;         &lt;&lt;NonEmptyStringType&gt;&gt;         &lt;&lt;NonEmptyStringType&gt;&gt;     }     collection &lt; -- parentCollectionIdentifier     collection &lt; -- collectionIdentifier     collection &lt; -- collectionName   </pre> <p>A container element for other elements associated with collections (e.g., collectionIdentifier, collectionName).</p> <p>Identifier for the parent collection for this sub-collection. Enables a hierarchy of collections and sub collections to...</p> <p>The URI (LSID or URL) of the collection. In RDF, used as URI of the collection resource.</p> <p>Official name of the Collection in the local language</p>						
Properties	<table border="1"> <tr> <td>content:</td> <td>complex</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> <tr> <td>maxOccurs:</td> <td>unbounded</td> </tr> </table>	content:	complex	minOccurs:	0	maxOccurs:	unbounded
content:	complex						
minOccurs:	0						
maxOccurs:	unbounded						
Model	parentCollectionIdentifier{0,1} , collectionIdentifier{0,1} , collectionName						
Children	collectionIdentifier, collectionName, parentCollectionIdentifier						
Instance	<collection>						

	<pre> &lt;parentCollectionIdentifier&gt;{0,1}&lt;/parentCollectionIdentifier&gt; &lt;collectionIdentifier&gt;{0,1}&lt;/collectionIdentifier&gt; &lt;collectionName&gt;{1,1}&lt;/collectionName&gt; &lt;/collection&gt; </pre>
Source	<pre> &lt;xs:element name="collection" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A container element for other elements associated with collections (e.g., collectionIdentifier, collectionName).&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="parentCollectionIdentifier" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Identifier for the parent collection for this sub-collection. Enables a hierarchy of collections and sub collections to be built.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="collectionIdentifier" type="NonEmptyStringType" minOccurs="0"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The URI (LSID or URL) of the collection. In RDF, used as URI of the collection resource.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="collectionName" type="NonEmptyStringType"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;Official name of the Collection in the local language&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt; </pre>

### Element additionalMetadata / metadata / gbif / collection / parentCollectionIdentifier

Namespace	No namespace				
Annotations	Identifier for the parent collection for this sub-collection. Enables a hierarchy of collections and sub collections to be built.				
Diagram	<pre> classDiagram     class parentCollectionIdentifier {         Type NonEmptyStringType     }     parentCollectionIdentifier "0..1" -- "1" NonEmptyStringType     Note over parentCollectionIdentifier: Identifier for the parent collection for this sub-collection. Enables a hierarchy of collections and sub collections to be built...     Note over NonEmptyStringType: Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...   </pre>				
Type	NonEmptyStringType				
Properties	<table> <tr> <td>content:</td> <td>simple</td> </tr> <tr> <td>minOccurs:</td> <td>0</td> </tr> </table>	content:	simple	minOccurs:	0
content:	simple				
minOccurs:	0				
Facets	<table> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[\\s]*[\\S][\\s\\S]*</td> </tr> </table>	minLength	1	pattern	[\\s]*[\\S][\\s\\S]*
minLength	1				
pattern	[\\s]*[\\S][\\s\\S]*				
Source	<pre> &lt;xs:element name="parentCollectionIdentifier" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Identifier for the parent collection for this sub-collection. Enables a hierarchy of collections and sub collections to be built.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt; </pre>				

### Element additionalMetadata / metadata / gbif / collection / collectionIdentifier

Namespace	No namespace
Annotations	The URI (LSID or URL) of the collection. In RDF, used as URI of the collection resource.
Diagram	<pre> classDiagram     class collectionIdentifier {         Type NonEmptyStringType     }     collectionIdentifier "0..1" -- "1" NonEmptyStringType     Note over collectionIdentifier: The URI (LSID or URL) of the collection. In RDF, used as URI of the collection resource...     Note over NonEmptyStringType: Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...   </pre>
Type	NonEmptyStringType

Properties	content: simple minOccurs: 0
Facets	minLength 1 pattern [\s]*[\S][\s\S]*
Source	<pre>&lt;xs:element name="collectionIdentifier" type="NonEmptyStringType" minOccurs="0"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The URI (LSID or URL) of the collection. In RDF, used as URI of the collection resource.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

**Element additionalMetadata / metadata / gbif / collection / collectionName**

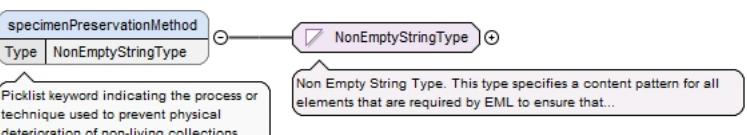
Namespace	No namespace
Annotations	Official name of the Collection in the local language
Diagram	<pre> classDiagram     class collectionName {         &lt;&lt;Official name of the Collection in the local language&gt;&gt;         &lt;&lt;Type NonEmptyStringType&gt;&gt;     }     class NonEmptyStringType {         &lt;&lt;Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...&gt;&gt;     }     collectionName "1" --o NonEmptyStringType   </pre>
Type	NonEmptyStringType
Properties	content: simple
Facets	minLength 1 pattern [\s]*[\S][\s\S]*
Source	<pre>&lt;xs:element name="collectionName" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Official name of the Collection in the local language&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

**Element additionalMetadata / metadata / gbif / formationPeriod**

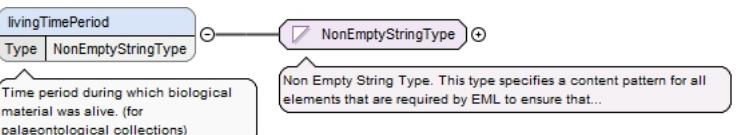
Namespace	No namespace
Annotations	Text description of the time period during which the collection was assembled e.g. "Victorian", or "1922 - 1932", or "c. 1750".
Diagram	<pre> classDiagram     class formationPeriod {         &lt;&lt;Text description of the time period during which the collection was assembled e.g. "Victorian", or "1922 - 1932", or...&gt;&gt;         &lt;&lt;Type NonEmptyStringType&gt;&gt;     }     class NonEmptyStringType {         &lt;&lt;Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that...&gt;&gt;     }     formationPeriod "1" --o NonEmptyStringType   </pre>
Type	NonEmptyStringType
Properties	content: simple minOccurs: 0 maxOccurs: unbounded
Facets	minLength 1 pattern [\s]*[\S][\s\S]*
Source	<pre>&lt;xs:element name="formationPeriod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Text description of the time period during which the collection was assembled e.g. "Victorian", or "1922 - 1932", or "c. 1750".&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

**Element additionalMetadata / metadata / gbif / specimenPreservationMethod**

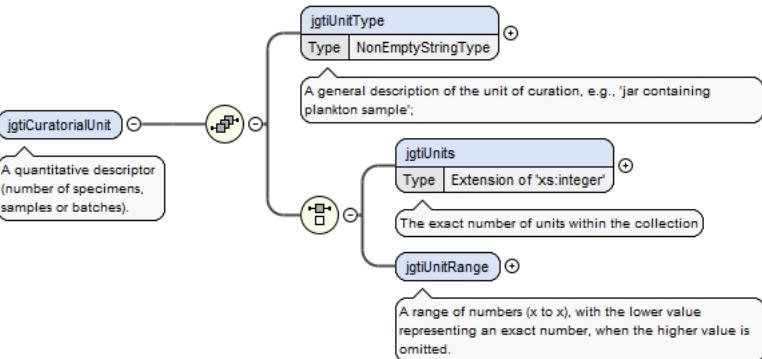
Namespace	No namespace
Annotations	Picklist keyword indicating the process or technique used to prevent physical deterioration of non-living collections. Expected to contain a value from the GBIF Specimen Preservation Method vocabulary

Diagram	
Type	NonEmptyStringType
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Facets	<p>minLength 1</p> <p>pattern <math>[\s]*[\S][\s]*</math></p>
Source	<pre>&lt;xs:element name="specimenPreservationMethod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Picklist keyword indicating the process or technique used to prevent physical deterioration of non-living collections. Expected to contain a value from the GBIF Specimen Preservation Method vocabulary&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element additionalMetadata / metadata / gbif / livingTimePeriod

Namespace	No namespace
Annotations	Time period during which biological material was alive. (for palaeontological collections)
Diagram	
Type	NonEmptyStringType
Properties	<p>content: simple</p> <p>minOccurs: 0</p> <p>maxOccurs: unbounded</p>
Facets	<p>minLength 1</p> <p>pattern <math>[\s]*[\S][\s]*</math></p>
Source	<pre>&lt;xs:element name="livingTimePeriod" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Time period during which biological material was alive. (for palaeontological collections)&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

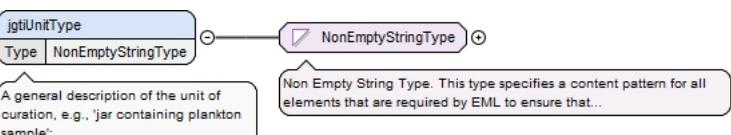
### Element jgtiCuratorialUnit

Namespace	No namespace
Annotations	A quantitative descriptor (number of specimens, samples or batches).
Diagram	

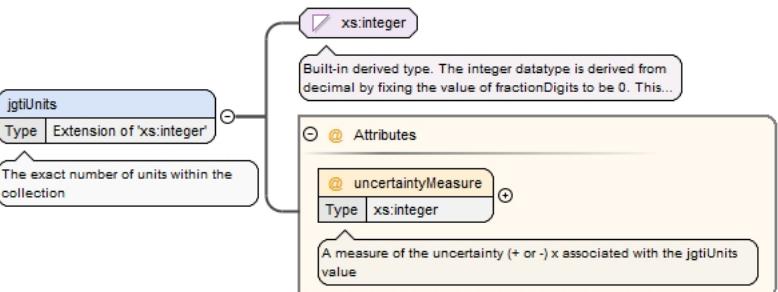
Properties	content: complex
Used by	Element additionalMetadata/metadata/gbif
Model	jgtiUnitType , (jgtiUnits   jgtiUnitRange)
Children	jgtiUnitRange, jgtiUnitType, jgtiUnits
Instance	<pre>&lt;jgtiCuratorialUnit&gt;   &lt;jgtiUnitType&gt;{1,1}&lt;/jgtiUnitType&gt;   &lt;jgtiUnits uncertaintyMeasure=""&gt;{1,1}&lt;/jgtiUnits&gt;   &lt;jgtiUnitRange&gt;{1,1}&lt;/jgtiUnitRange&gt; &lt;/jgtiCuratorialUnit&gt;</pre>
Source	<pre>&lt;xs:element name="jgtiCuratorialUnit"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A quantitative descriptor (number of specimens, samples or batches).&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="jgtiUnitType" type="NonEmptyStringType"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A general description of the unit of curation, e.g., 'jar containing plankton sample';&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:choice maxOccurs="1" minOccurs="1"&gt;         &lt;xs:element name="jgtiUnits"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;The exact number of units within the collection&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;           &lt;xs:complexType&gt;             &lt;xs:simpleContent&gt;               &lt;xs:extension base="xs:integer"&gt;                 &lt;xs:attribute name="uncertaintyMeasure" use="required" type="xs:integer"&gt;                   &lt;xs:annotation&gt;                     &lt;xs:documentation&gt;A measure of the uncertainty (+ or -) x associated with the jgtiUnits value&lt;/xs:documentation&gt;                   &lt;/xs:annotation&gt;                 &lt;/xs:attribute&gt;               &lt;/xs:extension&gt;             &lt;/xs:simpleContent&gt;           &lt;/xs:complexType&gt;         &lt;/xs:element&gt;         &lt;xs:element name="jgtiUnitRange"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;A range of numbers (x to x), with the lower value representing an exact number, when the higher value is omitted.&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;           &lt;xs:complexType&gt;             &lt;xs:sequence&gt;               &lt;xs:element name="beginRange" type="xs:integer"&gt;                 &lt;xs:annotation&gt;                   &lt;xs:documentation&gt;The lower value in a range of numbers. Use to represent an exact number by omitting the "endRange" value.&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;               &lt;/xs:element&gt;               &lt;xs:element name="endRange" type="xs:integer"&gt;                 &lt;xs:annotation&gt;                   &lt;xs:documentation&gt;The upper value in a range of numbers.&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;               &lt;/xs:element&gt;             &lt;/xs:sequence&gt;           &lt;/xs:complexType&gt;         &lt;/xs:element&gt;       &lt;/xs:choice&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

## Element jgtiCuratorialUnit / jgtiUnitType

Namespace	No namespace
Annotations	A general description of the unit of curation, e.g., 'jar containing plankton sample';

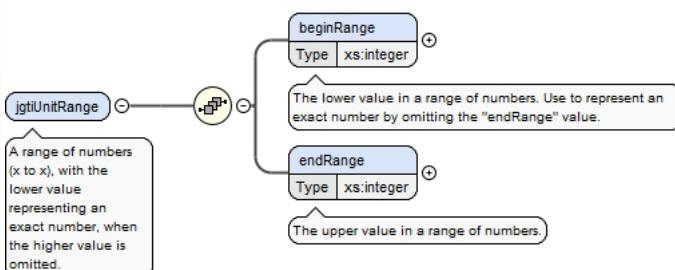
Diagram	
Type	NonEmptyStringType
Properties	content: simple
Facets	minLength 1 pattern [\s]*[\S][\s\S]*
Source	<pre>&lt;xs:element name="jgtiUnitType" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A general description of the unit of curation, e.g., 'jar containing plankton sample' ;&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Element jgtiCuratorialUnit / jgtiUnits

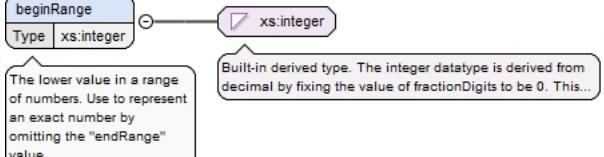
Namespace	No namespace														
Annotations	The exact number of units within the collection														
Diagram															
Type	extension of xs:integer														
Properties	content: complex														
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td>uncertaintyMeasure</td> <td>xs:integer</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">A measure of the uncertainty (+ or -) x associated with the jgtiUnits value</td></tr> </tbody> </table>	QName	Type	Use		uncertaintyMeasure	xs:integer	required			A measure of the uncertainty (+ or -) x associated with the jgtiUnits value				
QName	Type	Use													
uncertaintyMeasure	xs:integer	required													
	A measure of the uncertainty (+ or -) x associated with the jgtiUnits value														
Source	<pre>&lt;xs:element name="jgtiUnits"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The exact number of units within the collection&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:simpleContent&gt;       &lt;xs:extension base="xs:integer"&gt;         &lt;xs:attribute name="uncertaintyMeasure" use="required" type="xs:integer"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;A measure of the uncertainty (+ or -) x associated with the jgtiUnits value&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:attribute&gt;       &lt;/xs:extension&gt;     &lt;/xs:simpleContent&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>														

## Element jgtiCuratorialUnit / jgtiUnitRange

Namespace	No namespace
Annotations	A range of numbers (x to x), with the lower value representing an exact number, when the higher value is omitted.

Diagram	
Properties	content: complex
Model	beginRange , endRange
Children	beginRange, endRange
Instance	<pre>&lt;jgtiUnitRange&gt;   &lt;beginRange&gt;{1,1}&lt;/beginRange&gt;   &lt;endRange&gt;{1,1}&lt;/endRange&gt; &lt;/jgtiUnitRange&gt;</pre>
Source	<pre>&lt;xs:element name="jgtiUnitRange"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A range of numbers (x to x), with the lower value representing an exact number, when the higher value is omitted.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element name="beginRange" type="xs:integer"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The lower value in a range of numbers. Use to represent an exact number by omitting the "endRange" value.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="endRange" type="xs:integer"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The upper value in a range of numbers.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>

### Element jgtiCuratorialUnit / jgtiUnitRange / beginRange

Namespace	No namespace
Annotations	The lower value in a range of numbers. Use to represent an exact number by omitting the "endRange" value.
Diagram	
Type	xs:integer
Properties	content: simple
Source	<pre>&lt;xs:element name="beginRange" type="xs:integer"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The lower value in a range of numbers. Use to represent an exact number by omitting the "endRange" value.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

### Element jgtiCuratorialUnit / jgtiUnitRange / endRange

Namespace	No namespace
Annotations	The upper value in a range of numbers.

Diagram	
Type	xs:integer
Properties	content: simple
Source	<pre>&lt;xs:element name="endRange" type="xs:integer"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The upper value in a range of numbers.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:element&gt;</pre>

## Simple Type(s)

### Simple Type NonEmptyStringType

Namespace	No namespace						
Annotations	<p>Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that there is actual content (i.e., not just whitespace). The pattern described can be interpreted as "at least one non-whitespace character, followed by any number of whitespace plus not-whitespace characters." Leading and/or trailing whitespace is allowed, and whitespace may include carriage returns and newlines.</p>						
Diagram							
Type	restriction of xs:string						
Facets	<table border="1"> <tr> <td>minLength</td> <td>1</td> </tr> <tr> <td>pattern</td> <td>[ \s ]*[ \S ] [ \s \S ] *</td> </tr> </table>	minLength	1	pattern	[ \s ]*[ \S ] [ \s \S ] *		
minLength	1						
pattern	[ \s ]*[ \S ] [ \s \S ] *						
Used by	<table border="1"> <tr> <td>Elements</td> <td>additionalMetadata/metadata/gbif/collection/collectionIdentifier, additionalMetadata/metadata/gbif/collection/collectionName, additionalMetadata/metadata/gbif/collection/parentCollectionIdentifier, additionalMetadata/metadata/gbif/formationPeriod, additionalMetadata/metadata/gbif/hierarchyLevel, additionalMetadata/metadata/gbif/livingTimePeriod, additionalMetadata/metadata/gbif/specimenPreservationMethod, address/administrativeArea, address/city, address/country, address/deliveryPoint, address/postalCode, agentType/organizationName, agentType/positionName, awardType/awardNumber, awardType/awardUrl, awardType/funderIdentifier, awardType/funderName, awardType/title, characterEncoding, dataFormat/externallyDefinedFormat/formatVersion, dataFormat/textFormat/complex/textDelimited/literalCharacter, dataFormat/textFormat/complex/textDelimited/quoteCharacter, dataFormat/textFormat/simpleDelimited/literalCharacter, dataFormat/textFormat/simpleDelimited/quoteCharacter, dataset/maintenance/changeHistory/changeScope, dataset/maintenance/changeHistory/comment, dataset/maintenance/changeHistory/oldValue, dataset/shortName, electronicMailAddress, geographicCoverage/geographicDescription, individualName/givenName, individualName/salutation, individualName/surName, jgtiCuratorialUnit/jgtiUnitType, keywordSet/keyword, keywordSet/keywordThesaurus, language/licensed/identifier, licensed/licenseName, objectName, taxonomicCoverage/generalTaxonomicCoverage, taxonomicCoverage/taxonomicClassification/commonName, taxonomicCoverage/taxonomicClassification/taxonRankName, taxonomicCoverage/taxonomicClassification/taxonRankValue, ulink/citeTitle, userId</td> </tr> <tr> <td>Complex Types</td> <td>citationType, i18nString</td> </tr> <tr> <td>Attributes</td> <td>ulink/@url, userId/@directory</td> </tr> </table>	Elements	additionalMetadata/metadata/gbif/collection/collectionIdentifier, additionalMetadata/metadata/gbif/collection/collectionName, additionalMetadata/metadata/gbif/collection/parentCollectionIdentifier, additionalMetadata/metadata/gbif/formationPeriod, additionalMetadata/metadata/gbif/hierarchyLevel, additionalMetadata/metadata/gbif/livingTimePeriod, additionalMetadata/metadata/gbif/specimenPreservationMethod, address/administrativeArea, address/city, address/country, address/deliveryPoint, address/postalCode, agentType/organizationName, agentType/positionName, awardType/awardNumber, awardType/awardUrl, awardType/funderIdentifier, awardType/funderName, awardType/title, characterEncoding, dataFormat/externallyDefinedFormat/formatVersion, dataFormat/textFormat/complex/textDelimited/literalCharacter, dataFormat/textFormat/complex/textDelimited/quoteCharacter, dataFormat/textFormat/simpleDelimited/literalCharacter, dataFormat/textFormat/simpleDelimited/quoteCharacter, dataset/maintenance/changeHistory/changeScope, dataset/maintenance/changeHistory/comment, dataset/maintenance/changeHistory/oldValue, dataset/shortName, electronicMailAddress, geographicCoverage/geographicDescription, individualName/givenName, individualName/salutation, individualName/surName, jgtiCuratorialUnit/jgtiUnitType, keywordSet/keyword, keywordSet/keywordThesaurus, language/licensed/identifier, licensed/licenseName, objectName, taxonomicCoverage/generalTaxonomicCoverage, taxonomicCoverage/taxonomicClassification/commonName, taxonomicCoverage/taxonomicClassification/taxonRankName, taxonomicCoverage/taxonomicClassification/taxonRankValue, ulink/citeTitle, userId	Complex Types	citationType, i18nString	Attributes	ulink/@url, userId/@directory
Elements	additionalMetadata/metadata/gbif/collection/collectionIdentifier, additionalMetadata/metadata/gbif/collection/collectionName, additionalMetadata/metadata/gbif/collection/parentCollectionIdentifier, additionalMetadata/metadata/gbif/formationPeriod, additionalMetadata/metadata/gbif/hierarchyLevel, additionalMetadata/metadata/gbif/livingTimePeriod, additionalMetadata/metadata/gbif/specimenPreservationMethod, address/administrativeArea, address/city, address/country, address/deliveryPoint, address/postalCode, agentType/organizationName, agentType/positionName, awardType/awardNumber, awardType/awardUrl, awardType/funderIdentifier, awardType/funderName, awardType/title, characterEncoding, dataFormat/externallyDefinedFormat/formatVersion, dataFormat/textFormat/complex/textDelimited/literalCharacter, dataFormat/textFormat/complex/textDelimited/quoteCharacter, dataFormat/textFormat/simpleDelimited/literalCharacter, dataFormat/textFormat/simpleDelimited/quoteCharacter, dataset/maintenance/changeHistory/changeScope, dataset/maintenance/changeHistory/comment, dataset/maintenance/changeHistory/oldValue, dataset/shortName, electronicMailAddress, geographicCoverage/geographicDescription, individualName/givenName, individualName/salutation, individualName/surName, jgtiCuratorialUnit/jgtiUnitType, keywordSet/keyword, keywordSet/keywordThesaurus, language/licensed/identifier, licensed/licenseName, objectName, taxonomicCoverage/generalTaxonomicCoverage, taxonomicCoverage/taxonomicClassification/commonName, taxonomicCoverage/taxonomicClassification/taxonRankName, taxonomicCoverage/taxonomicClassification/taxonRankValue, ulink/citeTitle, userId						
Complex Types	citationType, i18nString						
Attributes	ulink/@url, userId/@directory						
Source	<pre>&lt;xs:simpleType name="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Non Empty String Type. This type specifies a content pattern for all elements that are required by EML to ensure that there is actual content (i.e., not just whitespace). The pattern described can be interpreted as "at least one non-whitespace character, followed by any number of whitespace plus not-whitespace characters." Leading and/or trailing whitespace is allowed, and whitespace may include carriage returns and newlines.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:minLength value="1"/&gt;     &lt;xs:pattern value="[ \s ]*[ \S ] [ \s \S ] *"/&gt;   &lt;/xs:restriction&gt;</pre>						

<code>&lt;/xs:simpleType&gt;</code>
-------------------------------------

## Simple Type IDType

Namespace	No namespace
Annotations	A unique identifier for this additional metadata that can be used to reference it elsewhere. This is a formal field in that it is an error to provide a value for the id attribute that is not unique within the document's set of id attributes. This is designed to allow other portions of the metadata to reference this section formally.
Diagram	<p>A diagram illustrating the relationship between the IDType and xs:string types. The IDType is represented by a purple rounded rectangle, and xs:string is represented by a blue rounded rectangle. A directed association line connects them. Callouts point from each type to their respective definitions: 'A unique identifier for this additional metadata that can be used to reference it elsewhere. This is a formal field in...' for IDType and 'Built-in primitive type. The string datatype represents character strings in XML.' for xs:string.</p>
Type	list of xs:string
Used by	Attribute agentType/@id
Source	<pre>&lt;xs:simpleType name="IDType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A unique identifier for this additional metadata that can be used to reference it elsewhere. This is a formal field in that it is an error to provide a value for the id attribute that is not unique within the document's set of id attributes. This is designed to allow other portions of the metadata to reference this section formally.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:list itemType="xs:string"/&gt; &lt;/xs:simpleType&gt;</pre>

## Simple Type SystemType

Namespace	No namespace
Annotations	The data management system within which an identifier is in scope and therefore unique. This is typically a URL (Uniform Resource Locator) that indicates a data management system. All identifiers that share a system must be unique. In other words, if the same identifier is used in two locations with identical systems, then by definition the objects at which they point are in fact the same object.
Diagram	<p>A diagram illustrating the relationship between the SystemType and xs:string types. The SystemType is represented by a purple rounded rectangle, and xs:string is represented by a blue rounded rectangle. A directed association line connects them. Callouts point from each type to their respective definitions: 'The data management system within which an identifier is in scope and therefore unique. This is typically a URL...' for SystemType and 'Built-in primitive type. The string datatype represents character strings in XML.' for xs:string.</p>
Type	list of xs:string
Used by	Attribute agentType/@system
Source	<pre>&lt;xs:simpleType name="SystemType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The data management system within which an identifier is in scope and therefore unique. This is typically a URL (Uniform Resource Locator) that indicates a data management system. All identifiers that share a system must be unique. In other words, if the same identifier is used in two locations with identical systems, then by definition the objects at which they point are in fact the same object.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:list itemType="xs:string"/&gt; &lt;/xs:simpleType&gt;</pre>

## Simple Type ScopeType

Namespace	No namespace
Annotations	The scope of the identifier. Scope is generally set to either "system", meaning that it is scoped according to the "system" attribute, or "document" if it is only to be in scope within this single document instance. In this particular use of

	scope, it is FIXED to be "system" because the packageId is required and always has the scope of the required "system".				
Diagram	<pre> classDiagram     class ScopeType {         &lt;&lt;Scope&gt;&gt;     }     class xsString {         &lt;&lt;xs:string&gt;&gt;     }     ScopeType &lt; -- xsString   </pre>				
Type	restriction of xs:string				
Facets	<table> <tr> <td>enumeration</td><td>system</td></tr> <tr> <td>enumeration</td><td>document</td></tr> </table>	enumeration	system	enumeration	document
enumeration	system				
enumeration	document				
Used by	Attribute agentType/@scope				
Source	<pre> &lt;xssimpleType name="ScopeType"&gt;   &lt;xssannotation&gt;     &lt;xssdocumentation&gt;The scope of the identifier. Scope is generally set to either "system", meaning that it is scoped according to the "system" attribute, or "document" if it is only to be in scope within this single document instance. In this particular use of scope, it is FIXED to be "system" because the packageId is required and always has the scope of the required "system".&lt;/xssdocumentation&gt;   &lt;/xssannotation&gt;   &lt;xssrestriction base="xs:string"&gt;     &lt;xssenumeration value="system"/&gt;     &lt;xssenumeration value="document"/&gt;   &lt;/xssrestriction&gt; &lt;/xssimpleType&gt;   </pre>				

## Simple Type yearDate

Namespace	No namespace
Annotations	A type allowing a year or date value. This type is the union of the built-in types for year and date. Example: 1999, or 2001-03-15
Diagram	<pre> classDiagram     class yearDate {         &lt;&lt;yearDate&gt;&gt;     }     class xsGYear {         &lt;&lt;xs:gYear&gt;&gt;     }     class xsDate {         &lt;&lt;xs:date&gt;&gt;     }     yearDate &lt; -- xsGYear     yearDate &lt; -- xsDate   </pre>
Type	union(xs:gYear, xs:date)
Used by	Elements calendarDate, pubDate
Source	<pre> &lt;xssimpleType name="yearDate"&gt;   &lt;xssannotation&gt;     &lt;xssdocumentation&gt;A type allowing a year or date value. This type is the union of the built-in types for year and date. Example: 1999, or 2001-03-15&lt;/xssdocumentation&gt;   &lt;/xssannotation&gt;   &lt;xssunion memberTypes="xs:gYear xs:date"/&gt; &lt;/xssimpleType&gt;   </pre>

## Simple Type MaintUpFreqType

Namespace	No namespace								
Diagram	<pre> classDiagram     class MaintUpFreqType {         &lt;&lt;MaintUpFreqType&gt;&gt;     }     class xsString {         &lt;&lt;xs:string&gt;&gt;     }     MaintUpFreqType &lt; -- xsString   </pre>								
Type	restriction of xs:string								
Facets	<table> <tr> <td>enumeration</td><td>annually</td></tr> <tr> <td>enumeration</td><td>asNeeded</td></tr> <tr> <td>enumeration</td><td>biannually</td></tr> <tr> <td>enumeration</td><td>continually</td></tr> </table>	enumeration	annually	enumeration	asNeeded	enumeration	biannually	enumeration	continually
enumeration	annually								
enumeration	asNeeded								
enumeration	biannually								
enumeration	continually								

	enumeration	daily
	enumeration	irregular
	enumeration	monthly
	enumeration	notPlanned
	enumeration	weekly
	enumeration	unknown
	enumeration	unkown
	enumeration	otherMaintenancePeriod
Used by	Element	dataset/maintenance/maintenanceUpdateFrequency
Source	<pre>&lt;xs:simpleType name="MaintUpFreqType"&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:enumeration value="annually"/&gt;     &lt;xs:enumeration value="asNeeded"/&gt;     &lt;xs:enumeration value="biannually"/&gt;     &lt;xs:enumeration value="continually"/&gt;     &lt;xs:enumeration value="daily"/&gt;     &lt;xs:enumeration value="irregular"/&gt;     &lt;xs:enumeration value="monthly"/&gt;     &lt;xs:enumeration value="notPlanned"/&gt;     &lt;xs:enumeration value="weekly"/&gt;     &lt;xs:enumeration value="unknown"/&gt;     &lt;xs:enumeration value="unkown"/&gt;     &lt;!-- deprecated, keep for compat., spelling error from prior version --&gt;     &lt;xs:enumeration value="otherMaintenancePeriod"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>	

## Simple Type descriptorEnum

Namespace	No namespace						
Diagram							
Type	restriction of xs:string						
Facets	<table border="1"> <tr> <td>enumeration</td> <td>thematic</td> </tr> <tr> <td>enumeration</td> <td>geographic</td> </tr> <tr> <td>enumeration</td> <td>generic</td> </tr> </table>	enumeration	thematic	enumeration	geographic	enumeration	generic
enumeration	thematic						
enumeration	geographic						
enumeration	generic						
Used by	Attribute descriptor/@name						
Source	<pre>&lt;xs:simpleType name="descriptorEnum"&gt;   &lt;xs:restriction base="xs:string"&gt;     &lt;xs:enumeration value="thematic"/&gt;     &lt;xs:enumeration value="geographic"/&gt;     &lt;xs:enumeration value="generic"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>						

## Complex Type(s)

### Complex Type i18nString

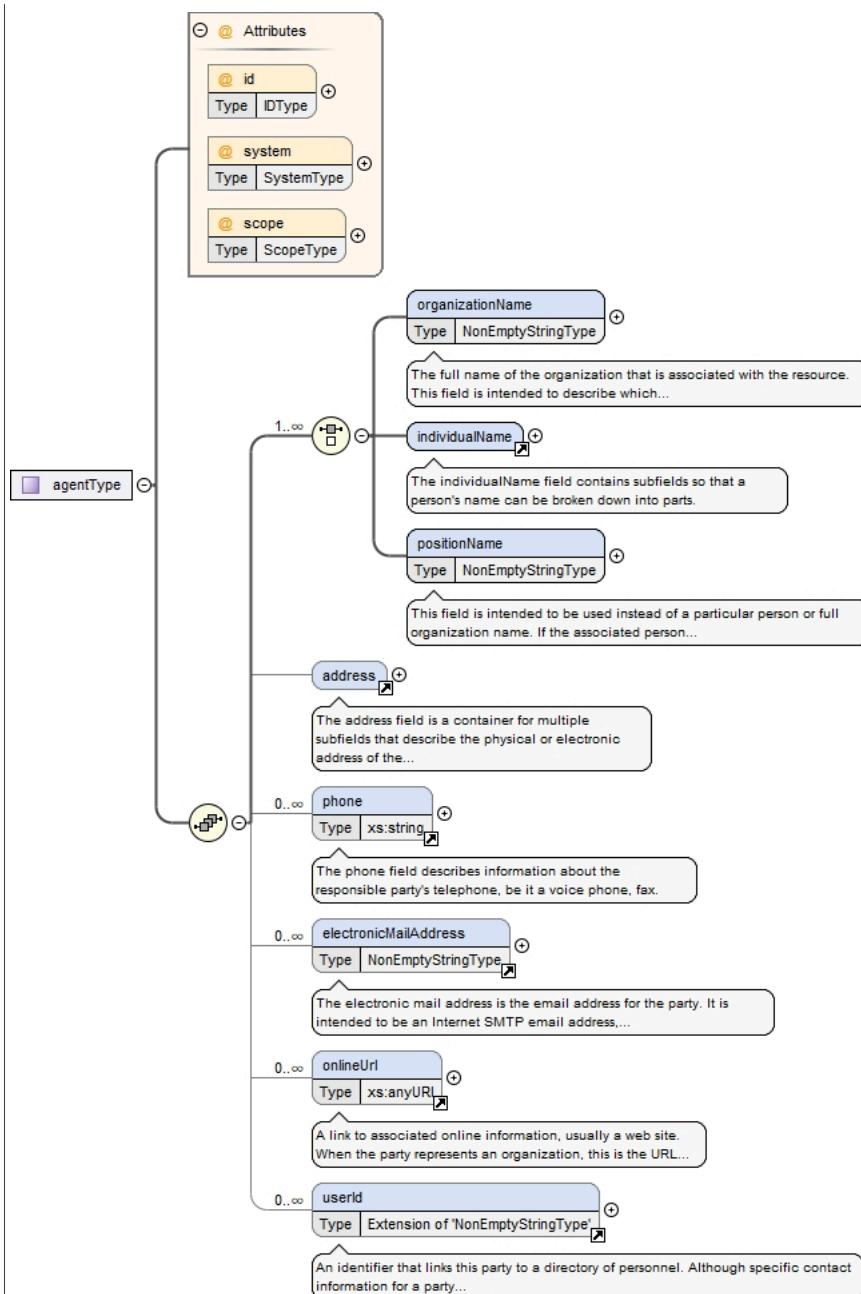
Namespace	No namespace
Diagram	

Type	extension of NonEmptyStringType		
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:string</li> <li>• NonEmptyStringType</li> <li>• i18nString</li> </ul>		
Used by	Elements	ParagraphType/emphasis/value, ParagraphType/literalLayout/value, ParagraphType/ulink/citetitle, ParagraphType/value, SectionType/title, SubSuperScriptType/value, title	
Attributes	<b>QName</b> <b>xml:lang</b>	<b>Type</b> union of(xs:language, restriction of xs:string)	<b>Use</b> optional
		<pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc- editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag- registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </pre>	
Source	<pre> &lt;xs:complexType name="i18nString"&gt;   &lt;xs:simpleContent&gt;     &lt;xs:extension base="NonEmptyStringType"&gt;       &lt;xs:attribute ref="xml:lang" use="optional"/&gt;     &lt;/xs:extension&gt;   &lt;/xs:simpleContent&gt; &lt;/xs:complexType&gt; </pre>		

## Complex Type agentType

Namespace	No namespace
-----------	--------------

## Diagram



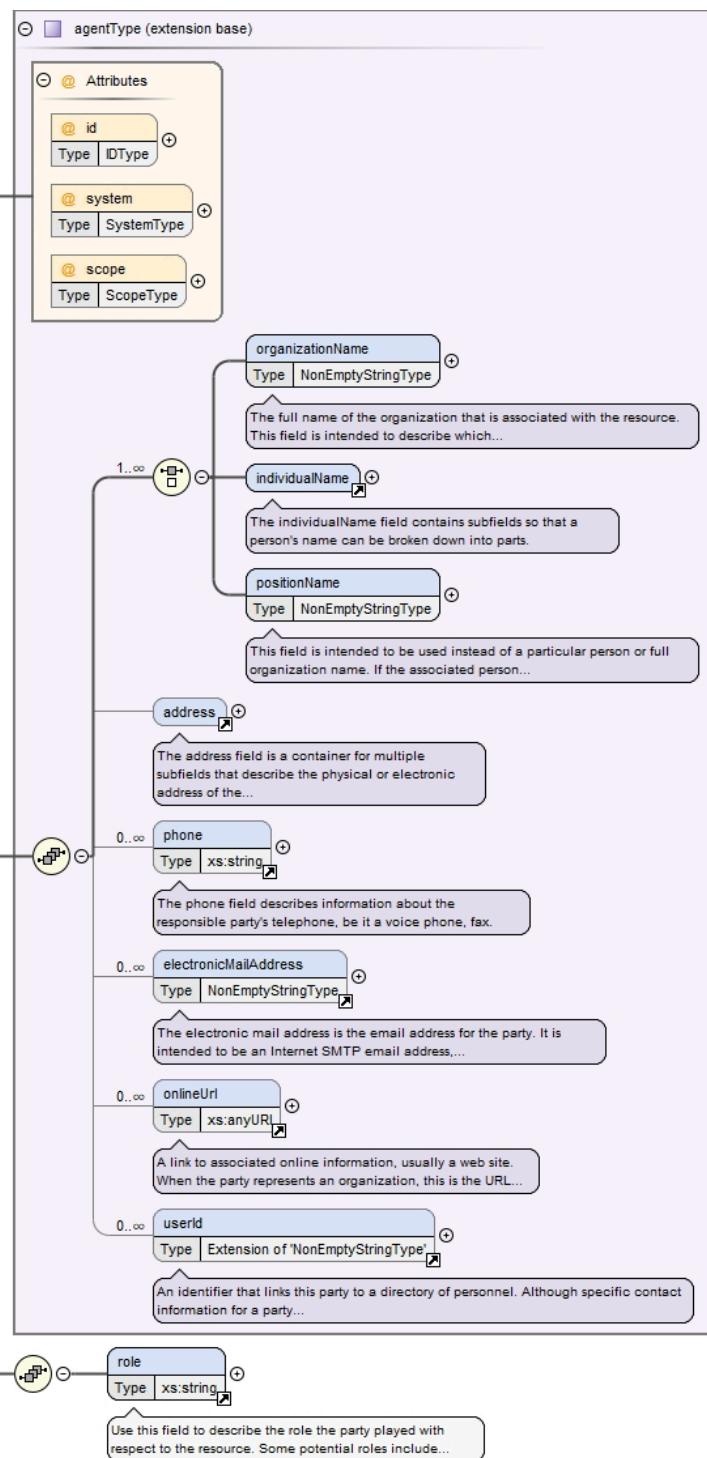
Used by	Elements dataset/contact, dataset/creator, dataset/metadataProvider, dataset/publisher Complex Type agentWithRoleType												
Model	(organizationName   individualName   positionName) , address{0,1} , phone* , electronicMailAddress* , onlineUrl* , userId*												
Children	address, electronicMailAddress, individualName, onlineUrl, organizationName, phone, positionName, userId												
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td><b>id</b></td><td>IDType</td><td>optional</td></tr> <tr> <td><b>scope</b></td><td>ScopeType</td><td>optional</td></tr> <tr> <td><b>system</b></td><td>SystemType</td><td>optional</td></tr> </tbody> </table>	QName	Type	Use	<b>id</b>	IDType	optional	<b>scope</b>	ScopeType	optional	<b>system</b>	SystemType	optional
QName	Type	Use											
<b>id</b>	IDType	optional											
<b>scope</b>	ScopeType	optional											
<b>system</b>	SystemType	optional											
Source	<pre>&lt;xs:complexType name="agentType"&gt;   &lt;xs:sequence&gt;     &lt;xs:choice maxOccurs="unbounded" minOccurs="1"&gt;       &lt;xs:element name="organizationName" type="NonEmptyStringType"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The full name of the organization that is associated with the resource. This field is intended to describe which institution or overall organization is associated with the resource being described.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;</pre>												

```
</xs:element>
<xs:element ref="individualName" />
<xs:element name="positionName" type="NonEmptyStringType" >
    <xs:annotation>
        <xs:documentation>This field is intended to be used instead of a particular person or full organization name. If the associated person who holds the role changes frequently, then Position Name would be used for consistency. E.g., GBIF Data Manager.</xs:documentation>
    </xs:annotation>
</xs:element>
</xs:choice>
<xs:element ref="address" minOccurs="0" />
<xs:element ref="phone" minOccurs="0" maxOccurs="unbounded" />
<xs:element ref="electronicMailAddress" minOccurs="0" maxOccurs="unbounded" />
<xs:element ref="onlineUrl" minOccurs="0" maxOccurs="unbounded" />
<xs:element ref="userId" minOccurs="0" maxOccurs="unbounded" />
</xs:sequence>
<xs:attribute name="id" type="IDType" use="optional" />
<xs:attribute name="system" type="SystemType" use="optional" />
<xs:attribute name="scope" type="ScopeType" use="optional" />
</xs:complexType>
```

## Complex Type agentWithRoleType

Namespace	No namespace
-----------	--------------

## Diagram



Type	extension of <code>agentType</code>									
Type hierarchy	<ul style="list-style-type: none"> <li>• <code>agentType</code></li> <li>• <code>agentWithRoleType</code></li> </ul>									
Used by	Elements associatedParty, projectType/personnel, relatedProjectType/personnel									
Model	( <code>organizationName</code>   <code>individualName</code>   <code>positionName</code> ) , <code>address</code> {0,1} , <code>phone</code> * , <code>electronicMailAddress</code> * , <code>onlineUrl</code> * , <code>userId</code> * , <code>role</code>									
Children	<code>address</code> , <code>electronicMailAddress</code> , <code>individualName</code> , <code>onlineUrl</code> , <code>organizationName</code> , <code>phone</code> , <code>positionName</code> , <code>role</code> , <code>userId</code>									
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th></tr> </thead> <tbody> <tr> <td><code>id</code></td><td><code>IDType</code></td><td>optional</td></tr> <tr> <td><code>scope</code></td><td><code>ScopeType</code></td><td>optional</td></tr> </tbody> </table>	QName	Type	Use	<code>id</code>	<code>IDType</code>	optional	<code>scope</code>	<code>ScopeType</code>	optional
QName	Type	Use								
<code>id</code>	<code>IDType</code>	optional								
<code>scope</code>	<code>ScopeType</code>	optional								

	QName	Type	Use	
	system	SystemType	optional	
Source	<pre>&lt;xs:complexType name="agentWithRoleType"&gt;   &lt;xs:complexContent&gt;     &lt;xs:extension base="agentType"&gt;       &lt;xs:sequence&gt;         &lt;xs:element ref="role"/&gt;       &lt;/xs:sequence&gt;     &lt;/xs:extension&gt;   &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt;</pre>			

## Complex Type TextType

Namespace	No namespace			
Annotations	<p>The "text" element allows for both formatted and unformatted text blocks to be included in EML. It can contain a number of relevant subsections that allow the use of titles, sections, and paragraphs in the text block. This markup is a subset of DocBook, or alternatively can be specified using Markdown text blocks.</p>			
Diagram	<pre> classDiagram     class TextType {         Mixed         true     }     class section {         Type         SectionType     }     class para {         Type         ParagraphType     }      TextType &lt; -- Mixed     TextType &lt; -- true     TextType --&gt; section : 1..oo     TextType --&gt; para : 0..oo   </pre> <p>The diagram illustrates the structure of the TextType complex type. It inherits from Mixed and true. It has two associations: one with the section element (multiplicity 1..oo) and another with the para element (multiplicity 0..oo). A callout box provides a detailed description of the "text" element, stating it allows for both formatted and unformatted text blocks and can contain sections and paragraphs.</p>			
Properties	mixed: true			
Used by	Elements abstract, acknowledgements, gettingStarted, introduction, purpose			
Model	section*   para*			
Children	para, section			
Attributes	QName	Type	Use	
	xml:lang	union of(xs:language, restriction of xs:string)	optional	
	<pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </pre>			
Source	<pre>&lt;xs:complexType name="TextType" mixed="true"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The "text" element allows for both formatted and unformatted text blocks to be included in EML. It can contain a number of relevant subsections that allow the use of titles, sections, and paragraphs in the text block. This markup is a subset of DocBook, or alternatively can be specified using Markdown text blocks.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:complexType&gt;</pre>			

```

</xs:annotation>
<xs:choice maxOccurs="unbounded">
  <xs:element name="section" type="SectionType" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
      <xs:documentation>The "section" element allows for grouping related paragraphs of text together, with an optional title. This markup is a subset of DocBook.</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="para" type="ParagraphType" minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
      <xs:documentation>The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.</xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:choice>
<xs:attribute ref="xml:lang"/>
</xs:complexType>

```

## Complex Type SectionType

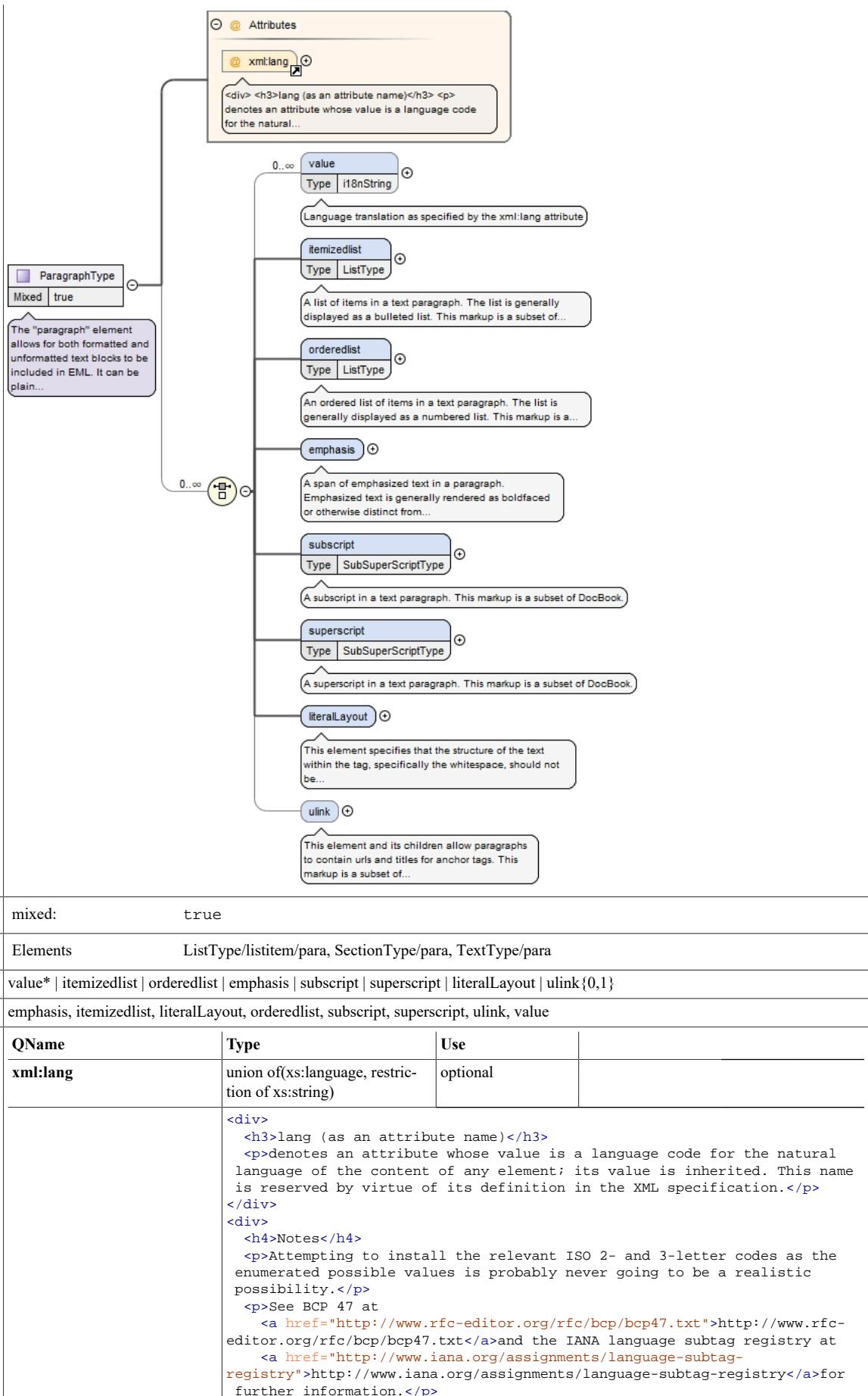
Namespace	No namespace									
Annotations	The "section" element allows for grouping related paragraphs (or other sections) of text together, with an optional title. This markup is a subset of DocBook.									
Diagram	<p>The diagram illustrates the structure of the SectionType complex type. It starts with a purple rounded rectangle labeled 'SectionType'. A line connects it to a box containing the XML schema definition. From this box, lines lead to four components: '@xml:lang' (an attribute), 'title' (with type 'i18nString'), 'para' (with type 'ParagraphType'), and 'section' (with type 'SectionType'). Each component has a descriptive callout box. The '@xml:lang' box says: '&lt;div&gt; lang (as an attribute name)&lt;/h3&gt; &lt;p&gt;denotes an attribute whose value is a language code for the natural...'. The 'title' box says: 'The optional title for a section. This markup is a subset of DocBook.' The 'para' box says: 'The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain...'. The 'section' box says: 'The "section" element allows for grouping related paragraphs of text together, with an optional title. This markup is a ...'.</p>									
Used by	Elements SectionType/section, TextType/section									
Model	title{0,1} , (para   section)									
Children	para, section, title									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>xml:lang</td> <td>union off(xs:language, restriction of xs:string)</td> <td>optional</td> </tr> <tr> <td></td> <td> <pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; </pre> </td> <td></td> </tr> </tbody> </table>	QName	Type	Use	xml:lang	union off(xs:language, restriction of xs:string)	optional		<pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; </pre>	
QName	Type	Use								
xml:lang	union off(xs:language, restriction of xs:string)	optional								
	<pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; </pre>									

	<b>QName</b>	<b>Type</b>	<b>Use</b>
Source	<pre>&lt;xs:complexType name="SectionType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The "section" element allows for grouping related paragraphs (or other sections) of text together, with an optional title. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="title" type="i18nString" minOccurs="0"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;The optional title for a section. This markup is a subset of DocBook.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;     &lt;xs:choice maxOccurs="unbounded"&gt;       &lt;xs:element name="para" type="ParagraphType"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;       &lt;xs:element name="section" type="SectionType"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;The "section" element allows for grouping related paragraphs of text together, with an optional title. This markup is a subset of DocBook.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:element&gt;     &lt;/xs:choice&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute ref="xml:lang"/&gt; &lt;/xs:complexType&gt;</pre>		

## Complex Type ParagraphType

Namespace	No namespace
Annotations	The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, lists and links. This markup is a subset of DocBook.

## Diagram



QName	Type	Use
Source		<p data-bbox="600 226 1416 294">&lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;</p>

```

<xs:documentation>The url attribute contains the location of the work for a link. This
markup is a subset of DocBook.</xs:documentation>
  </xs:annotation>
  </xs:attribute>
</xs:complexType>
</xs:element>
</xs:choice>
<xs:attribute ref="xml:lang"/>
</xs:complexType>

```

## Complex Type ListType

Namespace	No namespace
Annotations	A list of items in a text paragraph. The ListType is used by both orderedlist elements and itemizedlist elements. This markup is a subset of DocBook.
Diagram	<pre> classDiagram     class ListType     class listItem     ListType "1..oo" *-- listItem     </pre> <p>A list of items in a text paragraph. The ListType is used by both orderedlist elements and itemizedlist elements. This...</p> <p>An item in a list of items. Each list item is formatted as a bulleted or numbered item depending on the list type in...</p>
Used by	Elements ListType/listitem/itemizedlist, ListType/listitem/orderedlist, ParagraphType/itemizedlist, ParagraphType/orderedlist
Model	listitem+
Children	listitem
Source	<pre> &lt;xs:complexType name="ListType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A list of items in a text paragraph. The ListType is used by both orderedlist elements and itemizedlist elements. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="listitem" minOccurs="1" maxOccurs="unbounded"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;An item in a list of items. Each list item is formatted as a bulleted or numbered item depending on the list type in which it resides. List items contain paragraphs which in turn can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;       &lt;xs:complexType&gt;         &lt;xs:choice minOccurs="1" maxOccurs="unbounded"&gt;           &lt;xs:element name="para" type="ParagraphType"&gt;             &lt;xs:annotation&gt;               &lt;xs:documentation&gt;The "paragraph" element allows for both formatted and unformatted text blocks to be included in EML. It can be plain text or text with a limited set of markup tags, including emphasis, subscript, superscript, and lists. This markup is a subset of DocBook.&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;           &lt;/xs:element&gt;         &lt;/xs:choice&gt;         &lt;xs:element name="itemizedlist" type="ListType"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;A list of items in a text paragraph. The list is generally displayed as a bulleted list. This markup is a subset of DocBook.&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:element&gt;         &lt;xs:element name="orderedlist" type="ListType"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;An ordered list of items in a text paragraph. The list is generally displayed as a numbered list. This markup is a subset of DocBook.&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:element&gt;       &lt;/xs:complexType&gt;     &lt;/xs:sequence&gt;   &lt;/xs:complexType&gt; </pre>

## Complex Type SubSuperScriptType

Namespace	No namespace
Annotations	A subscript or a superscript in a text paragraph. This type is used by both subscript and superscript elements to define their recursive content. This markup is a subset of DocBook.

Diagram	<p>The diagram illustrates the UML class <code>SubSuperScriptType</code>. It is a <code>Mixed</code> type with <code>true</code> as its value. It inherits from <code>Mixed</code> and has an attribute <code>xml:lang</code>. It also contains four recursive associations: <code>value</code> (Type <code>i18nString</code>, multiplicity <code>0..oo</code>), <code>subscript</code> (Type <code>SubSuperScriptType</code>, multiplicity <code>0..oo</code>), and <code>superscript</code> (Type <code>SubSuperScriptType</code>, multiplicity <code>0..oo</code>). Each association is annotated with a box describing its purpose.</p>												
Properties	mixed: true												
Used by	Elements ParagraphType/subscript, ParagraphType/superscript, SubSuperScriptType/subscript, SubSuperScriptType/superscript												
Model	value*   subscript   superscript												
Children	subscript, superscript, value												
Attributes	<table border="1"> <thead> <tr> <th>QName</th><th>Type</th><th>Use</th><th></th></tr> </thead> <tbody> <tr> <td><code>xml:lang</code></td><td>union of(xs:language, restriction of xs:string)</td><td>optional</td><td></td></tr> <tr> <td></td><td> <pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt;</pre> </td><td></td><td></td></tr> </tbody> </table>	QName	Type	Use		<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional			<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt;</pre>		
QName	Type	Use											
<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional											
	<pre>&lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt;</pre>												
Source	<pre>&lt;x:complexType name="SubSuperScriptType" mixed="true"&gt;   &lt;x:annotation&gt;     &lt;x:documentation&gt;A subscript or a superscript in a text paragraph. This type is used by both subscript and superscript elements to define their recursive content. This markup is a subset of DocBook.&lt;/x:documentation&gt;   &lt;/x:annotation&gt;   &lt;x:choice minOccurs="0" maxOccurs="unbounded"&gt;     &lt;x:element name="value" type="i18nString" minOccurs="0" maxOccurs="unbounded"&gt;       &lt;x:annotation&gt;         &lt;x:documentation&gt;Language translation as specified by the xml:lang attribute&lt;/x:documentation&gt;       &lt;/x:annotation&gt;     &lt;/x:element&gt;     &lt;x:element name="subscript" type="SubSuperScriptType"&gt;       &lt;x:annotation&gt;         &lt;x:documentation&gt;A subscript in a text paragraph. This markup is a subset of DocBook.&lt;/x:documentation&gt;       &lt;/x:annotation&gt;     &lt;/x:element&gt;     &lt;x:element name="superscript" type="SubSuperScriptType"&gt;       &lt;x:annotation&gt;         &lt;x:documentation&gt;A superscript in a text paragraph. This markup is a subset of DocBook.&lt;/x:documentation&gt;       &lt;/x:annotation&gt;     &lt;/x:element&gt;   &lt;/x:choice&gt; &lt;/x:complexType&gt;</pre>												

```

</xs:element>
</xs:choice>
<xs:attribute ref="xml:lang" />
</xs:complexType>

```

## Complex Type calendarDate

Namespace	No namespace
Diagram	<p>The diagram illustrates the structure of the calendarDate complex type. It starts with a box labeled "calendarDate" with a self-referencing association loop. This leads to another box labeled "calendarDate" with a multiplicity of 1..*. From this, a single association arrow points to a box labeled "Type". Finally, an association arrow points to a box labeled "yearDate". A callout box provides a detailed description: "The calendar date field is used to express a date, giving the year, month, and day. The format should be one that...".</p>
Used by	Elements temporalCoverage/rangeOfDates/beginDate, temporalCoverage/rangeOfDates/endDate, temporalCoverage/singleDateTime
Model	calendarDate
Children	calendarDate
Source	<pre> &lt;xs:complexType name="calendarDate"&gt;   &lt;xs:sequence&gt;     &lt;xs:element ref="calendarDate" /&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

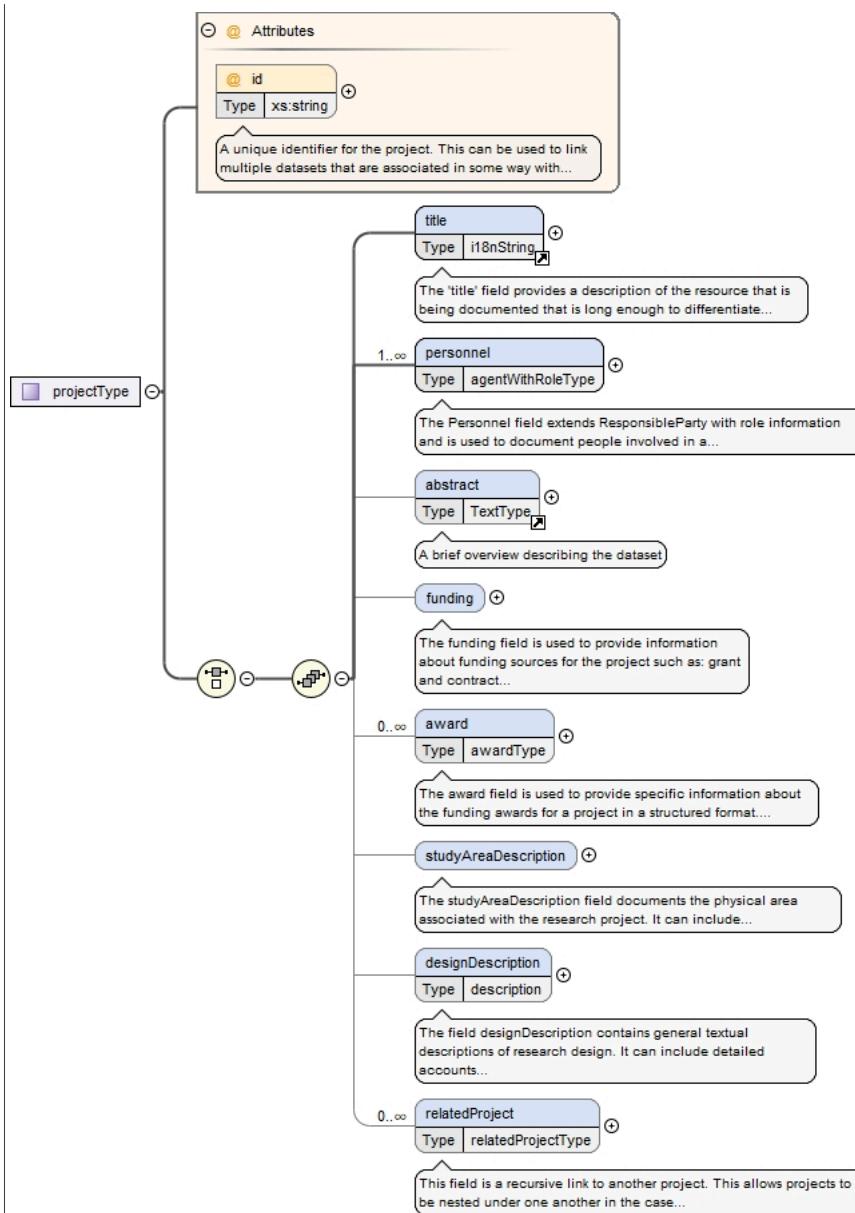
## Complex Type description

Namespace	No namespace
Diagram	<p>The diagram illustrates the structure of the description complex type. It starts with a box labeled "description" with a self-referencing association loop. This leads to another box labeled "description" with a multiplicity of 1..*. Finally, an association arrow points to a third box labeled "description". A callout box provides a detailed description: "The field Description contains general textual descriptions."</p>
Used by	Elements methods/methodStep, methods/qualityControl, methods/sampling/studyExtent, projectType/designDescription
Model	description
Children	description
Source	<pre> &lt;xs:complexType name="description"&gt;   &lt;xs:sequence&gt;     &lt;xs:element ref="description" /&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt; </pre>

## Complex Type projectType

Namespace	No namespace
-----------	--------------

## Diagram



Used by	Element <code>project</code>		
Model	<code>(title , personnel+, abstract{0,1} , funding{0,1} , award* , studyAreaDescription{0,1} , designDescription{0,1} , relatedProject*)</code>		
Children	abstract, award, designDescription, funding, personnel, relatedProject, studyAreaDescription, title		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>id</b>	<code>xs:string</code>	optional
		A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.	
Source	<pre> &lt;xs:complexType name="projectType"&gt;     &lt;xs:choice&gt;         &lt;xs:sequence&gt;             &lt;xs:element ref="title"/&gt;             &lt;xs:element name="personnel" type="agentWithRoleType" maxOccurs="unbounded"&gt;                 &lt;xs:annotation&gt;                     &lt;xs:documentation&gt;The Personnel field extends ResponsibleParty with role information and is used to document people involved in a research project by providing contact information and their role in the project.&lt;/xs:documentation&gt;                 &lt;/xs:annotation&gt;             &lt;/xs:element&gt;             &lt;xs:element ref="abstract" minOccurs="0"/&gt;             &lt;xs:element name="funding" minOccurs="0"&gt;                 &lt;xs:annotation&gt; </pre>		

```

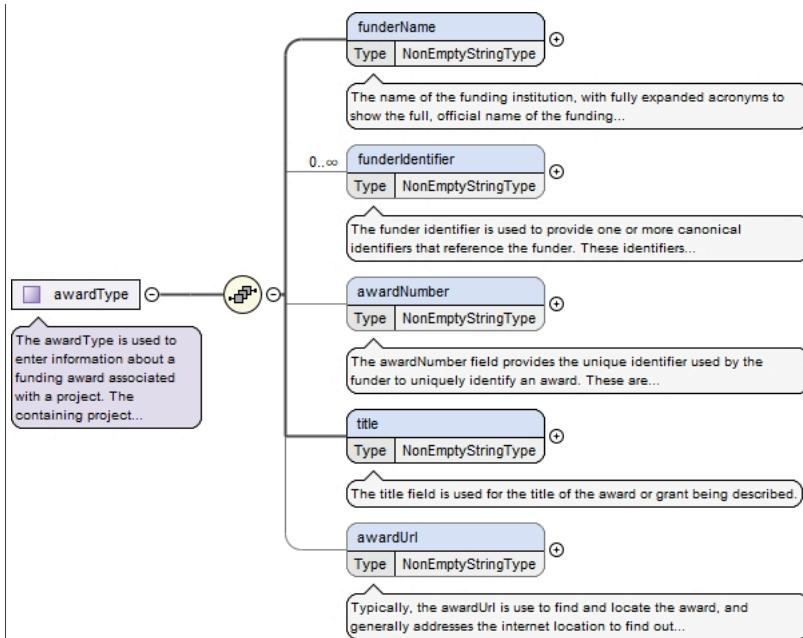
<xs:documentation>The funding field is used to provide information about funding sources for the project such as: grant and contract numbers; names and addresses of funding sources.</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element ref="para"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="award" type="awardType" minOccurs="0" maxOccurs="unbounded">
<xs:annotation>
<xs:documentation>The award field is used to provide specific information about the funding awards for a project in a structured format. Sub-fields are provided for the name of the funding agency, the Open Funder Registry identifiers for the agency and program that made the award, the award number assigned, the title of the award, and the URL to the award page describing the award. In general, the funding agency should be listed with a cross-reference to the appropriate identifier from the Open Funder Registry (included in the EML distribution but updated periodically from the Open Funder Registry).</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="studyAreaDescription" minOccurs="0">
<xs:annotation>
<xs:documentation>The studyAreaDescription field documents the physical area associated with the research project. It can include descriptions of the geographic, temporal, and taxonomic coverage of the research location and descriptions of domains (themes) of interest such as climate, geology, soils or disturbances.</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<xs:element ref="descriptor"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="designDescription" type="description" minOccurs="0">
<xs:annotation>
<xs:documentation>The field designDescription contains general textual descriptions of research design. It can include detailed accounts of goals, motivations, theory, hypotheses, strategy, statistical design, and actual work.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="relatedProject" type="relatedProjectType" minOccurs="0" maxOccurs="unbounded">
<xs:annotation>
<xs:documentation>This field is a recursive link to another project. This allows projects to be nested under one another in the case where one project spawns another.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:choice>
<xs:attribute name="id" type="xs:string" use="optional">
<xs:annotation>
<xs:documentation>A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.</xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>

```

## Complex Type awardType

Namespace	No namespace
Annotations	The awardType is used to enter information about a funding award associated with a project. The containing project contains the list of investigators and for the award, while the 'award' field contains specifics such as the agency name, award number, and funding program identifiers.

## Diagram



Used by	Element	projectType/award
Model		funderName , funderIdentifier* , awardNumber{0,1} , title , awardUrl{0,1}
Children		awardNumber, awardUrl, funderIdentifier, funderName, title
Source		<pre> &lt;xs:complexType name="awardType"&gt;     &lt;xs:annotation&gt;         &lt;xs:documentation&gt;The awardType is used to enter information about a funding award associated with a project. The containing project contains the list of investigators and for the award, while the `award` field contains specifics such as the agency name, award number, and funding program identifiers.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;     &lt;xs:sequence&gt;         &lt;xs:element name="funderName" type="NonEmptyStringType" minOccurs="1" maxOccurs="1"&gt;             &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The name of the funding institution, with fully expanded acronyms to show the full, official name of the funding agency. In general, this should match the official name of the funder as listed in an Authority such as the Open Funder Registry. The Open Funder Registry and other organizational authorities may provide a list of other alternative names for the funding agency.&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;         &lt;/xs:element&gt;         &lt;xs:element name="funderIdentifier" type="NonEmptyStringType" minOccurs="0" maxOccurs="unbounded"&gt;             &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The funder identifier is used to provide one or more canonical identifiers that reference the funder. These identifiers should be globally unique. The most common form of a funder identifier is a DOI identifier of an institution or program drawn from the CrossRef Open Funder Registry (<a href="https://gitlab.com/crossref/open_funder_registry">https://gitlab.com/crossref/open_funder_registry</a>), which assigns DOIs to each funding agency and to their programs, and links these together in a navigable hierarchy. A copy of the current Funder Registry is included as an RDF file with EML for reference, but as the list is constantly growing, users can retrieve new copies of the RDF file to get updates and current metadata about funders.&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;         &lt;/xs:element&gt;         &lt;xs:element name="awardNumber" type="NonEmptyStringType" minOccurs="0" maxOccurs="1"&gt;             &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The awardNumber field provides the unique identifier used by the funder to uniquely identify an award. These are typically alphanumeric values that are unique within the system used by a given funder. The number should be listed using the canonical form that each funder uses to express its award numbers, and not be prefixed or postfixed with extra text such as the acronym of the funder or the name of the funder, which is available instead in the funderName field.&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;         &lt;/xs:element&gt;         &lt;xs:element name="title" type="NonEmptyStringType" minOccurs="1" maxOccurs="1"&gt;             &lt;xs:annotation&gt;                 &lt;xs:documentation&gt;The title field is used for the title of the award or grant being described.&lt;/xs:documentation&gt;             &lt;/xs:annotation&gt;         &lt;/xs:element&gt;         &lt;xs:element name="awardUrl" type="NonEmptyStringType" minOccurs="0" maxOccurs="1"&gt;             &lt;xs:annotation&gt; </pre>

```

<xs:documentation>Typically, the awardUrl is used to find and locate the award, and generally addresses the internet location to find out more information about the award. This should point to a funder site for the award, rather than a project site.</xs:documentation>
</xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>

```

## Complex Type relatedProjectType

Namespace	No namespace									
Diagram	<pre> classDiagram     class relatedProjectType {         @ Attributes         @ id : xs:string         title : i18nString         personnel : agentWithRoleType         abstract : TextType     }     relatedProjectType &lt; --&gt; id     relatedProjectType &lt; --&gt; title     relatedProjectType &lt; --&gt; personnel     relatedProjectType &lt; --&gt; abstract </pre> <p>The diagram shows the structure of the relatedProjectType complex type. It consists of four fields: id (xs:string), title (i18nString), personnel (agentWithRoleType), and abstract (TextType). The id field is annotated with a note: "A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with...". The title field is annotated with a note: "The 'title' field provides a description of the resource that is being documented that is long enough to differentiate...". The personnel field is annotated with a note: "1..oo personnel agentWithRoleType". The abstract field is annotated with a note: "A brief overview describing the dataset".</p>									
Used by	Element projectType/relatedProject									
Model	(title , personnel+, abstract{0,1})									
Children	abstract, personnel, title									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><b>id</b></td> <td>xs:string</td> <td>optional</td> </tr> <tr> <td></td> <td></td> <td>A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.</td> </tr> </tbody> </table>	QName	Type	Use	<b>id</b>	xs:string	optional			A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.
QName	Type	Use								
<b>id</b>	xs:string	optional								
		A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.								
Source	<pre> &lt;xs:complexType name="relatedProjectType"&gt;     &lt;xs:choice&gt;         &lt;xs:sequence&gt;             &lt;xs:element ref="title"/&gt;             &lt;xs:element name="personnel" type="agentWithRoleType" maxOccurs="unbounded"/&gt;             &lt;xs:element ref="abstract" minOccurs="0"/&gt;         &lt;/xs:sequence&gt;     &lt;/xs:choice&gt;     &lt;xs:attribute name="id" type="xs:string" use="optional"&gt;         &lt;xs:annotation&gt;             &lt;xs:documentation&gt;A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>									

## Complex Type CitationListType

Namespace	No namespace
Diagram	<pre> classDiagram     class CitationListType {         1..oo bibtex : xs:string     } </pre> <p>The diagram shows the structure of the CitationListType complex type. It consists of one field: bibtex (xs:string). The bibtex field is annotated with a note: "The bibtex field provides a parseable list of citations formatted according to the Bibtex formatting conventions. Each..."</p>
Used by	Element dataset/literatureCited
Model	bibtex
Children	bibtex
Source	<pre> &lt;xs:complexType name="CitationListType"&gt;     &lt;xs:choice minOccurs="1" maxOccurs="unbounded"&gt; </pre>

```

<xs:element name="bibtex" type="xs:string">
  <xs:annotation>
    <xs:documentation>The bibtex field provides a parseable list of citations formatted according to the Bibtex formatting conventions. Each citation entry is assigned a unique key that must be unique across all bibtex fields in the EML document. The citation key can be used in markdown sections of the text to refer to this citation using the pandoc-style of inline citation keys. See the markdown element for more details. The record is delimited using curly braces. Most reference software can both import and export citations in Bibtex format, so this is a simpler representation to produce and consume than native EML citation representations.</xs:documentation>
  </xs:annotation>
</xs:element>
</xs:choice>
</xs:complexType>

```

## Complex Type citationType

Namespace	No namespace									
Annotations	A single literature citation with an optional identifier									
Diagram	<p>The diagram shows the UML class <code>citationType</code> which is a <code>NonEmptyStringType</code>. It has an attribute <code>@identifier</code> of type <code>xs:string</code>. A callout box explains that <code>NonEmptyStringType</code> specifies a content pattern for all elements required by EML.</p>									
Type	extension of NonEmptyStringType									
Type hierarchy	<ul style="list-style-type: none"> <li>• <code>xs:string</code></li> <li>• <code>NonEmptyStringType</code></li> <li>• <code>citationType</code></li> </ul>									
Used by	Elements additionalMetadata/metadata/gbif/bibliography/citation, additionalMetadata/metadata/gbif/citation									
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td><code>identifier</code></td> <td><code>xs:string</code></td> <td>optional</td> </tr> <tr> <td></td> <td></td> <td>A URI, DOI or other persistent identifier for the citation</td> </tr> </tbody> </table>	QName	Type	Use	<code>identifier</code>	<code>xs:string</code>	optional			A URI, DOI or other persistent identifier for the citation
QName	Type	Use								
<code>identifier</code>	<code>xs:string</code>	optional								
		A URI, DOI or other persistent identifier for the citation								
Source	<pre> &lt;xs:complexType name="citationType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A single literature citation with an optional identifier&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleContent&gt;     &lt;xs:extension base="NonEmptyStringType"&gt;       &lt;xs:attribute name="identifier" use="optional" type="xs:string"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;A URI, DOI or other persistent identifier for the citation&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:attribute&gt;     &lt;/xs:extension&gt;   &lt;/xs:simpleContent&gt; &lt;/xs:complexType&gt; </pre>									

## Attribute(s)

### Attribute `userId` / `@directory`

Namespace	No namespace				
Annotations	This attribute names the directory system to which this <code>userId</code> applies. This will generally be a URL that shows how to look up information, for example an LDAP url. However, it could also be a non-parsable description of the directory system if that is all that is available.				
Type	NonEmptyStringType				
Properties	use: required				
Facets	<table border="1"> <tr> <td><code>minLength</code></td> <td>1</td> </tr> <tr> <td><code>pattern</code></td> <td><code>[\s]*[\S][\s\S]*</code></td> </tr> </table>	<code>minLength</code>	1	<code>pattern</code>	<code>[\s]*[\S][\s\S]*</code>
<code>minLength</code>	1				
<code>pattern</code>	<code>[\s]*[\S][\s\S]*</code>				

Used by	Element	userId
Source		<pre>&lt;xs:attribute name="directory" use="required" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This attribute names the directory system to which this userId applies. This will generally be a URL that shows how to look up information, for example an LDAP url. However, it could also be a non-parsable description of the directory system if that is all that is available.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

**Attribute agentType / @id**

Namespace	No namespace	
Type	IDType	
Properties	use:	optional
Used by	Complex Type	agentType
Source	<pre>&lt;xs:attribute name="id" type="IDType" use="optional"/&gt;</pre>	

**Attribute agentType / @system**

Namespace	No namespace	
Type	SystemType	
Properties	use:	optional
Used by	Complex Type	agentType
Source	<pre>&lt;xs:attribute name="system" type="SystemType" use="optional"/&gt;</pre>	

**Attribute agentType / @scope**

Namespace	No namespace	
Type	ScopeType	
Properties	use:	optional
Facets	enumeration	system
	enumeration	document
Used by	Complex Type	agentType
Source	<pre>&lt;xs:attribute name="scope" type="ScopeType" use="optional"/&gt;</pre>	

**Attribute ParagraphType / ulink / @url**

Namespace	No namespace	
Annotations	The url attribute contains the location of the work for a link. This markup is a subset of DocBook.	
Properties	use:	optional
Used by	Element	ParagraphType/ulink
Source	<pre>&lt;xs:attribute name="url" use="optional"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;The url attribute contains the location of the work for a link. This markup is a subset of DocBook.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>	

**Attribute ulink / @url**

Namespace	No namespace	
Annotations	the url attribute contains the location of the work for a link.	
Type	NonEmptyStringType	
Properties	use:	required
Facets	minLength	1
	pattern	[ \s]*[ \S][ \s\S]*

Used by	Element	ulink
Source		<pre>&lt;xs:attribute name="url" use="required" type="NonEmptyStringType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;the url attribute contains the location of the work for a link.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

**Attribute url / @function**

Namespace	No namespace
Type	xs:string
Properties	use: required
Used by	Element url
Source	<pre>&lt;xs:attribute name="function" use="required" type="xs:string"/&gt;</pre>

**Attribute distribution / @scope**

Namespace	No namespace
Type	xs:string
Properties	content: simple
Used by	Element distribution
Source	<pre>&lt;xs:attribute name="scope" type="xs:string"/&gt;</pre>

**Attribute descriptor / @citableClassificationSystem**

Namespace	No namespace
Type	xs:boolean
Properties	use: optional
Used by	Element descriptor
Source	<pre>&lt;xs:attribute name="citableClassificationSystem" use="optional" type="xs:boolean"/&gt;</pre>

**Attribute descriptor / @name**

Namespace	No namespace						
Type	descriptorEnum						
Properties	use: optional						
Facets	<table> <tr> <td>enumeration</td> <td>thematic</td> </tr> <tr> <td>enumeration</td> <td>geographic</td> </tr> <tr> <td>enumeration</td> <td>generic</td> </tr> </table>	enumeration	thematic	enumeration	geographic	enumeration	generic
enumeration	thematic						
enumeration	geographic						
enumeration	generic						
Used by	Element descriptor						
Source	<pre>&lt;xs:attribute name="name" use="optional" type="descriptorEnum"/&gt;</pre>						

**Attribute relatedProjectType / @id**

Namespace	No namespace
Annotations	A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.
Type	xs:string
Properties	use: optional
Used by	Complex Type relatedProjectType
Source	<pre>&lt;xs:attribute name="id" type="xs:string" use="optional"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

**Attribute projectType / @id**

Namespace	No namespace	
Annotations	A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.	
Type	xs:string	
Properties	use: optional	
Used by	Complex Type	projectType
Source	<pre>&lt;xs:attribute name="id" type="xs:string" use="optional"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A unique identifier for the project. This can be used to link multiple datasets that are associated in some way with the same project.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>	

**Attribute citationType / @identifier**

Namespace	No namespace	
Annotations	A URI, DOI or other persistent identifier for the citation	
Type	xs:string	
Properties	use: optional	
Used by	Complex Type	citationType
Source	<pre>&lt;xs:attribute name="identifier" use="optional" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A URI, DOI or other persistent identifier for the citation&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>	

**Attribute jgtiCuratorialUnit / jgtiUnits / @uncertaintyMeasure**

Namespace	No namespace	
Annotations	A measure of the uncertainty (+ or -) x associated with the jgtiUnits value	
Type	xs:integer	
Properties	use: required	
Used by	Element	jgtiCuratorialUnit/jgtiUnits
Source	<pre>&lt;xs:attribute name="uncertaintyMeasure" use="required" type="xs:integer"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A measure of the uncertainty (+ or -) x associated with the jgtiUnits value&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>	

**Attribute eml:eml / @packageId**

Namespace	No namespace	
Annotations	Unique global ID for this exact version of the EML document	
Type	xs:string	
Properties	use: required	
Used by	Element	eml:eml
Source	<pre>&lt;xs:attribute name="packageId" use="required" type="xs:string"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Unique global ID for this exact version of the EML document&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>	

**Attribute eml:eml / @scope**

Namespace	No namespace	
Type	xs:string	
Properties	use:	required
	fixed:	system

Used by	Element	eml:eml
Source	<xs:attribute name="scope" use="required" type="xs:string" fixed="system" />	

### Attribute eml:eml / @system

Namespace	No namespace
Type	list of xs:string
Properties	use: required
Used by	Element eml:eml
Source	<xs:attribute name="system" use="required"> <xs:simpleType> <xs:list itemType="xs:string" /> </xs:simpleType> </xs:attribute>

## Namespace: "https://eml.ecoinformatics.org/eml-2.2.0"

### Schema(s)

#### Imported schema eml.xsd

Namespace	https://eml.ecoinformatics.org/eml-2.2.0
Properties	attribute form default: unqualified element form default: qualified

### Element(s)

#### Element eml:eml

Namespace	https://eml.ecoinformatics.org/eml-2.2.0								
Diagram	<pre> graph LR     eml[eml] --&gt; packageId[packageId]     eml --&gt; scope[scope]     eml --&gt; system[system]     eml --&gt; xmllang[xmllang]     packageId --- desc1[Unique global ID for this exact version of the EML document]     scope --- desc2[The scope of the metadata]     system --- desc3[The language in which the metadata is written]     xmllang --- desc4[The language in which the metadata is written]     dataset[dataset] --- desc5[The dataset element is a wrapper for all other elements relating to a single dataset]     additionalMetadata[additionalMetadata] --- desc6[A flexible field for including any other relevant metadata that pertains to the resource being described. This field...]   </pre>								
Properties	content: complex								
Model	dataset , additionalMetadata{0,1}								
Children	additionalMetadata, dataset								
Instance	<eml:eml xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:eml="https://eml.ecoinformatics.org/eml-2.2.0">   <dataset>{1,1}</dataset>   <additionalMetadata>{0,1}</additionalMetadata> </eml:eml>								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Fixed</th> <th>Use</th> </tr> </thead> <tbody> <tr> <td>packageId</td> <td>xs:string</td> <td></td> <td>required</td> </tr> </tbody> </table>	QName	Type	Fixed	Use	packageId	xs:string		required
QName	Type	Fixed	Use						
packageId	xs:string		required						

QName	Type	Fixed	Use	
Unique global ID for this exact version of the EML document				
<b>scope</b>	xs:string	system	required	
<b>system</b>	list of xs:string		required	
<b>xml:lang</b>	union of(xs:language, restriction of xs:string)		optional	
	Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at <a href="http://www.ietf.org/rfc/rfc3066.txt">http://www.ietf.org/rfc/rfc3066.txt</a> and the IANA registry at <a href="http://www.iana.org/assignments/lang-tag-apps.htm">http://www.iana.org/assignments/lang-tag-apps.htm</a> for further information.			
	The union allows for the 'un-declaration' of xml:lang with the empty string.			
Source	<pre> &lt;xs:element name="eml"&gt;   &lt;xs:complexType&gt;     &lt;xs:sequence&gt;       &lt;xs:element ref="dataset"/&gt;       &lt;xs:element ref="additionalMetadata" maxOccurs="1" minOccurs="0"/&gt;     &lt;/xs:sequence&gt;     &lt;xs:attribute name="packageId" use="required" type="xs:string"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;Unique global ID for this exact version of the EML document&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;     &lt;xs:attribute name="scope" use="required" type="xs:string" fixed="system"/&gt;     &lt;xs:attribute name="system" use="required"&gt;       &lt;xs:simpleType&gt;         &lt;xs:list itemType="xs:string"/&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:attribute&gt;     &lt;xs:attribute ref="xml:lang" use="optional"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;The language in which the metadata (as opposed to the resource being described by the metadata) is written&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:attribute&gt;   &lt;/xs:complexType&gt; &lt;/xs:element&gt;</pre>			

## Namespace: "http://www.w3.org/XML/1998/namespace"

### Schema(s)

#### Imported schema `xml.xsd`

Namespace	http://www.w3.org/XML/1998/namespace
Annotations	<p>See <a href="http://www.w3.org/XML/1998/namespace.html">http://www.w3.org/XML/1998/namespace.html</a> and <a href="http://www.w3.org/TR/REC-xml">http://www.w3.org/TR/REC-xml</a> for information about this namespace.</p> <p>This schema document describes the XML namespace, in a form suitable for import by other schema documents.</p> <p>Note that local names in this namespace are intended to be defined only by the World Wide Web Consortium or its subgroups. The following names are currently defined in this namespace and should not be used with conflicting semantics by any Working Group, specification, or document instance:</p> <p>base (as an attribute name): denotes an attribute whose value provides a URI to be used as the base for interpreting any relative URIs in the scope of the element on which it appears; its value is inherited. This name is reserved by virtue of its definition in the XML Base specification.</p> <p>id (as an attribute name): denotes an attribute whose value should be interpreted as if declared to be of type ID. This name is reserved by virtue of its definition in the <code>xml:id</code> specification.</p> <p>lang (as an attribute name): denotes an attribute whose value is a language code for the natural language of the content of</p>

	<p>any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.</p> <p>space (as an attribute name): denotes an attribute whose value is a keyword indicating what whitespace processing discipline is intended for the content of the element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.</p> <p>Father (in any context at all): denotes Jon Bosak, the chair of the original XML Working Group. This name is reserved by the following decision of the W3C XML Plenary and XML Coordination groups:</p> <pre>In appreciation for his vision, leadership and dedication the W3C XML Plenary on this 10th day of February, 2000 reserves for Jon Bosak in perpetuity the XML name xml:Father</pre> <p>This schema defines attributes and an attribute group suitable for use by schemas wishing to allow xml:base, xml:lang, xml:space or xml:id attributes on elements they define.</p> <p>To enable this, such a schema must import this schema for the XML namespace, e.g. as follows:</p> <pre>&lt;schema . . .&gt;   .   . &lt;import namespace="http://www.w3.org/XML/1998/namespace"   schemaLocation="http://www.w3.org/2001/xml.xsd"/&gt;</pre> <p>Subsequently, qualified reference to any of the attributes or the group defined below will have the desired effect, e.g.</p> <pre>&lt;type . . .&gt;   .   . &lt;attributeGroup ref="xml:specialAttrs"/&gt;  will define a type which will schema-validate an instance element with any of those attributes</pre> <p>In keeping with the XML Schema WG's standard versioning policy, this schema document will persist at <a href="http://www.w3.org/2007/08/xml.xsd">http://www.w3.org/2007/08/xml.xsd</a>. At the date of issue it can also be found at <a href="http://www.w3.org/2001/xml.xsd">http://www.w3.org/2001/xml.xsd</a>. The schema document at that URI may however change in the future, in order to remain compatible with the latest version of XML Schema itself, or with the XML namespace itself. In other words, if the XML Schema or XML namespaces change, the version of this document at <a href="http://www.w3.org/2001/xml.xsd">http://www.w3.org/2001/xml.xsd</a> will change accordingly; the version at <a href="http://www.w3.org/2007/08/xml.xsd">http://www.w3.org/2007/08/xml.xsd</a> will not change.</p>				
Properties	<table border="1"> <tr> <td>attribute form default:</td><td>unqualified</td></tr> <tr> <td>element form default:</td><td>unqualified</td></tr> </table>	attribute form default:	unqualified	element form default:	unqualified
attribute form default:	unqualified				
element form default:	unqualified				

## Imported schema `xml.xsd`

Namespace	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>
Annotations	<pre>&lt;div&gt;   &lt;h1&gt;About the XML namespace&lt;/h1&gt;   &lt;div class="bodytext"&gt;     &lt;p&gt;This schema document describes the XML namespace, in a form suitable for import by other schema documents.&lt;/p&gt;     &lt;p&gt;See       &lt;a href="http://www.w3.org/XML/1998/namespace.html"&gt;http://www.w3.org/XML/1998/       namespace.html&lt;/a&gt;and       &lt;a href="http://www.w3.org/TR/REC-xml"&gt;http://www.w3.org/TR/REC-xml&lt;/a&gt;for information about       this namespace.&lt;/p&gt;     &lt;p&gt;Note that local names in this namespace are intended to be defined only by the World Wide       Web Consortium or its subgroups. The names currently defined in this namespace are listed below.       They should not be used with conflicting semantics by any Working Group, specification, or document       instance.&lt;/p&gt;     &lt;p&gt;See further below in this document for more information about       &lt;a href="#usage"&gt;how to refer to this schema document from your own XSD schema documents&lt;/a&gt;     &lt;a href="#"&gt;and about       &lt;a href="#nsversioning"&gt;the namespace-versioning policy governing this schema document&lt;/a&gt;.&lt;/p&gt;   &lt;/div&gt; &lt;/div&gt;</pre>

```

<div>
  <h3>Father (in any context at all)</h3>
  <div class="bodytext">
    <p>denotes Jon Bosak, the chair of the original XML Working Group. This name is reserved by the following decision of the W3C XML Plenary and XML Coordination groups:</p>
    <blockquote>
      <p>In appreciation for his vision, leadership and dedication the W3C XML Plenary on this 10th day of February, 2000, reserves for Jon Bosak in perpetuity the XML name "xml:Father".</p>
    </blockquote>
  </div>
</div>

<div xml:id="usage" id="usage">
  <h2>
    <a name="usage">About this schema document</a>
  </h2>
  <div class="bodytext">
    <p>This schema defines attributes and an attribute group suitable for use by schemas wishing to allow
      <code>xml:base</code>,
      <code>xml:lang</code>,
      <code>xml:space</code>or
      <code>xml:id</code>attributes on elements they define.</p>
    <p>To enable this, such a schema must import this schema for the XML namespace, e.g. as follows:</p>
    <pre><schema . . .> . . . <import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/></pre>
    <p>or</p>
    <pre><import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="http://
www.w3.org/2009/01/xml.xsd"/></pre>
    <p>Subsequently, qualified reference to any of the attributes or the group defined below will have the desired effect, e.g.</p>
    <pre><type . . .> . . . <attributeGroup ref="xml:specialAttrs"/></pre>
    <p>will define a type which will schema-validate an instance element with any of those attributes.</p>
  </div>
</div>

<div id="nsversioning" xml:id="nsversioning">
  <h2>
    <a name="nsversioning">Versioning policy for this schema document</a>
  </h2>
  <div class="bodytext">
    <p>In keeping with the XML Schema WG's standard versioning policy, this schema document will persist at
      <a href="http://www.w3.org/2009/01/xml.xsd">http://www.w3.org/2009/01/xml.xsd</a>.</p>
    <p>At the date of issue it can also be found at
      <a href="http://www.w3.org/2001/xml.xsd">http://www.w3.org/2001/xml.xsd</a>.</p>
    <p>The schema document at that URI may however change in the future, in order to remain compatible with the latest version of XML Schema itself, or with the XML namespace itself. In other words, if the XML Schema or XML namespaces change, the version of this document at
      <a href="http://www.w3.org/2001/xml.xsd">http://www.w3.org/2001/xml.xsd</a>will change accordingly; the version at
      <a href="http://www.w3.org/2009/01/xml.xsd">http://www.w3.org/2009/01/xml.xsd</a>will not change.</p>
    <p>Previous dated (and unchanging) versions of this schema document are at:</p>
    <ul>
      <li>
        <a href="http://www.w3.org/2009/01/xml.xsd">http://www.w3.org/2009/01/xml.xsd</a>
      </li>
      <li>
        <a href="http://www.w3.org/2007/08/xml.xsd">http://www.w3.org/2007/08/xml.xsd</a>
      </li>
      <li>
        <a href="http://www.w3.org/2004/10/xml.xsd">http://www.w3.org/2004/10/xml.xsd</a>
      </li>
      <li>
        <a href="http://www.w3.org/2001/03/xml.xsd">http://www.w3.org/2001/03/xml.xsd</a>
      </li>
    </ul>
  </div>
</div>

```

Properties	attribute form default: unqualified element form default: unqualified
------------	--

## Attribute(s)

### Attribute @xml:lang

Namespace	http://www.w3.org/XML/1998/namespace
-----------	--------------------------------------

Annotations	<pre> &lt;div&gt;   &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;   &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt; &lt;/div&gt; &lt;div&gt;   &lt;h4&gt;Notes&lt;/h4&gt;   &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;   &lt;p&gt;See BCP 47 at     &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at     &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;   &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt; &lt;/div&gt; </pre>						
Type	union of(xs:language, restriction of xs:string)						
Properties	content: simple						
Used by	<table> <tr> <td>Elements</td><td>ParagraphType/emphasis, eml:eml</td></tr> <tr> <td>Complex Types</td><td>ParagraphType, SectionType, SubSuperScriptType, TextType, i18nString</td></tr> <tr> <td>Attribute Group</td><td>xml:specialAttrs</td></tr> </table>	Elements	ParagraphType/emphasis, eml:eml	Complex Types	ParagraphType, SectionType, SubSuperScriptType, TextType, i18nString	Attribute Group	xml:specialAttrs
Elements	ParagraphType/emphasis, eml:eml						
Complex Types	ParagraphType, SectionType, SubSuperScriptType, TextType, i18nString						
Attribute Group	xml:specialAttrs						
Source	<pre> &lt;xs:attribute name="lang"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;       &lt;div&gt;         &lt;h3&gt;lang (as an attribute name)&lt;/h3&gt;         &lt;p&gt;denotes an attribute whose value is a language code for the natural language of the content of any element; its value is inherited. This name is reserved by virtue of its definition in the XML specification.&lt;/p&gt;       &lt;/div&gt;       &lt;div&gt;         &lt;h4&gt;Notes&lt;/h4&gt;         &lt;p&gt;Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility.&lt;/p&gt;         &lt;p&gt;See BCP 47 at           &lt;a href="http://www.rfc-editor.org/rfc/bcp/bcp47.txt"&gt;http://www.rfc-editor.org/rfc/bcp/bcp47.txt&lt;/a&gt;and the IANA language subtag registry at           &lt;a href="http://www.iana.org/assignments/language-subtag-registry"&gt;http://www.iana.org/assignments/language-subtag-registry&lt;/a&gt;for further information.&lt;/p&gt;         &lt;p&gt;The union allows for the 'un-declaration' of xml:lang with the empty string.&lt;/p&gt;       &lt;/div&gt;     &lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:simpleType&gt;     &lt;xs:union memberTypes="xs:language"&gt;       &lt;xs:simpleType&gt;         &lt;xs:restriction base="xs:string"&gt;           &lt;xs:enumeration value="" /&gt;         &lt;/xs:restriction&gt;       &lt;/xs:simpleType&gt;     &lt;/xs:union&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; </pre>						

## Attribute @xml:space

Namespace	http://www.w3.org/XML/1998/namespace				
Type	restriction of xs:NCName				
Properties	content: simple				
Facets	<table> <tr> <td>enumeration</td><td>default</td></tr> <tr> <td>enumeration</td><td>preserve</td></tr> </table>	enumeration	default	enumeration	preserve
enumeration	default				
enumeration	preserve				
Used by	Attribute Group xml:specialAttrs				
Source	<pre> &lt;xs:attribute name="space"&gt;   &lt;xs:simpleType&gt;     &lt;xs:restriction base="xs:NCName"&gt;       &lt;xs:enumeration value="default" /&gt;       &lt;xs:enumeration value="preserve" /&gt;     &lt;/xs:restriction&gt;   &lt;/xs:simpleType&gt; &lt;/xs:attribute&gt; </pre>				

## Attribute @xml:base

Namespace	http://www.w3.org/XML/1998/namespace
Annotations	See <a href="http://www.w3.org/TR/xmlbase/">http://www.w3.org/TR/xmlbase/</a> for information about this attribute.
Type	xs:anyURI
Properties	content: simple
Used by	Attribute Group xml:specialAttrs
Source	<pre>&lt;xs:attribute name="base" type="xs:anyURI"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;See <a href="http://www.w3.org/TR/xmlbase/">http://www.w3.org/TR/xmlbase/</a> for information about this attribute.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

## Attribute @xml:id

Namespace	http://www.w3.org/XML/1998/namespace
Annotations	See <a href="http://www.w3.org/TR/xml-id/">http://www.w3.org/TR/xml-id/</a> for information about this attribute.
Type	xs:ID
Properties	content: simple
Used by	Attribute Group xml:specialAttrs
Source	<pre>&lt;xs:attribute name="id" type="xs:ID"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;See <a href="http://www.w3.org/TR/xml-id/">http://www.w3.org/TR/xml-id/</a> for information about this attribute.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt;</pre>

## Attribute Group(s)

### Attribute Group xml:specialAttrs

Namespace	http://www.w3.org/XML/1998/namespace																																		
Diagram	<p>The diagram illustrates the structure of the <code>xml:specialAttrs</code> attribute group. It shows a central box labeled <code>specialAttrs</code> containing four sub-boxes: <code>@ xml:base</code>, <code>@ xml:lang</code>, <code>@ xml:space</code>, and <code>@ xml:id</code>. Each of these sub-boxes has a callout line pointing to its corresponding XML documentation box.</p>																																		
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td><code>xml:base</code></td> <td>xs:anyURI</td> <td>optional</td> <td></td> </tr> <tr> <td></td> <td colspan="3">See <a href="http://www.w3.org/TR/xmlbase/">http://www.w3.org/TR/xmlbase/</a> for information about this attribute.</td></tr> <tr> <td><code>xml:id</code></td> <td>xs:ID</td> <td>optional</td> <td></td> </tr> <tr> <td></td> <td colspan="3">See <a href="http://www.w3.org/TR/xml-id/">http://www.w3.org/TR/xml-id/</a> for information about this attribute.</td></tr> <tr> <td><code>xml:lang</code></td> <td>union of(xs:language, restriction of xs:string)</td> <td>optional</td> <td></td> </tr> <tr> <td></td> <td colspan="3">Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going...</td></tr> <tr> <td></td> <td colspan="3">Attempt to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at <a href="http://www.ietf.org/rfc/rfc3066.txt">http://www.ietf.org/rfc/rfc3066.txt</a> and the IANA registry</td></tr> </tbody> </table>	QName	Type	Use		<code>xml:base</code>	xs:anyURI	optional			See <a href="http://www.w3.org/TR/xmlbase/">http://www.w3.org/TR/xmlbase/</a> for information about this attribute.			<code>xml:id</code>	xs:ID	optional			See <a href="http://www.w3.org/TR/xml-id/">http://www.w3.org/TR/xml-id/</a> for information about this attribute.			<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional			Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going...				Attempt to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at <a href="http://www.ietf.org/rfc/rfc3066.txt">http://www.ietf.org/rfc/rfc3066.txt</a> and the IANA registry				
QName	Type	Use																																	
<code>xml:base</code>	xs:anyURI	optional																																	
	See <a href="http://www.w3.org/TR/xmlbase/">http://www.w3.org/TR/xmlbase/</a> for information about this attribute.																																		
<code>xml:id</code>	xs:ID	optional																																	
	See <a href="http://www.w3.org/TR/xml-id/">http://www.w3.org/TR/xml-id/</a> for information about this attribute.																																		
<code>xml:lang</code>	union of(xs:language, restriction of xs:string)	optional																																	
	Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going...																																		
	Attempt to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at <a href="http://www.ietf.org/rfc/rfc3066.txt">http://www.ietf.org/rfc/rfc3066.txt</a> and the IANA registry																																		

	<b>QName</b>	<b>Type</b>	<b>Use</b>	
		at <a href="http://www.iana.org/assignments/lang-tag-apps.htm">http://www.iana.org/assignments/lang-tag-apps.htm</a> for further information.		
		The union allows for the 'un-declaration' of xml:lang with the empty string.		
	<b>xml:space</b>	restriction of xs:N QName	optional	
Source	<pre>&lt;xs:attributeGroup name="specialAttrs"&gt;   &lt;xs:attribute ref="xml:base"/&gt;   &lt;xs:attribute ref="xml:lang"/&gt;   &lt;xs:attribute ref="xml:space"/&gt;   &lt;xs:attribute ref="xml:id"/&gt; &lt;/xs:attributeGroup&gt;</pre>			

## Namespace: "http://purl.org/dc/terms/"

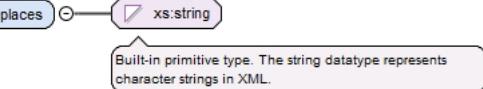
### Schema(s)

#### Imported schema dc.xsd

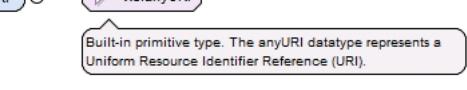
Namespace	http://purl.org/dc/terms/
Properties	attribute form default: unqualified element form default: unqualified

### Element(s)

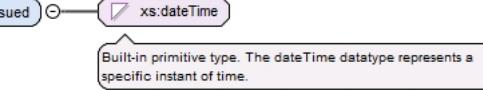
#### Element replaces

Namespace	http://purl.org/dc/terms/
Diagram	 <p>Built-in primitive type. The string datatype represents character strings in XML.</p>
Type	xs:string
Properties	content: simple
Used by	Element additionalMetadata/metadata/gbif
Source	<xs:element name="replaces" type="xs:string"/>

#### Element URI

Namespace	http://purl.org/dc/terms/
Diagram	 <p>Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).</p>
Type	xs:anyURI
Properties	content: simple
Source	<xs:element name="URI" type="xs:anyURI"/>

#### Element issued

Namespace	http://purl.org/dc/terms/
Diagram	 <p>Built-in primitive type. The dateTime datatype represents a specific instant of time.</p>
Type	xs:dateTime
Properties	content: simple
Source	<xs:element name="issued" type="xs:dateTime"/>

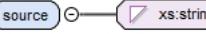
## Element type

Namespace	http://purl.org/dc/terms/
Diagram	 <div style="border: 1px solid black; padding: 5px; margin-top: 5px;">           Built-in primitive type. The string datatype represents character strings in XML.         </div>
Type	xs:string
Properties	content: simple
Source	<xs:element name="type" type="xs:string"/>

## Element title

Namespace	http://purl.org/dc/terms/
Diagram	 <div style="border: 1px solid black; padding: 5px; margin-top: 5px;">           Built-in primitive type. The string datatype represents character strings in XML.         </div>
Type	xs:string
Properties	content: simple
Source	<xs:element name="title" type="xs:string"/>

## Element source

Namespace	http://purl.org/dc/terms/
Diagram	 <div style="border: 1px solid black; padding: 5px; margin-top: 5px;">           Built-in primitive type. The string datatype represents character strings in XML.         </div>
Type	xs:string
Properties	content: simple
Source	<xs:element name="source" type="xs:string"/>

## Element subject

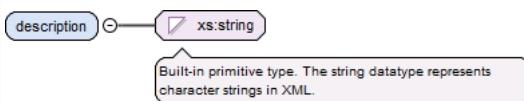
Namespace	http://purl.org/dc/terms/
Diagram	 <div style="border: 1px solid black; padding: 5px; margin-top: 5px;">           Built-in primitive type. The string datatype represents character strings in XML.         </div>
Type	xs:string
Properties	content: simple
Source	<xs:element name="subject" type="xs:string"/>

## Element relation

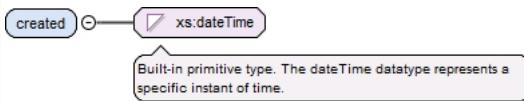
Namespace	http://purl.org/dc/terms/
Diagram	 <div style="border: 1px solid black; padding: 5px; margin-top: 5px;">           Built-in primitive type. The anyURI datatype represents a Uniform Resource Identifier Reference (URI).         </div>
Type	xs:anyURI
Properties	content: simple
Source	<xs:element name="relation" type="xs:anyURI"/>

## Element description

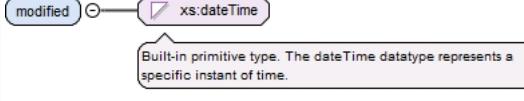
Namespace	http://purl.org/dc/terms/
-----------	---------------------------

Diagram	
Type	xs:string
Properties	content: simple
Source	<code>&lt;xss:element name="description" type="xs:string" /&gt;</code>

## Element created

Namespace	http://purl.org/dc/terms/
Diagram	
Type	xs:dateTime
Properties	content: simple
Source	<code>&lt;xss:element name="created" type="xs:dateTime" /&gt;</code>

## Element modified

Namespace	http://purl.org/dc/terms/
Diagram	
Type	xs:dateTime
Properties	content: simple
Source	<code>&lt;xss:element name="modified" type="xs:dateTime" /&gt;</code>