Best Practices for Dataset Metadata in Ecological Metadata Language

Version 4.0

Gregory Maurer Corinna Gries Gabriel Kamener
Sage Lichtenwalner Mary Martin Margaret O'Brien
John Porter Tim Whiteaker
LTER Network Information Management Committee

2026-08-26

This book presents "best practice" recommendations for creating metadata documents in the Ecological Metadata Language (EML), a widely accepted standard for research metadata exchange in the environmental sciences. The document focuses on the most common use-cases for EML, and recommends content and formatting for its most important and frequently-used metadata elements. As such, these recommendations can be applied to most research datasets that include EML metadata. There are also recommendations specific to the U.S. LTER Network and Environmental Data Initiative (EDI) repository. This is Version 4 of the book, updated in 2024 by the EML Best Practices Working Group of the LTER Network Information Management Committee.

Table of contents

1 Best Practices for Dataset Metadata in Ecological Metadata Language

Version 4.0

This book presents "best practice" recommendations for creating metadata documents in the Ecological Metadata Language (EML), a widely accepted standard for research metadata exchange in the environmental sciences. The document focuses on the most common use-cases for EML, and recommends content and formatting for its most important and frequently-used metadata elements. As such, these recommendations can be applied to most research datasets that include EML metadata. There are also recommendations specific to the U.S. LTER Network and Environmental Data Initiative (EDI) repository. This is Version 4 of the book, updated in 2024 by the EML Best Practices Working Group of the LTER Network Information Management Committee.

2 Introducing the Ecological Metadata Language

Key points

- EML provides a framework for metadata, but still leaves many choices up to the creator.
- This guide helps dataset creators make those choices based on community-level experience with EML metadata.

The Ecological Metadata Language (EML) is a standard for exchanging structured metadata that describes environmental or ecological research and data products. The research products, or data, being described may be tabular data files, such as a table of numerical values, spatial data files, prose documents, images, computer code, or any number of other digital outputs from scientific research or data collection activities. The purpose of EML is to contain the metadata, defined as "data that provides information about other data," necessary to understand and reuse data that is shared, published, or otherwise distributed. Metadata are most useful when they are detailed and can be easily understood by data users, and with appropriate preparation, EML metadata can easily meet this expectation.

The EML standard is a dialect of the eXtensible Markup Language (XML, see Appendix B) and, as such, is a very flexible, modular, and extendable container for standardized information. All XML documents, including EML, are designed to be both human and machine readable. Metadata documents written in EML can therefore describe many types of research data, with as much detail as necessary or desired, while remaining accessible to data users. This makes EML highly effective in a wide range of applications for publishing, discovering, accessing, and reusing research data.

Preparing clear, useful metadata to describe a research data product is not an easy task. Data managers and researchers face many choices about which metadata content is appropriate to describe their data products, and users of EML will find that it is a very flexible and extensible standard. Creating descriptive metadata content, and the nuts-and-bolts of formatting it to a standard like EML, are both important for successfully sharing research data. In this guide we strive to give the best possible advice on 1) how to create rich, descriptive metadata that follow the standards of the environmental science research community, and 2) how and where to include these metadata elements in EML documents.

2.1 History

The first version of the EML standard was written by Matthew Jones at the National Center for Ecological Analysis and Synthesis (NCEAS) and released in 1997. The standard was internally developed at NCEAS until EML version 2.0 was released as a community-maintained, open specification with substantial contributions from the LTER network. This standard was adopted by the LTER Network as the network's metadata exchange format soon after.

Because the EML standard is highly flexible and scalable, it soon became clear that establishing best practices for using EML to describe environmental data would be beneficial. The first version of this best practices guide was written as a collaborative effort between LTER Information Managers and the LTER Network Office, and released in 2004. The document was revised in 2011 and 2017, each time with contributions from a growing community of data managers and researchers collaborating in working groups and workshops. EML has been widely used for several years with multiple applications written against it, and the community has had the opportunity to observe the consequences of many EML design patterns. As much as possible, recommendations in this document have been aligned with those experiences, as well as with the capability of data contributors.

Figure 1: Ecological Metadata Language timeline and previous revisions of the EML Best Practices document.

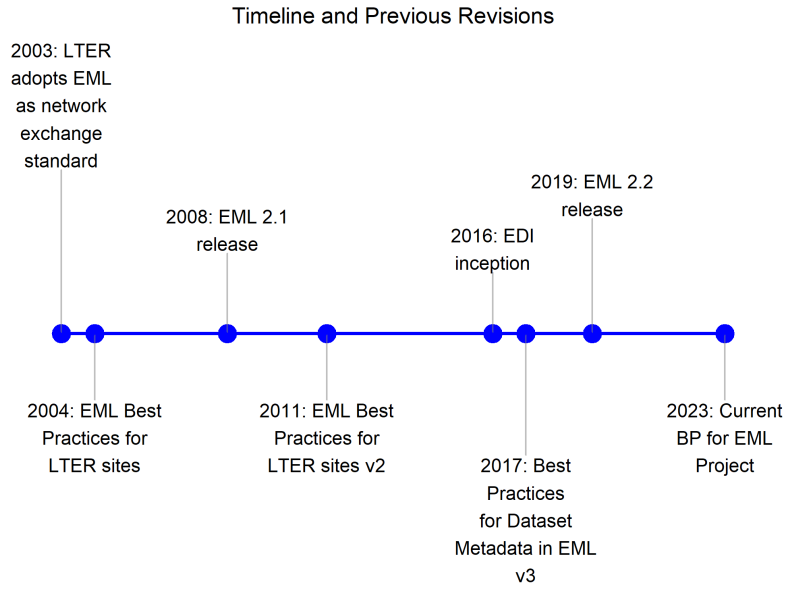

Figure 2.1: An EML event timeline drawing

The current document is the fourth version of the EML Best Practices. Many contributed to earlier versions of this document, including LTER Information Managers, personnel from the Environmental Data Initiative data repository (EDI) and NCEAS, and others. We appreciate these contributions and acknowledge that the current document is built on the work, research, and hard-won experience of many who came before.

2.2 XML, schemas, and EML

As a dialect of XML, EML documents are encoded in a text markup language that is both machine and human readable. All XML documents have a hierarchical, or tree-like, structure defined by markers called *tags*, which are text names enclosed in angle brackets (like <this>). Tags must be paired into opening and closing tags, with closing tags having the name prefixed with a forward-slash (like </this>). Information content is placed between the opening and closing tags to form an *element*, which is the most basic unit of information in an XML (or EML) document. Elements may be nested within other elements to form the tree-like structure characteristic of XML. Nested elements are often referred to using inheritance terminology, where a "child" element is nested within the "parent" element. From here forward in this document, we will commonly refer to elements using their starting tag in boldface type, like <**this**>.

To make XML documents useful and understandable for particular applications, a set of rules and definitions can be defined in an XML *schema*. The EML standard is based on a community-developed XML schema for storing scientific metadata about environmental data. There are many rules in the EML schema regarding what tags are allowed, how elements should be nested, and what content elements may contain. If an EML document doesn't break any of those rules it is said to be schema-*valid* EML. Most EML documents have a basic structure like that shown in Example 1.1, below.

Example 1.1: An abbreviated example of a valid EML document, including a declaration, the required EML root, and dataset elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<eml:eml xsi:schemaLocation="https://eml.ecoinformatics.org/eml-2.2.0
 ↪  https://eml.ecoinformatics.org/eml-2.2.0/eml.xsd" packageId="edi.1001.1"
 ↪  system="edi">
  <dataset>
    <title>My first EML dataset, created in 2024</title>
    <creator>
      …
    </creator>
    …
  </dataset>
```

```
<eml:eml>
```

Example 1.1 shows a few important features of an EML. The XML *declaration* on the first line is required to provide metadata about the XML document. It looks like an element but is not. The <**eml:eml**> element is the *root* element of the tree, and the starting tag of this element has three required *attributes*. Attributes are key:value statements used to provide additional information about an element, and they can serve a variety of purposes. The "xsi:schemaLocation" attribute within root element (<**eml:eml**>) start tag, for example, provides the online location of the EML schema that this document should match to be valid as two web addresses (each as a URL, or Uniform Resource Locator). Every EML root element must have one of several possible child elements. In this best practices document we focus on using EML to describe research datasets, so the <**dataset**> element is the required child in this case. Two child elements of <**dataset**> are shown, but this is an incomplete EML file and usually there are many more.

When constructing EML documents or evaluating them, it is a frequent necessity to check whether they are valid according to the EML schema. The EML development project maintains a summary of EML validation rules (EML schema documentation, section 6), and an online EML parser and validator (link). For additional validation methods, and further information about XML, the EML schema, and constructing valid EML documents, please see Appendix B. The rest of this best practices document focuses mainly on the content of EML documents.

2.3 What's new?

Recommendations and best practices for preparing datasets in EML evolve with each new edition of this document. In this update to the *EML Best Practices* guide, one of the main things the authors focused on was to provide guidance for the new features of the EML schema introduced in version 2.2. This was a significant change to the EML schema, and more details on changes is provided in the schema documentation (https://eml.ecoinformatics.org/whats-new-in-eml-2-2-0). Some specific changes to guidance in this document are below.

1. The responsible party recommendations in Chapter 4, especially those regarding the use of ORCID and other identifiers, have been expanded.
2. This version provides richer guidance about the <**project**> tree (Chapter 5), including prioritizing the <**award**> and <**funderIdentifier**> child elements.
3. Guidance for using semantic annotation, a new feature of EML 2.2 has been included in Chapters 7 & 11.
4. The use of new citation and reference elements introduced in the `eml-literature` module for version 2.2 (<**literatureCited**>, <**referencePublication**> & <**usageCitation**>), is described in Chapter 8.

5. The new document prioritizes recommendations for the most common data entities - <**dataTable**> and <**otherEntity**>. Recommendations for more specialized entities and data types are now covered in Appendix 1 or the *Dataset Preparation Guide for Special Cases*.
6. Significant new features and recommendations were also provided for access and usage rights (<**licensing**>, <**intellectualRights**>), <**methods**>, and <**geographicCoverage**> elements (Chapters 5, 7, and 9, respectively).

2.4 Conventions and definitions

2.4.1 Audience

This document is intended for people in a research data management role. This can include researchers managing their own data, or professional data managers doing so for a research group or organization. It assumes that readers are familiar with

1. the general purpose and content of scientific research datasets, including the data themselves and metadata describing them.
2. markup languages like HTML or XML. EML documents are written using XML.
3. the process for contributing data to a repository. If you reached this document from a repository's help page, contact them for more information.

2.4.2 Text styles and fonts

Font and typeface conventions are used throughout this guide when referring to the XML used in actual EML documents. References to XML tags, attributes, declarations, comments and other XML markup will be presented in boldface, with their surrounding angle brackets, as they would appear in valid XML. For example, the start tag of a "data table" EML element would appear as <**dataTable**>. This document also uses `XPath` expressions to indicate the location of elements or attributes within a hierarchical EML document. These will also be in boldface type with document nodes separated by forward slashes (/) and attributes prefixed with the @ symbol. For example, the `XPath` expression for the "packageId" attribute of an EML document would appear as **/eml:eml/@packageId**. When referring to the attribute alone, this document uses boldface type with the @ symbol, like @**this**.

Numbered examples of XML elements and schema-valid EML are used throughout this guide and have captions numbering them sequentially within chapters. All example XML snippets are presented in a fixed-width font with syntax highlighting and a colored background box, as in Example 1.1. Any chapters describing EML elements and their use will have a highlighted section at the bottom of the chapter listing the "XPaths referenced in this chapter". More

extensive details and definitions for XML markup language (tags, elements, etc.) and `XPath` expressions are given in Appendix B.

2.4.3 Context notes

Some recommendations have special context, e.g., an XML element or attribute may be requested by a community (e.g., LTER), or required by specific data repositories (e.g., EDI). Recommendations for EML usage in a specific context are called "context notes," and are placed in green callout boxes, as below.

> EDI context note
>
> This is an example of a context note about EDI

2.4.4 Definitions

Data contributor: the person or organization that collected the data and makes it available for publication. Data contributors can be individual researchers or research organizations (such as an LTER site).

EML preparer: the person responsible for "building" the EML metadata record. Generally, this is a data manager working with a project or research site that produces data.

Dataset: the EML metadata together with its data entity or entities. This is generally the unit housed in repositories. In the context of a repository like EDI, the term "data package" may be used instead to avoid confusion with the EML element <**dataset**>. The two terms are generally used interchangeably.

2.5 Research metadata and EML resources

As noted above, the EML standard is highly flexible, and there is no one way to create an EML document. Below are some resources that describe general best practices for research metadata, links to the EML standard itself, and tools for EML creation. Many of these resources are referenced in upcoming chapters.

2.5.1 General resources for creating scientific metadata

The practice of scientific metadata creation has a long history. Some of the foundational work outlining what metadata for the environmental sciences should contain, and how it can be used, are found in the works of William Michener and others, including the book "Environmental information management and analysis: Ecosystem to global scales" or the article "NONGEOSPATIAL METADATA FOR THE ECOLOGICAL SCIENCES." The EDI repository maintains this best-practices guide, and many other useful resources and tutorials on their website - https://edirepository.org. None of these are required reading for creating EML, but reading on the larger topic may be beneficial to some EML users and data managers. Chapter 2 of this guide also gives an overview of high priority elements of metadata for publishing research datasets.

2.5.2 The EML standard

The EML standard is developed and maintained by NCEAS, with contributions from the LTER network, ecoinformatics, and environmental research communities. Important EML resources are:

1. The current EML standard. The best practices document you are reading was written for EML version 2.2.0. The developers of the EML standard also maintain a schema browser, a GitHub repository, and other resources for community users and contributors.
2. Earlier versions of the EML standard
3. The committee and the paper that inspired the EML standard

2.5.3 Software tools for creating EML

The EML standard is under periodic development and is widely used for publishing environmental data, and consequently, a number of projects have developed software to create EML documents. Some that may be useful for EML creation in certain contexts, and for related tasks like schema validation, are listed below.

- ezEML - A web-based EML creation tool created and maintained by the EDI repository. The ezEML prompts and documentation are also an excellent resource for current best practices in preparing EML documents.
- EML - An R package for constructing EML documents
- EMLassemblyline - An R package for creating EML from metadata template files. It is a wrapper to the EML R package (above).
- metapype-eml - A Python package for constructing EML. The ezEML tool at the EDI repository is built with this library.
- LTER-core-metabase - A relational database schema for research groups managing large volumes of EML metadata. Currently implemented in PostgreSQL.

- MetaEgress - An R package for creating EML from an LTER-core-metabase instance
- MetaShark - An R shiny application for preparing EML documents - based on EMLassemblyline.

3 Metadata and dataset design principles

Key points

- There are some high-priority metadata elements that should be included in EML metadata, including titles, keywords, abstracts, coverages, methods and attributes.
- Repeatable elements can be placed at many levels in an EML document, but the recommendation is to place them at the highest level whenever possible (often **dataset**).

3.1 Definition and Structure of a Dataset

Datasets (or data packages) consist of two major components: the data and the metadata which describes that data. Metadata helps meet several needs. First, it enables data discovery by providing searchable content which allows researchers to find data based on creators, title, keywords, and geographical, temporal and taxonomic information. Second, it can provide information required to assess fitness-for-use. This includes information on the details of the data, including an abstract or description, what was measured, units, and methods. Third, metadata provides detailed information such as data format descriptions and methods that allow a researcher to actually make use of data. Last but not least, it provides unique, persistent identification of the data (e.g., a Digital Object Identifier, or DOI) that can be used for accurate data citation, therefore allowing proper attribution and credit to the data contributor when data are re-used.

Good metadata meets all those needs and converts obscure sets of numbers into meaningful data that are ready for use.

There is no perfect dataset. When designing a dataset, it is important to consider multiple factors in order to satisfy local or community research needs as well as reusability for new research. Sometimes these needs conflict, requiring compromises. Considerations such as whether data from a study should be published as a single package or in a set of related packages must be made to ultimately create a dataset that is optimized for discovery and reuse. Here are links to documents that summarize general considerations and also special considerations that may be associated with specific types of data:

- General considerations
- Special cases (e.g., spatial data, model outputs, images, etc.)

3.2 High-priority Metadata

In recent years community standards for metadata have evolved to best provide for data Findability, Accessibility, Interoperability, and Reusability (FAIR data, Wilkinson et al. 2016). General community recommendations for FAIR data have been developed (Bahim et al. 2020), along with more specific guides to important metadata elements in EML (Jones et al. 2019). Even more recently, artificial intelligence (AI) readiness requirements for data, which relate to FAIR criteria, are being identified (ESIP 2022), and new metadata authoring systems are focusing on meeting these new standards. Some data repositories have also adopted these standards and implemented specific metadata requirements for contributed datasets.

Metadata to describe scientific data can be very extensive and it can be difficult to know where to begin when assembling a dataset. Community standards for metadata have converged on some recommended areas of metadata to prioritize, shown below.

- **Citation metadata** for a dataset include the names of the data contributors (creators or authors), a descriptive title, the publication date, the dataset publisher (usually an online repository), and a unique identifier (usually a DOI provided by the repository). Data contributors should be identified with an appropriate, community accepted identifier (e.g. ORCID for individuals, ROR for organizations).
- **Titles** should be meaningful and include elements that make them understandable outside the local context. For example a title of "Primary production in a coastal stream in Northampton County, Virginia, 2020" provides thematic, location and temporal context and is much preferred over a title that fails to provide such context such as "Stream primary production" or even worse simply "Primary production."
- **Keywords** should be comprehensive and include higher level concepts applicable to the data in the areas of what ecosystem it was measured in, what was measured, methods used, and organism groups.
- An **abstract** is usually encountered early during evaluation of a dataset and should contain enough information to allow potential data users to understand the data and decide on its fitness for use. The abstract should be about the data, not the paper the data were used in.
- **Coverage** metadata, including the time periods and geographic location of data collection, and any taxonomic groups sampled or observed, will greatly improve discoverability for any environmental dataset.
- The **methods** for generating the data are critical information for potential users to evaluate the relevance and usability of the dataset for their research or synthesis projects. Methods metadata should include detailed descriptions of sampling and experimental

design, data collection procedures, quality assurance and control procedures, and any analysis or post-processing as they apply to interpreting and using the data.
- The **attributes of the data** for each data object included in a dataset, usually files of some kind (each referred to as a "data entity" in EML), should also be provided. For table data (e.g. tabular data in delimited text files) these attributes should include column header names, categorical data codes used, measurement units for each variable, and other information. For other, non-tabular data types (imagery, audio, unstructured documents, etc.), similar attributes may apply, but there will be variations depending on the object.

In the chapters ahead, each of these categories of metadata is described in detail, with particular attention to how they should be included in a valid EML document. This is, however, not an exhaustive list, and this best practices document contains recommendations for including many other metadata elements.

3.3 Unique Identifiers, versions, and data immutability

EML requires a package identifier attribute (**/eml:eml/@packageId**) in order to be valid. However, the format and management thereof and implementation of versioning are governed by the community and the repository. A versioning system allows for updates to datasets while at the same time guaranteeing data immutability, i.e., all older versions of data are unchanged and publicly available. It is the responsibility of the submitters to understand the practices of their intended repository when assigning identifiers. A repository should be able to mint and manage a Digital Object Identifier (DOI or equivalent) in addition to the EML package ID.

> EDI context note
>
> In the EDI repository, a packageId is constructed with three parts, the scope, a scope-specific identification number or accession number, and the version number. For example, the packageId "edi.20.3" has a scope of "edi," an accession number of 20 and a version number of 3.

3.4 Semantic annotations

EML 2.2 and above supports semantic annotations. These are elements that provide additional information, in the form of unique identifiers linking to online ontologies or other resources relevant to the element they are associated with. An <**annotation**> element can be added to each dataset, data entity, and attribute element, or to a list of annotations (<**annotations**> element), or even to the <**additionalMetadata**> elements (not recommended). For detailed discussion see Chapter 7.

3.5 Markdown support

EML 2.2 and above supports <**markdown**> child elements within <**abstract**> .and <**methods**> elements to allow for better formatting of content for human readability. For more information see example 9.2 (Chapter 9 - Methods), the update in the EML schema documentation (What's New in EML 2.2.0), and Appendix B.

3.6 Repeatable Elements

Several EML elements, for example methods, coverage, citations and responsible parties are flexible in that they can be used within several different levels of an EML document. For example, <**methods**> could be optionally be used to describe a dataset, a data entity, and a data attribute in an EML document (e.g., as a child element of <**dataset**>, <**dataTable**>, or <**attribute**>), representing the overall methods for the dataset, methods specific to a given entity, or methods describing how data was collected for a given variable (attribute). Similarly, <**coverage**> elements can be used at the dataset, methods, data entity, and attribute levels to document locations, dates/times and taxonomic composition at each of those levels.

The general best practice for using these repeatable, or multi-level, elements is to use them only at the "highest" level of an EML document (typically as part of a <**dataset**> element) and not at more deeply nested levels (e.g., as part of data entities or attributes). The rationale for this recommendation is that these elements will be most visible to data users, and most available to support data searches, at the dataset level. If there are application- or repository-specific use-cases for placing these elements in more deeply-nested positions then this recommendation may be relaxed, but currently such use-cases are rare. See Chapters 7 and 9 for more discussion of this subject with respect to the <**coverage**> and <**methods**> elements.

3.7 Connecting data and research

Creating reference and attribution links between datasets and other research publications is important for data providers, data users, and publishers (repositories, journals, etc.), and there are many use-cases for such links. Datasets used to generate research results should be properly cited in the resulting publications (e.g. journal articles) to allow attribution and reproducibility. Authors of datasets should appropriately cite literature used in generating their published data (such as published procedures or analytical methods). Researchers are increasingly using published dataset descriptor articles (or "data papers") to release and publicize high value datasets in repositories. Finally, when creating a dataset derived from other data sources, it is important to describe the data provenance by referencing those sources accurately. Each of these use cases has an implementation in EML for appropriate referencing or linking. See Chapter 8 for various methods of literature citation, and Chapter 9 for describing data provenance.

4 General dataset information

Key points

- Titles should be meaningful in a global context.
- Dataset abstracts should focus on the data, not research results.

The elements below are required or recommended to provide general information about the dataset and the resources it contains. All of these will be most useful when placed as direct child elements of <**dataset**>.

4.1 Dataset title

The <**title**> element contains the title for the dataset, which should be concise but descriptive enough to help a potential data user decide whether the dataset may be fit for their use. The dataset <**title**> element is typically used in full-text searches, so to facilitate discovery it should identify the data collected, the geographic context or research site, and the time frame (what, where, and when). Dataset titles should be meaningful in a global context, so avoid short, overly general titles like "Biomass Data" in favor of a more useful title like "Plant Biomass in Marshes on Hog Island, VA, USA, 1994-2003". Avoid titles that are excessively long and include too much detail on specific attributes, locations or dates, and note that titles may need to be updated if they do feature changeable metadata (such as data collection dates). Also keep in mind that dataset titles are distinct from manuscript or journal article titles, though it may sometimes be useful to refer to a related publication in a dataset title.

4.2 Abstract

The <**abstract**> element should contain the dataset abstract, which is a brief, clearly written overview of the dataset. The <**abstract**> element is used in full-text searches, so its contents should be rich with descriptive text that expands on the "what," "when," and "where" information introduced in the dataset title. Other useful abstract text can include taxonomic information, general methods descriptions, and a listing of measured variables in the dataset. The abstract is also a good place to include useful information that does not fit, or is not

searchable, in other parts of EML's structured metadata. For example, if a repository search system does not index EML maintenance information (described below), including a line describing whether the dataset is "ongoing" or "complete" would allow users to search for datasets being actively updated. Before placing search terms in the abstract, be sure to consider whether there are other searchable and more standardizable places to insert them in EML, such as in keyword elements.

As a general rule, make the abstract easy to understand by limiting technical jargon and spelling out acronyms. For datasets with a large number of variables consider using descriptive categories instead of listing all variables by name (e.g. use the term "nutrients" instead of nitrate, phosphate, calcium, etc.), unless the variable names are particularly relevant for searches. Note that <**abstract**> elements can also appear in the <**project**> tree (see Chapter 5) and are a **TextType** element in EML (see Appendix B), so abstract text will generally be formatted within <**para**> tag pairs, and sometimes <**section**> or <**markdown**> tags.

The abstract for a dataset is not the same as for a journal article, report, or other publications that may be associated with the data. Do not copy a manuscript abstract into the dataset <**abstract**> element and instead briefly describe the dataset itself following the guidance above. In some cases using the dataset <**abstract**> to describe the purpose of the data or the intent of data collection ("Why") is useful to dataset users, but avoid describing research results or interpreting the data.

4.3 Publication date (<pubDate>)

The date of public release of the dataset online should be placed in the <**pubDate**> element. This element is commonly used for constructing citations, so the <**pubDate**> value should be updated when the dataset receives significant metadata or data revisions or additions (e.g., corrected data, or additions to an ongoing time series). New, published versions of a dataset, especially when issued a new DOI, should always have <**pubDate**> updated to the current date, but note that dataset versioning practices vary by repository system.

> EDI context note
>
> When submitting to the EDI repository, the <**pubDate**> element for the dataset is automatically populated with the date the dataset is published and receives a DOI.

Example 3.1: An example of useful child elements to <**dataset**> describing a dataset from the Fictional Research Site (FRS). Note the <**para**> tags for formatting text in the <**abstract**> element (a TextType).

```
<dataset id="FRS-1" system="FRS" scope="system">
  <alternateIdentifier>FRS-1</alternateIdentifier>
```

```
  <shortName>FRS Arthropods</shortName>
  <title>Long-term Ground Arthropod Monitoring Dataset at Fictitious Research
↪  Site, USA from 1998 to 2003</title>
  <abstract>
    <para>
      This dataset contains ground arthropod weights and other measures
      collected at the Fictitious Research Site between 1998 to 2003.
      Arthropods, mainly spiders, lobsters, and trilobites, were captured at
      27 sampling locations using pitfall traps. Captured individuals were
      measured and then released unharmed. Variables for each capture record
      in the data table include weight (g), body length (mm), and number of
      leg pairs (n). This study is complete.
    </para>
  </abstract>
  <pubDate>2004-12-25</pubDate>
  …
</dataset>
```

4.4 Maintenance information (<maintenance>)

The <**maintenance**> element is used to describe how a dataset will be updated and maintained once published, and can optionally be used to document specific changes to included data tables or metadata over time. Several child elements within <**maintenance**> are useful for these tasks. The <**description**> child element can be used to enter free text about the data collection schedule or dataset updates and maintenance. It can contain both formatted and unformatted text blocks (TextType). It was once common practice to add the search terms "ongoing" or "complete" here to indicate whether new data will be added to the dataset in the future. Because the maintenance element is not usually indexed by repository search tools, it is now recommended to place "ongoing" or "complete" in the dataset keywords or abstract if discovery with those terms is desired. Be aware that the "ongoing" term needs to be removed when data collection is complete, which is an easy step to overlook. The <**maintenanceUpdateFrequency**> child element should be used to indicate the frequency of planned updates to the dataset. This element has a controlled vocabulary (see the EML schema) that contains commonly recognized frequency vocabulary words such as annually, monthly, asNeeded, notPlanned, and others.

Example 3.2: A <**maintenance**> element providing a detailed description and planned update frequency in <**maintenanceUpdateFrequency**>. Here, <**description**> is an unformatted TextType element (no <**para**> or other formatting tags).

```
<maintenance>
  <description>This is an ongoing dataset with data collected each summer
 ↪  field season (May-Sept). New observations will be appended to the dataset
 ↪  by the end of each calendar year.</description>
  <maintenanceUpdateFrequency>annually</maintenanceUpdateFrequency>
</maintenance>
```

The EML schema also provides the <**changeHistory**> element and its child elements (<**changeDate**>, <**changeScope**>, <**comment**>, <**oldValue**>) that can be used to construct a version history (i.e. change log) for the dataset and its associated entities. There are no agreed upon standards for using the <**maintenance**> element for this purpose, so best practices may be decided by individual data contributors, data managers, sites, or research networks. Whatever approach is used, including three important pieces of information is generally recommended.

1. Give a summary of the change.
2. Indicate whether the change occurred in the data, metadata, or both.
3. Describe when the change occurred using dataset version numbers or timestamps.

This information can be provided using a series of <**changeHistory**> elements (BLE LTER example) or formatted text in the maintenance <**description**> element (FCE LTER example).

4.5 Alternate identifier (<alternateIdentifier>)

The contributing organization's local data set identifier should be listed as the EML <**alternateIdentifier**> whenever this value differs from the "**packageId**" attribute in the <**eml:eml**> element. The <**alternateIdentifier**> element can also be used to denote that a package belongs to more than one contributing organization by including each individual organization's ID as a separate <**alternateIdentifier**> element. At the entity level, the <**alternateIdentifier**> should contain an alternate name for the data table or other entity itself (see Chapter 10).

> EDI context note
>
> When submitting to the EDI repository, an <**alternateIdentifier**> element will be added to EML containing the DOI that the dataset is published under.

4.6 Short name (<shortName>)

The <**shortName**> element should contain an abbreviation or shortened name for the dataset. There are no generally accepted best practices for the content of this element other than that it should be shorter than the dataset <**title**> element. Whatever shortened dataset naming scheme is acceptable and useful to data managers, users, sites, or research networks can be used here.

4.7 XPaths referenced in this chapter

Dataset title element: **/eml:eml/dataset/title**

Dataset abstract: **/eml:eml/dataset/abstract**

Project tree abstract: **/eml:eml/dataset/project/abstract**

Publication date: **/eml:eml/dataset/pubDate**

Maintenance: **/eml:eml/dataset/maintenance**

Alternate identifier: **/eml:eml/dataset/alternateIdentifier**

Short name: **/eml:eml/dataset/shortName**

5 People and organizations (ResponsibleParty)

Key points

- **userId** elements can reduce the need to duplicate detailed contact information within each **ResponsibleParty** element.
- Position-centric email addresses, rather than individual addresses, are recommended for **ResponsibleParty** elements where the individual filling a role may change over time, such as a data manager.
- Use of individual (ORCID) and organizational (ROR) registries is strongly encouraged.

The EML schema provides several elements to describe the parties responsible for a dataset. Responsible party elements may refer to either individual people, organizations, or positions within an organization, and all are derived from the **ResponsibleParty** EML type (see Appendix B). Examples include <**creator**>, <**contact**>, <**associatedParty**>, <**metadataProvider**> and <**publisher**>. Responsible party elements are typically used multiple times, and often in multiple places, in an EML document. Note that in this chapter we focus on <**dataset**> level responsible parties, but also see the <**project**> tree for similar recommendations there (Chapter 5). Any **ResponsibleParty** element must be further described using one or more of the <**individualName**>, <**positionName**>, or <**organizationName**> elements, and may include <**userId**>, <**address**>, <**electronicMailAddress**>, and <**phone**> child elements. Recommendations for populating these child elements are described first since they apply generally to any of the specific **ResponsibleParty** elements that follow.

5.1 ResponsibleParty child elements

When a **ResponsibleParty** element describes a person, the <**individualName**> element should be filled with the person's full name. An <**individualName**> element should contain both the <**surName**> (i.e. family name) and <**givenName**> child elements whenever possible. Multiple <**givenName**> elements should be used for additional given names (such as middle name) and should appear in the person's preferred order. The <**salutation**> element is for salutations that come before a name (e.g., Dr.). If the person uses a name suffix it should be added after the name within the <**surName**> element. Be aware that

conventions on the ordering of given names and surnames vary with culture and that the content of <**surName**> is usually used for alphabetical sorting. To further describe an individual person, consider including an <**organizationName**> element with the person's institutional affiliation or employer, and contact information in <**address**>, <**phone**>, and <**electronicMailAddress**> elements, though these can be left out when using the <**userId**> child element as described below. When there is any doubt about content or placement of information within these child elements the best practice is to ask the individual for their preference.

If the **ResponsibleParty** element is describing an organization instead of an individual person, no <**individualName**> is needed and <**organizationName**> should be included instead. Use a clear and complete value for the organization's name, spelling out any acronyms and limiting abbreviations. Also consider including contact information in <**address**>, <**phone**>, <**onlineUrl**>, and <**electronicMailAddress**> elements, but again, use of the <**userId**> child element may obviate this need (see below).

In some cases **ResponsibleParty** elements should describe an organization's personnel positions (such as a research site data manager) without referencing the individual person holding the position. To do this, leave out <**individualName**> and include <**positionName**> and <**organizationName**>. Referencing such positions and using position-based email addresses (e.g. data.manager@myuniversity.edu) in the <**electronicMailAddress**> element can reduce record keeping overhead as personnel change over time. See the section for the <**contact**> party below, and Example 4.4.

5.1.1 Making use of <userId>

As indicated above, **ResponsibleParty** elements can hold a great deal of information about people and organizations that contribute to published datasets. Unfortunately, it can become difficult to maintain personnel information when curating datasets for the long term. For example, it may be impractical for data managers to keep dataset contact information up-to-date as students involved in a large research project move to new institutions and positions over time. With the advent of unique identifier registries for individual researchers and research organizations, unique identifiers can be used to reference metadata in an independent, centralized database. The <**userId**> child element is intended to hold such identifiers for **ResponsibleParty** elements and it is strongly recommended to include person or organization identifiers in this way whenever possible. Doing so makes it unnecessary to populate the <**address**>, <**phone**>, <**organizationName**>, and <**onlineURL**> elements if that information is current on the target database of the ID in a <**userId**> element, and it reduces the burden of metadata maintenance over time.

For each **ResponsibleParty** element describing an individual (containing <**individualName**>), use identifiers from the Open Researcher and Contributor ID registry (ORCID) to uniquely identify individuals in <**userId**>. ORCID identifiers are quick and easy to obtain at

https://orcid.org so dataset contributors should be encouraged to register if they don't yet have one. For party elements describing organizations (containing <**organizationName**>, but not <**individualName**> or <**positionName**>), there are multiple possible identifier registries, but we recommend using the Research Organization Registry (ROR). If the organization does not have an ROR ID, the ISNI or Wikidata identifier are acceptable options that may already exist. If none of these organizational identifiers already exist, it is still worth the effort to register one, though the process may take slightly more planning. Position-based party elements (containing <**positionName**> and <**organizationName**>, but not <**individualName**>) should not contain <**userId**> elements.

When placing the identifier in the <**userId**> element, use a resolvable URI, e.g., https://orcid.org/0000-0000-0000-0000 for an ORCID and https://ror.org/021nxhr62 for an ROR ID. The opening <**userId**> tag should also contain the @**directory** attribute with a value of the registry URI, e.g., <**userId directory="https://orcid.org"**>. When providing both person and organization identifiers (ORCID and ROR), do not mix them by creating **ResponsibleParty** elements with more than one <**userId**>, and instead use two separate elements. For example, place a researcher's ORCID in the <**creator**> element, and their organization's ROR in the <**metadataProvider**> element (since metadata is typically managed at the organization level).

5.2 Dataset authors (<creator>)

The <**creator**> element is used to identify the authors of a dataset, i.e. the person(s) who provided intellectual input into creating the dataset. Data repositories such as EDI typically list the creators as the dataset authors in the citation. At least one <**creator**> is required in every EML document, but multiple are allowed and should be entered in EML in the authorship order preferred in the citation. While authorship policies are beyond the scope of this document, we recommend that programs or projects that are publishing data should develop clear authorship policies that outline which contributions and roles merit inclusion in dataset author lists. Appropriate roles to include in dataset citations may include investigators, students, personnel involved in data collection or quality assurance (technicians, data managers) and others. Many research-oriented organizations have developed policies that can be used as guidance in this regard (see NCEAS and VCR-LTER policies, for example). Once a policy is decided, consider adding <**creator**> elements as needed to achieve the desired citation as generated by the chosen repository.

A <**creator**> element's <**individualName**> (including <**surName**> and <**givenName**>) child element is used to build citations and each should therefore be completed fully, preferably with input from the person themselves, to allow proper attribution. The <**positionName**> element can be used to further define a person's role authoring the dataset, for example by entering "Principal investigator" for a scientist that initiated the dataset. Some data contributors prefer to have the research site or organization appear as an author. To do this,

include a <**creator**> element populated with research organization or site information using the <**organizationName**> element (and leaving out <**individualName**>). The <**userId**> element is useful in <**creator**> elements, so include ORCID or ROR identifiers as appropriate whenever possible. Keep in mind that stewardship of long-term datasets may change over time, and searchers frequently default to searching by author last name. Therefore it is a reasonable practice to add to the list of <**creator**> elements over time, including more authors rather than fewer, even if it blurs the credit for long-term data.

Example 4.1: An example using the <**creator**> **ResponsibleParty** element to describe an author of a dataset. Note the <**positionName**>, which is not required but may be helpful, and the ORCID identifier.

```
<creator>
  <individualName>
    <givenName>Jane</givenName>
    <surName>McInvestigator</surName>
  </individualName>
  <electronicMailAddress>mcinvestigator@ficstate.edu</electronicMailAddress>
  <userId directory="https://orcid.org">
    https://orcid.org/0000-0000-0000-0000
  </userId>
  <positionName>Principal investigator</positionName>
</creator>
```

5.3 Metadata providers (<metadataProvider>)

The <**metadataProvider**> element lists the person or organization responsible for providing the metadata to describe the published dataset. This can be a person, but many research sites, organizations, or networks derive metadata from multiple sources, so it is common to place the organization (e.g., an LTER site) under <**metadataProvider**> instead. Note that in many cases the person or organization in <**metadataProvider**> is not one of those appearing in the dataset's <**creator**> or <**associatedParty**> elements.

Example 4.2: A <**metadataProvider**> element used to describe an organization that produced the metadata for a dataset. Note the ROR identifier in <**userId**>.

```
<metadataProvider>
  <organizationName>Fictitious Research Site</organizationName>
  <electronicMailAddress>frs@ficstate.edu</electronicMailAddress>
  <onlineUrl>http://frs.ficstate.edu/</onlineUrl>
  <userId directory="https://ror.org">
    https://ror.org/12345xy67
```

```
  </userId>
</metadataProvider>
```

5.4 Associated parties (<associatedParty>)

Use <**associatedParty**> elements to list other people who were involved with the data in some way (field technicians or assistants, data analysts, etc.). All <**associatedParty**> elements require a <**role**> child element that defines the party's role with respect to the dataset. There are suggested role terms included in the EML schema ("technician", "editor", "originator", "principalInvestigator", etc.; documentation here) but there are no restrictions on using other terms. Parent institutions or named research organizations could be listed in <**associatedParty**> using a <**role**> of "owner", "originator", or similar, for example. Like dataset creators, the number of <**associatedParty**> elements may increase over time.

Example 4.3: An <**associatedParty**> element for a technician. Note that the <**role**> child element is required for <**associatedParty**>.

```
<associatedParty>
  <individualName>
    <givenName>Ima</givenName>
    <surName>Tech</surName>
  </individualName>
  <electronicMailAddress>itech@ficstate.edu</electronicMailAddress>
  <userId directory="https://orcid.org">
    https://orcid.org/0000-0000-0000-0000
  </userId>
  <role>Technician</role>
</associatedParty>
```

5.5 Dataset contacts (<contact>)

A <**contact**> element is used to list designated contact persons or organizations for the dataset and is required in all EML documents. It is recommended to include a <**contact**> element populated for a data manager or similar position that is independent of the individual holding the position (e.g., an email alias such as data.manager@ficstate.edu) so that contact information remains current with personnel changes. If multiple contacts are listed (e.g., a data manager and a lead PI) all will need to be kept current. Technicians who performed the work belong under <**associatedParty**> rather than <**contact**>. It may be advisable to complete the <**address**>, <**phone**>, <**electronicMailAddress**>, and <**onlineURL**> elements for the

<**contact**> element to make multiple methods of communication available to users if they are available.

Example 4.4: The required <**contact**> **ResponsibleParty** element. This example demonstrates a position-based contact that is independent of an individual person, and full contact information. Note the lack of a <**userId**> element since these are ambiguous for position-based parties.

```
<contact>
  <positionName>Information Manager</positionName>
  <organizationName>Fictitious Research Site</organizationName>
  <address>
    <deliveryPoint>Department for Ecology</deliveryPoint>
    <deliveryPoint>Fictional State University</deliveryPoint>
    <deliveryPoint>PO Box 111111</deliveryPoint>
    <city>Ficity</city>
    <administrativeArea>FI</administrativeArea>
    <postalCode>11111-1111</postalCode>
  </address>
  <phone phonetype="voice">(999) 999-9999</phone>
  <electronicMailAddress>data.manager@ficstate.edu</electronicMailAddress>
  <onlineUrl>http://frs.ficstate.edu/im</onlineUrl>
</contact>
```

5.6 Dataset publisher (<publisher>)

This element defines the publisher of the EML dataset, and is typically used for the publisher part of the citation. The recommended content of <**publisher**> depends on how and where the dataset will be published. If the dataset is being published on a "local" system (such as a project website), the organization producing the EML metadata (e.g., an LTER site or field station) should be placed in the <**publisher**> element. Spell out the organization's name in <**organizationName**> and complete the <**address**>, <**phone**>, <**electronicMailAddress**>, and <**onlineURL**> elements. Upon publication of the dataset in a research data repository, that repository usually becomes the publisher, and this element should be populated with appropriate repository information. The repository may populate this information in the EML itself upon publication.

> EDI context note
>
> When publishing an EML document to the EDI repository, EDI will insert a <**publisher**> element upon data submission, which will overwrite any other <**publisher**> element.

Example 4.5: A <**publisher**> element containing information for the EDI repository, as it would appear after publication of the dataset.

```
<publisher>
  <organizationName>Environmental Data Initiative</organizationName>
  <electronicMailAddress>info@edirepository.org</electronicMailAddress>
  <onlineUrl>https://edirepository.org</onlineUrl>
  <userId directory="https://ror.org">
    https://ror.org/0330j0z60
  </userId>
</publisher>
```

5.7 XPaths referenced in this chapter

Dataset creator: **/eml:eml/dataset/creator**

Dataset creator user given name: **/eml:eml/dataset/creator/individualName/givenName**

Dataset creator user surname: **/eml:eml/dataset/creator/individualName/surName**

Dataset creator user ID: **/eml:eml/dataset/creator/userId**

Dataset creator userId directory attribute: **/eml:eml/dataset/creator/userId/@directory**

Metadata provider: **/eml:eml/dataset/metadataProvider**

Associated party: **/eml:eml/dataset/associatedParty**

Dataset contact: **/eml:eml/dataset/contact**

Dataset publisher: **/eml:eml/dataset/publisher**

6 The project tree

Key points

- In the project tree, focus on key descriptive child elements (project title, abstract, and key personnel), and funding information.
- Describe project funding using the **award** element.

Use the <**project**> tree to describe the project under which the dataset was created. This element is an EML ResearchProjectType (see Appendix B) and it is therefore possible to include a large range of structured child elements within a <**project**> tree. We recommend focusing on a few key descriptive child elements (project title, abstract, and key personnel), and funding information, which are described below. There are optional child elements for describing the study area and project design (<**studyAreaDescription**> and <**designDescription**>), but if those descriptions are important and can be kept brief, we recommend putting them in the project abstract instead (**/eml:eml/dataset/project/abstract**).

The <**title**> and <**personnel**> elements are required for any project tree. The <**title**> element should distinctly identify the project. If the project is supported by a grant from a funding agency (NSF, for example) the project may already have a title that can be used for this element. If not, choose something concise and meaningful to a dataset user. Personnel associated with the project, such as principal investigators, should be included in one or more <**personnel**> elements, which are **ResponsibleParty** XML types (described in Appendix B). A <**personnel**> element must contain a child <**role**> element, so include project personnel roles (principal investigator, co-PI, etc.) here. The project <**abstract**> element is a brief, clearly-written overview of the project. It may contain information about the project objectives, study design, location, and other relevant details. The project <**title**> and <**abstract**> elements are essentially the same as elements presented in Chapter 3, while the <**personnel**> element is similar to <**associatedParty**> in Chapter 4. Guidelines presented in those sections also apply here.

It is strongly recommended to describe project funding using the <**award**> element and its child elements (<**funderName**>, <**funderIdentifier**>, <**awardNumber**>, etc.). These provide a machine-interpretable description of the funding support for the project, including funding agency and grant number. For the optional <**funderIdentifier**>, use the funding agency's ROR ID if they have one. If not, see if they have a Crossref ID, or failing that, a Wikidata

ID. These identifiers should be entered in the form of a URL, e.g., https://ror.org/021nxhr62. The <**funding**> element (**/eml:eml/dataset/project/funding/**) can be used to provide a nuanced, human-friendly version of funding information. It is a **TextType** element (see Appendix B) and thus can be formatted with <**section**>, <**para**> and other formatting elements.

It is common for research activities to have multiple sources of support, either as a series of grants awarded sequentially over time, or multiple grants supporting research at the same time. To describe a project supported by a series of awards (such as an LTER site), you may include the most recent award and PIs directly within the <**project**> and previous awards and PIs as <**relatedProject**> elements (or even simply include additional <**award**> elements). For projects with multiple simultaneous funding sources, you may nest as many <**relatedProject**> elements within the <**project**> tree as needed. Any nested <**relatedProject**> elements can be populated in the same way as the parent <**project**> element, but there is no standard practice for this, so do what works for you or your research organization. Be sure to include appropriate <**funding**> and <**award**> elements in any <**project**> or <**relatedProject**> element to make clear how projects are supported.

Example 5.1: A <**project**> element describing an NSF-funded environmental monitoring study with two principal investigators. Note the <**award**>, which is a high priority for any funded project.

```
<project>
  <title>FRS basic monitoring program</title>
  <personnel>
    <individualName>
      <salutation>Dr.</salutation>
      <givenName>Eva</givenName>
      <surName>Scientist</surName>
    </individualName>
    <electronicMailAddress>eva@ficstate.edu</electronicMailAddress>
    <userId system="https://orcid.org">
      https://orcid.org/0000-0000-0000-0000
    </userId>
    <role>principalInvestigator</role>
  </personnel>
  <personnel>
    <individualName>
      <givenName>Monica</givenName>
      <surName>Techy</surName>
    </individualName>
    <electronicMailAddress>monica@ficstate.edu</electronicMailAddress>
    <userId system="https://orcid.org">
      https://orcid.org/0000-0000-0000-0000
```

```
    </userId>
    <role>principalInvestigator</role>
  </personnel>
  <abstract>
    <para>
      The FRS basic monitoring program consists of monitoring of
      arthropod populations, plant net primary productivity, and bird
      populations. Monitoring takes place at 3 locations, 4 times a year.
      Climate parameters are continuously measured at all stations.
    </para>
  </abstract>
  <funding>
    <para>
      Funding is from the National Science Foundation under award #1546024
 ↪  (2016-02-22 to 2021-01-28).
    </para>
  </funding>
  <award>
    <funderName>National Science Foundation</funderName>
    <funderIdentifier>https://ror.org/021nxhr62</funderIdentifier>
    <awardNumber>1546024</awardNumber>
    <title>Scientia Arctica: A Knowledge Archive for Discovery and
 ↪  Reproducible Science in the Arctic</title>
    <awardUrl>
https://www.nsf.gov/awardsearch/showAward?AWD_ID=1546024
    </awardUrl>
  </award>
</project>
```

6.1 XPaths referenced in this chapter

The project tree: **/eml:eml/dataset/project**

Project title: **/eml:eml/dataset/project/title**

Project abstract: **/eml:eml/dataset/project/abstract**

Project personnel: **/eml:eml/dataset/project/personnel**

Project funding: **/eml:eml/dataset/project/funding**

Funding award: **/eml:eml/dataset/project/award**

Award funder name: **/eml:eml/dataset/project/award/funderName**

Funding award title: **/eml:eml/dataset/project/award/title**

Funding award number: **/eml:eml/dataset/project/award/awardNumber**

A related project: **/eml:eml/dataset/project/relatedProject**

7 Data access and usage rights

Key points

- Online distribution is recommended.
- Open licenses with minimal restrictions on data use are recommended for most datasets.
- CC0 and CC-BY are frequently used licenses.

The primary purpose of using EML to describe a dataset is to make the data reusable. While the majority of an EML is devoted to describing the data and their origin, there are also several EML elements used to determine whether, how, and by whom the data should be used.

1. Permission to access the dataset itself is set in the <**access**> element. These permissions are decided by the data contributor and the publisher (usually a repository), and open access permissions are usually recommended. The EML <**access**> element is being deprecated in favor of repository-specific access functions, but is still useful for compatibility with some systems.
2. The method for distributing a dataset and its associated data entities is described in the <**distribution**> element. Generally datasets are shared by distribution of digital files over the internet using URLs, but there are other methods to consider.
3. Licensing of the dataset and associated usage rights (or intellectual rights) are communicated in the <**intellectualRights**> and <**licensed**> elements. These elements describe the legal requirements and other policies and expectations for use of the data once it is obtained. Open licenses with minimal restrictions on data use are usually recommended for research datasets.

These sections of EML are described in order below.

7.1 Permission to access the dataset (<access>)

An <**access**> element contains a list of rules defining access permissions for an EML document's metadata and any data files (or entities) that the metadata describes. Any access rules defined must be applicable to the system where the dataset is stored. Usually, that system is a research

data repository. See Example 6.1 below if you plan to construct your own access element. With the exception of certain sensitive information, metadata should be publicly accessible.

In EML 2.1, <**access**> trees were allowed at two places: as the first child of the <**eml:eml**> root element (at the same level as <**dataset**>) for controlling access to the entire document, and in a data entity <**distribution**> element for controlling access to the data entity. The <**access**> element is now deprecated in EML 2.2 and access control is transitioning to repository specific applications. Nevertheless, the <**access**> element is still available in EML 2.2 for backward compatibility and EML preparers should note that if <**access**> is omitted some repositories will presume that only the dataset submitter should be allowed access.

EDI context note

The EDI repository still recommends including the <**access**> element as the first child of the root (<**eml:eml**>) element and uses an access control format that conforms to the KNB system of using the LDAP "distinguishedName (dn)" for an individual, as in "uid=userID,o=EDI,dc=edirepository,dc=org." Public access is recommended, but a temporary embargo on the data entities may be requested, with certain exceptions.

Example 6.1: An <**access**> element for the EDI repository allowing full permission ("all") to the "userID" user that uploaded the dataset, and read access to all other users ("public").

```
<access authSystem="https://pasta.edirepository.org/authentication"
 ↪  order="allowFirst" scope="document"
 ↪  system="https://pasta.edirepository.org">
  <allow>
    <principal>uid=userID,o=EDI,dc=edirepository,dc=org</principal>
    <permission>all</permission>
  </allow>
  <allow>
    <principal>public</principal>
    <permission>read</permission>
  </allow>
</access>
```

7.2 Distribution of data entities (<distribution>)

The <**distribution**> element provides methods to obtain either the data entities themselves, metadata records, or further resources related to the dataset. This element can appear in many places in an EML document. At the dataset level (direct child of <**dataset**>) it would provide a method to obtain the dataset itself. At the entity level (e.g. within a <**dataTable**> element) the element provides a method to obtain the data entity files. The <**distribution**> element

has one of three children for describing the location of the resource: <**online**>, <**offline**>, and <**inline**>.

The <**online**> element describes resources distributed over the internet. Distribution via a URL is recommended and will usually be the case when publishing to a research data repository. The <**online**> element may contain two sub elements, <**url**>, populated with the URL for the resource, and <**onlineDescription**>, which describes the online resource and is optional. Any <**url**> element should have an optional **function** attribute which can be set to either "download" or "information." When **function="download"**, accessing the URL directly streams the data object for download. When **function="information"**, accessing the URL leads to a website such as a data catalog, intended-use page, or other address that provides information about downloading the object but doesn't directly return the data stream. If the **function** attribute is omitted, then "download" is implied. If a data entity can be distributed via database connection, the <**connection**> child element is also permitted as an alternative to <**url**>, though this is rarely used.

EDI context note

For an EML dataset to be accepted into the EDI repository, it must include a distribution URL at the entity level (e.g., /eml:eml/dataset/dataTable/physical/distribution/url) with a function attribute having the value "download" (or empty, which defaults to "download"). This URL may be added before uploading the EML file by adding static links manually or with ezEML, or can be added upon upload to the repository using the web-based portal interface. The EDI repository system also has alternatives for uploading data entities if you do not have a method to deliver entities via URL (http). More details are available here, or contact EDI for information.

The <**offline**> element describes resources (including data entities) that are not available online and must be distributed by other means. Often this refers to very large datasets that are best distributed on physical media like hard drives. For offline resources the <**offline**> element should contain at least the <**mediumName**> child element to describe the medium the data is distributed on (e.g. tape, hardcopy, etc.). It may also be advisable to include the <**mediumNote**> element to describe how and where resources can be obtained in the offline format.

EDI context note

The EDI repository has specific recommendations for handling large datasets and you may wish to first contact EDI support.

An <**inline**> element can contain data that is stored directly within the EML document. This data can be included as text or a string that can be parsed by the user. In general, including data in <**inline**> elements is not recommended because it makes data access more difficult for users.

At the dataset level the <**distribution**> element should be used to distribute metadata or other information, not for the download of data or related resources. Therefore, include <**url**> elements with the @**function** attribute indicating an informational resource (`function="information"`) with links to resources that provide information about the entire dataset instead of a specific data entity. URLs provided here may point back to the researcher or site's local information page or data catalog record. When uploading to a research data repository this element may be populated with a URL pointing to the dataset's online address at the repository.

> EDI context note
>
> The EDI repository will add a <**distribution**> element at the dataset level that includes the dataset's DOI, which points to the dataset landing page within the repository.

Example 6.2: A <**distribution**> element at the dataset level that refers to a research site's data catalog entry for a dataset.

```
<distribution id="frs-1" scope="system"
 ↪  system="https://frs.fictate.edu/data">
  <online>
    <onlineDescription>frs-1 Data Catalog Page</onlineDescription>
    <url function="information">
      http://frs.ficstate.edu/data/frs-1.htm
    </url>
  </online>
</distribution>
```

An entity level <**distribution**> element should contain information on how that specific data entity (e.g., data table) can be accessed. It is most common, and recommended, to make data entities available for download over the the internet with a URL. Therefore, include <**url**> elements with the "download" attribute (`function="download"`) containing links that will stream the data object for download by the user. In a research data repository the download URLs are managed by the repository itself and will usually be added upon publication of the dataset. As mentioned above, the <**connection**> element is available as an alternative to <**url**> for database connections. Note that in EML 2.1 an <**access**> element is also permitted within entity level <**distribution**> elements to control access to the entity. See the access section above for more information.

> EDI context note
>
> For all data uploaded to EDI, the data access URL is inserted during the submission process using the fully qualified data entity ID. However, for automated data submission

> to EDI, this element may be used for staging data on a local server.

Example 6.3: A <**distribution**> element at the data entity level. This is an example of how a data repository (EDI in this case) might distribute a published data entity via a download URL.

```
<dataTable>
  <physical>
    ...
    <distribution>
      <online>
        <url function="download">
↪   https://pasta.lternet.edu/package/data/eml/frs/36/5/d22ec5961958e900a65fb1d402132c89
        </url>
      </online>
    </distribution>
  </physical>
  ...
</dataTable>
```

7.3 Licensing and intellectual rights (<licensed>, <intellectualRights>)

Two elements are available to govern licensing and data usage rights: <**licensed**> and <**intellectualRights**>. It is recommended that licensing, policies, and expectations for use of published datasets be clearly articulated in these elements. In general, research datasets should be released with the fewest restrictions on use possible. Information in these elements may be scrutinized by journal reviewers and funding agencies when datasets are cited in papers or proposals.

The <**licensed**> element was added in EML 2.2.0 and provides a structured approach for including common licenses. It is strongly recommended to choose community standard licenses for published datasets and describe them in <**licensed**> using the <**licenseName**>, <**url**>, and <**identifier**> elements as shown in Example 6.5. The System Package Data Exchange standard (SPDX) provides a vocabulary of machine-interpretable license URLs, and the EML schema recommends populating the <**url**> element with values from this list (https://spdx.org/licenses/). Creative Commons (https://creativecommons.org) offers a number of well-established licenses that are commonly used and appropriate for research datasets, and these can be readily found in the SPDX list.

The <**intellectualRights**> element provides a free text field to explain data use policies. It is a **TextType** element (see Appendix B) and thus can be formatted with <**section**>, <**para**> and other formatting elements. Use this element to clearly describe the policies and expectations for use of the dataset, particularly those not already stated in the dataset license. These policies and expectations should match those agreed upon by the dataset creators and affiliated institutions, funding agencies, and/or research networks. If data access at the repository level or in <**access**> elements is different than any policy articulated here (e.g. restricted-access packages), explain why and provide expectations or a timeframe for data release.

Community context note

Dataset licensing practices are set by the community. For the LTER Network and the EDI repository, in-depth discussion of data licensing may be found at the EDI website and in the LTER Network Data Access Policy. The recommended public licenses for this community are Creative Commons open access with attribution (CC-BY) or public domain (CC0).
If no <**intellectualRights**> or <**licensed**> element is included in an EML file, EDI will insert text that releases data under "CC0". The LTER Network-wide default policy is "CC-BY." Please consult those organizations for more information.

Example 6.4: An <**intellectualRights**> element populated with text following the LTER Network data use policy. This TextType element is formatted with <**section**> and <**para**> elements.

```
<intellectualRights>
  <section>
    <title>Data Policy</title>
    <para>
      This data package is released to the "public domain" under
      Creative Commons CC0 1.0 "No Rights Reserved" (see:
      https://creativecommons.org/publicdomain/zero/1.0/). It is considered
      professional etiquette to provide attribution of the original work if
      this data package is shared in whole or by individual components. A
      generic citation is provided for this data package on the website
      https://portal.edirepository.org (herein "website") in the summary
      metadata page. Communication (and collaboration) with the creators of
      this data package is recommended to prevent duplicate research or
      publication. This data package (and its components) is made available
      "as is" and with no warranty of accuracy or fitness for use. The
      creators of this data package and the website shall not be liable for
      any damages resulting from misinterpretation or misuse of the data
      package or its components. Periodic updates of this data package may
      be available from the website. Thank you.
```

```
    </para>
  </section>
</intellectualRights>
```

Example 6.5: A <**licensed**> element populated with information for the CC-BY license.

```
<licensed>
  <licenseName>Creative Commons Attribution 4.0 International</licenseName>
  <url>https://spdx.org/licenses/CC-BY-4.0</url>
  <identifier>CC-BY-4.0</identifier>
</licensed>
```

7.4 XPaths referenced in this chapter

Dataset access (recommended by EDI): **/eml:eml/access**

Data entity access (deprecated): **/eml:eml/dataset/dataTable/physical/distribution/access**

Dataset distribution: **/eml:eml/dataset/distribution**

Dataset distribution function attrib: **/eml:eml/dataset/distribution/@function=”information”**

Entity distribution: **/eml:eml/dataset/dataTable/physical/distribution**

URL for entity distribution: **/eml:eml/dataset/dataTable/physical/distribution/online/url**

Licensing element: **/eml:eml/dataset/licensed**

License name **/eml:eml/dataset/licensed/licenseName**

Intellectual rights: **/eml:eml/dataset/intellectualRights**

8 Keywords, coverage, and annotations

Key points

- Whenever possible use keyword terms drawn from established vocabularies.
- **keywordSet** can be used to group terms drawn from a similar **keywordThesaurus**.
- **@keywordType** can be used to identify the purpose of a keyword (e.g., identifying themes or places).
- **geographicCoverage** elements are designed to facilitate data discovery, but not to fully describe spatial datasets.
- **endDate** tags should be used to identify the end date of data already in the dataset, even if future data collection is planned.
- When taxa are numerous, **taxonomicCoverage** elements with fully-populated taxonomic trees may become extremely large. If so include only the species-level names, along with a **taxonId** element.
- Annotations can provide a valuable supplement to standard EML elements, but look for advice and feedback on using semantic annotation in your research and data management communities before implementing them in EML.

8.1 Keywords (<keywordSet>, <keyword>)

Meaningful keywords are essential metadata for describing published datasets and enhancing their discoverability. In EML documents, include keywords as <**keyword**> elements nested within one or more parent <**keywordSet**> elements and it is recommended to place <**keywordSet**> elements as children of <**dataset**> (/**eml:eml/dataset/keywordSet**). Whenever possible, it is recommended to populate a <**keywordSet**> keyword terms selected from established, community-accepted controlled vocabularies (CV) or keyword lists, and then include the name or identifier of the CV or list within a <**keywordThesaurus**> child element in the same <**keywordSet**>. Using this same approach, research groups, institutions, or individual data managers may create and reference keyword thesauri specific to the subject matter of the data and their data management needs. Because <**keyword**> elements, along with <**title**> and <**abstract**> elements, are frequently accessed by full-text search tools,

following thoughtful, consistent keywording practices will help optimize the searchability of datasets.

Individual <**keyword**> elements can be categorized using the @**keywordType** attribute, which accepts a predefined list of values. For example, it is appropriate to include meaningful keywords for geographic places (e.g. state, city, county) which can be given the ”place” attribute value. The ”theme” attribute value is commonly used to describe keywords related to research themes. Other valid @**keywordType** values include “taxonomic”, “stratum”, and “temporal”. Using this attribute is optional and not all EML creation systems support it, but it is recommended.

> Community context note
>
> Communities often have specific standards for keywords to assist in searches. For example, the LTER Network requests that datasets should include keywords for LTER core research areas (e.g. primary production), the network and three-letter site acronyms (e.g. LTER, FLS), and relevant conceptual and subject-matter keywords from the LTER Controlled Vocabulary. To highlight that LTER core research area and controlled vocabulary terms are present, it may be appropriate to include clearly named <**keywordThesaurus**> elements in their <**keywordSet**>, as suggested in Example 7.1. This practice is not well standardized in LTER network datasets at this time. Conventions vary widely by community, so be aware of the best practices employed by your site, institution, data repository, or research community.

Example 7.1: Several <**keywordSet**> elements containing keywords derived from an FLS LTER site list (probably developed internally to the site), the LTER controlled vocabulary, the LTER core research areas, and the USGS Geographic Names Information System (GNIS).

```
<keywordSet>
  <keyword keywordType="theme">FLS</keyword>
  <keyword keywordType="theme">Fictitious LTER Site</keyword>
  <keyword keywordType="theme">LTER</keyword>
  <keyword keywordType="theme">Arthropods</keyword>
  <keyword keywordType="theme">Richness</keyword>
  <keywordThesaurus>FLS site thesaurus</keywordThesaurus>
</keywordSet>
<keywordSet>
  <keyword keywordType="theme">ecology</keyword>
  <keyword keywordType="theme">biodiversity</keyword>
  <keyword keywordType="theme">population dynamics</keyword>
  <keyword keywordType="theme">terrestrial</keyword>
  <keyword keywordType="theme">arthropods</keyword>
  <keyword keywordType="theme">pitfall trap</keyword>
```

```
  <keyword keywordType="theme">monitoring</keyword>
  <keyword keywordType="theme">abundance</keyword>
  <keywordThesaurus>LTER controlled vocabulary</keywordThesaurus>
</keywordSet>
<keywordSet>
  <keyword keywordType="theme">populations</keyword>
  <keywordThesaurus>LTER core research areas</keywordThesaurus>
</keywordSet>
<keywordSet>
  <keyword keywordType="place">Atigun River</keyword>
  <keyword keywordType="place">State of Alaska</keyword>
  <keywordThesaurus>Geographic Names Information System</keywordThesaurus>
<keywordSet>
```

8.2 Coverage elements (<coverage>)

The <**coverage**> element is intended to describe a dataset's coverage in terms of space, time, and taxonomy, and may therefore contain three types of child element: <**geographicCoverage**>, <**temporalCoverage**>, and <**taxonomicCoverage**>. Populating these elements as recommended enables discovery by advanced search tools, and facilitates successful interpretation and reuse of the data. This best practice recommends using <**coverage**> elements at the dataset level (**eml:eml/dataset/coverage**), but they may also be placed at methods, entity and attribute levels for some use cases.

8.2.1 Geographic Coverage (<geographicCoverage>)

The <**geographicCoverage**> element describes geographic features related to the data, such as research sites or sample locations. A <**coverage**> element can contain multiple <**geographicCoverage**> elements, each describing a geographic feature such as a point or bounding box. Include at least one <**geographicCoverage**> element per dataset, and as many as necessary to facilitate dataset discovery and evaluate fitness for use. Note, however, that this element is not designed to exhaustively describe a dataset's spatial entities. The most common features described in <**geographicCoverage**> are points and bounding boxes, but more complex spatial entities can also be defined in EML (see Appendix A).

Creating a <**geographicCoverage**> element to describe a point or bounding box requires two child elements: <**boundingCoordinates**> and <**geographicDescription**>. The <**boundingCoordinates**> element defines these geographic features by listing the coordinates (longitude or latitude) defining their western, eastern, northern, and southern limits, in that order, enclosed in appropriate child elements (e.g. <**westBoundingCoordinate**>,

<**northBoundingCoordinate**>, etc.). For bounding boxes, the north and south coordinate pair and the east and west coordinate pair will contain distinct values (Example 7.2), but for a point location these coordinate pairs must share the same value (Example 7.3). Latitudes and longitudes must be expressed in decimal degrees and should use the same datum (e.g., WGS 84). Use an appropriate number of decimal places to accurately represent a location or feature; more decimal places are appropriate for precise point locations. Longitudes west of the Prime Meridian and latitudes south of the Equator must be prefixed with a minus sign (-).

The <**geographicDescription**> element is a string type and should describe the geographic feature in a detailed, comprehensive way. Include useful geographic search terms such as country, state, county or province, city, general topography, landmarks, rivers, the datum of coordinates in <**boundingCoordinates**> (if known), and other relevant information. The method for determining coordinates and other spatial information can be included with the <**methods**> elements in the EML document.

Example 7.2: A simple bounding box <**geographicCoverage**> element. Note that the west, east, north, south ordering of coordinates is required.

```
<coverage>
  <geographicCoverage>
    <geographicDescription>
      Ficity, FI metropolitan area, USA. Coordinates based on WGS84 datum.
    </geographicDescription>
    <boundingCoordinates>
      <westBoundingCoordinate>-112.373614</westBoundingCoordinate>
      <eastBoundingCoordinate>-111.612936</eastBoundingCoordinate>
      <northBoundingCoordinate>33.708829</northBoundingCoordinate>
      <southBoundingCoordinate>33.298975</southBoundingCoordinate>
    </boundingCoordinates>
  </geographicCoverage>
</coverage>
```

Example 7.3: A <**geographicCoverage**> element for three discrete point locations. Note the identical values in the west/east and north/south bounding coordinate pairs.

```
<coverage>
  <geographicCoverage>
    <geographicDescription>site 1, Ficity, FI metropolitan area, USA.
↪  Coordinates based on WGS84 datum.</geographicDescription>
    <boundingCoordinates>
      <westBoundingCoordinate>-112.2</westBoundingCoordinate>
      <eastBoundingCoordinate>-112.2</eastBoundingCoordinate>
      <northBoundingCoordinate>33.5</northBoundingCoordinate>
```

```
      <southBoundingCoordinate>33.5</southBoundingCoordinate>
    </boundingCoordinates>
    <geographicDescription>site 2, Ficity, FI metropolitan area, USA.
 ↪  Coordinates based on WGS84 datum.</geographicDescription>
    <boundingCoordinates>
      <westBoundingCoordinate>-111.7</westBoundingCoordinate>
      <eastBoundingCoordinate>-111.7</eastBoundingCoordinate>
      <northBoundingCoordinate>33.6</northBoundingCoordinate>
      <southBoundingCoordinate>33.6</southBoundingCoordinate>
    </boundingCoordinates>
    <geographicDescription>site 3, Ficity, FI metropolitan area, USA.
 ↪  Coordinates based on WGS84 datum.</geographicDescription>
    <boundingCoordinates>
      <westBoundingCoordinate>-112.1</westBoundingCoordinate>
      <eastBoundingCoordinate>-112.1</eastBoundingCoordinate>
      <northBoundingCoordinate>33.7</northBoundingCoordinate>
      <southBoundingCoordinate>33.7</southBoundingCoordinate>
    </boundingCoordinates>
  <geographicCoverage>
</coverage>
```

Point and bounding box elements may be used together, and the content and number of <**geographicCoverage**> elements included in a dataset is at the discretion of the data contributor and EML preparer. As a sensible default, include at least one <**geographicCoverage**> element defining the maximum geographic extent represented in the data. This may be a point, as for a single weather station, or a bounding box, as for a research site or observational area. If there are significant distances between observations or study sites and grouping them into one bounding box would be misleading or confusing, include <**geographicCoverage**> elements for each nominal site or group (at your discretion). For example, a dataset for a cross-site study should probably have bounding boxes for each site, and a few widely spaced monitoring stations should not be enclosed by one very large bounding box.

Providing appropriate geographic detail is recommended, but keep in mind that the <**geographicCoverage**> element is not intended to fully describe spatial datasets. Including large numbers of <**geographicCoverage**> elements in a dataset (more than 10, say) may be impractical for data managers, and extracting useful spatial data from those elements can be onerous for users. For example, when a dataset needs to describe numerous geographic features, such as data from a regular sample grid or spatially dense sampling area, it is appropriate to describe the grid or area with a bounding box rather than a long list of points in individual <**geographicCoverage**> elements. More advanced spatial entities can also be added to a <**geographicCoverage**> using the <**datasetGPoly**> child element. This element is not used frequently but can be useful and is described in Appendix A. If spatial

location is important to use or interpret the data, the best practice is to include geographic features as a separate data entity with the dataset. When the dataset's geographic coverage is lengthy, complex and/or intended for use in GIS software, consider including the locations as tabular data (a CSV file), or in geospatial files, like a GeoPackage or KMZ file, attached as an <**otherEntity**> (see Chapter 10).

8.2.2 Temporal Coverage (<temporalCoverage>)

The <**temporalCoverage**> element represents the period(s) of time covered in the dataset. Most often this means the dates that included data were collected. If the dataset is for a study using retrospective or historical data, this element should not refer to the dates the study was conducted, but the dates represented in the data.

Either <**singleDateTime**> or <**rangeOfDates**> are required child elements, and the <**temporalCoverage**> element therefore allows for three descriptions: a single date and time (one <**singleDateTime**>), multiple dates and times (>1 <**singleDateTime**>) and a range of dates and times (<**rangeOfDates**>). A <**rangeOfDates**> element must contain valid <**beginDate**> and <**endDate**> child elements. All <**singleDateTime**>, <**beginDate**>, and <**endDate**> elements must contain either the <**calendarDate**> and optional <**time**> elements, or an <**alternativeTimeScale**>. Two formats are allowed for <**calendarDate**>, either a 4-digit year, or a date in ISO 8601 format: YYYY-MM-DD. The <**alternativeTimeScale**> is appropriate in cases where temporal descriptions such as "years before present" are used (see the schema), e.g., for long-term tree ring chronologies dating back thousands of years.

For datasets considered "ongoing," i.e., data are planned to be added at intervals, it is not valid to leave an empty <**endDate**> tag in EML. Further, EML is intended to house immutable "snapshots" of data (depending on repository support). So, for ongoing datasets the best practice is to populate <**endDate**> with the latest date in the included data file and then update the field when new data are added. Do not include <**temporalCoverage**> elements indicating times when no data are present. Use the <**maintenance**> element to describe the update frequency and dataset version history, and the <**title**>, <**abstract**>, and/or <**methods**> elements to describe plans for ongoing data collection (see Chapter 3).

Example 7.4: A simple <**temporalCoverage**> element describing a range of dates represented in a long-term dataset.

```
<temporalCoverage>
  <rangeOfDates>
    <beginDate>
      <calendarDate>1998-11-12</calendarDate>
    </beginDate>
    <endDate>
      <calendarDate>2003-12-31</calendarDate>
```

```
    </endDate>
  </rangeOfDates>
</temporalCoverage>
```

8.2.3 Taxonomic Coverage (<taxonomicCoverage>)

The <**taxonomicCoverage**> element documents taxonomic information for all organisms relevant to the study. The lowest available level, preferably the species binomial, and common name should always be included, but higher-level taxa can also be included to support broader taxonomic searches. Blocks of <**taxonomicClassification**> elements should be hierarchically nested within a single <**taxonomicCoverage**> element rather than repeated at the same level. The optional <**generalTaxonomicCoverage**> element can include general descriptions of a) the procedure for how taxonomy was determined (keys used, etc.), b) the flora/fauna included in the study (scope), and c) granularity of the taxonomy - for example, identification to family, genus, or species.

It is strongly recommended to include external taxonomic identifiers (such as from ITIS or WoRMS) for a given taxon using the <**taxonId**> element. When indicating the taxonomic provider in <**taxonId**>, use a URL for the provider, e.g., https://www.itis.gov/. It is also advisable to provide common names within <**taxonomicClassification**> elements using the <**commonName**> element if possible.

The <**taxonomicCoverage**> element can become very large when numerous taxa are described, and the EML schema allows this element to have a flexible structure. When taxa are numerous, it may be advisable not to expand the full taxonomic tree within the <**taxonomicCoverage**>, i.e. only include species level <**taxonRankName**> and <**taxonRankValue**> elements, along with a resolvable <**taxonId**> element for each (compare Examples 7.5 and 7.6). It is also allowable to combine taxonomic elements in the hierarchy under like <**taxonRankName**> elements to create a taxonomic "tree" (not illustrated), but this practice may impede combining and re-using <**taxonomicClassification**> information from multiple documents so should be considered carefully. Note that the EML schema does not prescribe or validate any particular taxonomic classification system, and can ultimately support whatever classification hierarchy a user wants. However, linking to a community-standard taxonomic system (ITIS, WoRMS, etc.) is generally recommended. In some cases, alternatives to using the <**taxonomicCoverage**> element, such as including long lists of taxa with relevant identifiers as a <**dataTable**> entity, or a taxonomic database as an <**otherEntity**>, can be considered. There are downsides to this approach, however, because data entity contents are not typically indexed by search tools, and discovery of data by taxa would therefore be limited.

The <**taxonomicCoverage**> element can include several more elements that describe taxonomic identification resources, methods and protocols used for taxonomic classification,

classification systems used, etc. (see Appendix A); however, for simplicity one should include these details in the <**methods**> element of the EML document.

Example 7.5: A <**taxonomicCoverage**> element describing two species. Note the ITIS identifiers.

```
<taxonomicCoverage>
  <generalTaxonomicCoverage>
    Mollusks were identified to species
  </generalTaxonomicCoverage>
  <taxonomicClassification>
    <taxonRankName>Kingdom</taxonRankName>
    <taxonRankValue>Animalia</taxonRankValue>
    <taxonId provider="https://www.itis.gov/">202423</taxonId>
    <taxonomicClassification>
      <taxonRankName>Phylum</taxonRankName>
      <taxonRankValue>Mollusca</taxonRankValue>
      <commonName>mollusks</commonName>
      <taxonId provider="https://www.itis.gov/">69458</taxonId>
      <taxonomicClassification>
        <taxonRankName>Class</taxonRankName>
        <taxonRankValue>Gastropoda</taxonRankValue>
        <commonName>gastropods</commonName>
        <commonName>snails</commonName>
        <taxonId provider="https://www.itis.gov/">69459</taxonId>
        <taxonomicClassification>
          <taxonRankName>Order</taxonRankName>
          <taxonRankValue>Archaeopulmonata</taxonRankValue>
          <taxonId provider="https://www.itis.gov/">78782</taxonId>
          <taxonomicClassification>
            <taxonRankName>Family</taxonRankName>
            <taxonRankValue>Ellobiidae</taxonRankValue>
            <taxonId provider="https://www.itis.gov/">76453</taxonId>
            <taxonomicClassification>
              <taxonRankName>Genus</taxonRankName>
              <taxonRankValue>Detracia</taxonRankValue>
              <taxonId provider="https://www.itis.gov/">76462</taxonId>
              <taxonomicClassification>
                <taxonRankName>Species</taxonRankName>
                <taxonRankValue>Detracia floridana</taxonRankValue>
                <commonName>florida melampus</commonName>
                <taxonId provider="https://www.itis.gov/">76463</taxonId>
              </taxonomicClassification>
            </taxonomicClassification>
```

```
        </taxonomicClassification>
      </taxonomicClassification>
    </taxonomicClassification>
  </taxonomicClassification>
</taxonomicClassification>
<taxonomicClassification>
  <taxonRankName>Kingdom</taxonRankName>
  <taxonRankValue>Animalia</taxonRankValue>
  <taxonId provider="https://www.itis.gov/">202423</taxonId>
  <taxonomicClassification>
    <taxonRankName>Phylum</taxonRankName>
    <taxonRankValue>Mollusca</taxonRankValue>
    <commonName>mollusks</commonName>
    <taxonId provider="https://www.itis.gov/">69458</taxonId>
    <taxonomicClassification>
      <taxonRankName>Class</taxonRankName>
      <taxonRankValue>Bivalvia</taxonRankValue>
      <commonName>bivalves</commonName>
      <commonName>clams</commonName>
      <taxonId provider="https://www.itis.gov/">79118</taxonId>
      <taxonomicClassification>
        <taxonRankName>Order</taxonRankName>
        <taxonRankValue>Mytiloida</taxonRankValue>
        <taxonId provider="https://www.itis.gov/">79450</taxonId>
        <taxonomicClassification>
          <taxonRankName>Family</taxonRankName>
          <taxonRankValue>Mytilidae</taxonRankValue>
          <taxonId provider="https://www.itis.gov/">79451</taxonId>
          <taxonomicClassification>
            <taxonRankName>Genus</taxonRankName>
            <taxonRankValue>Geukensia</taxonRankValue>
            <taxonId provider="https://www.itis.gov/">79554</taxonId>
            <taxonomicClassification>
              <taxonRankName>Species</taxonRankName>
              <taxonRankValue>Geukensia demissa</taxonRankValue>
              <commonName>ribbed mussel</commonName>
              <taxonId provider="https://www.itis.gov/">79555</taxonId>
            </taxonomicClassification>
          </taxonomicClassification>
        </taxonomicClassification>
      </taxonomicClassification>
    </taxonomicClassification>
  </taxonomicClassification>
```

```
  </taxonomicClassification>
</taxonomicCoverage>
```

Example 7.6: A <**taxonomicCoverage**> element describing two species, omitting the descriptions of higher level taxa. In this example, it is essential to include taxonomic identifiers since a given taxon name could appear under more than one higher level taxonomic rank.

```
<taxonomicCoverage>
  <generalTaxonomicCoverage>
    Mollusks were identified to species
  </generalTaxonomicCoverage>
  <taxonomicClassification>
    <taxonRankName>Species</taxonRankName>
    <taxonRankValue>Detracia floridana</taxonRankValue>
    <commonName>florida melampus</commonName>
    <taxonId provider="https://www.itis.gov/">76463</taxonId>
  </taxonomicClassification>
  <taxonomicClassification>
    <taxonRankName>Species</taxonRankName>
    <taxonRankValue>Geukensia demissa</taxonRankValue>
    <commonName>ribbed mussel</commonName>
    <taxonId provider="https://www.itis.gov/">79555</taxonId>
  </taxonomicClassification>
</taxonomicCoverage>
```

8.3 Annotations (<annotation>, <annotations>)

Semantic annotation is the practice of enhancing the context, utility, and meaning of a dataset by establishing relationships between its metadata and external resources like community ontologies or controlled vocabularies. Elements for semantic annotations are relatively new to EML, and applications are still in development. See the annotation primer by the EML developers for a thorough overview (7 Semantic Annotation Primer), and it is strongly recommended to look for advice and feedback on using semantic annotation in your research and data management communities before implementing them in EML. In brief, EML 2.2.0 supports entering terms from web-accessible ontologies and vocabularies via <**annotation**> elements.

For example, a unit for an attribute might be listed as “g/m2” or “g m-2” or “gramPerMeterSquared” depending on the local conventions for input of units, but an annotation indicating a Uniform Resource Identifier (URI) of http://qudt.org/vocab/unit/GM-PER-M2 would provide a way to indicate that they all are associated with the same underlying unit that is described

in the community-maintained QUDT ontology. The corresponding XML is shown in example 7.7 below.

Example 7.7: An <**annotation**> element for describing a measurement unit by describing its relationship to a unit in community ontology (QUDT). The ontology entry is indicated with a URI.

```
<annotation>
  <propertyURI label="has unit">
    http://qudt.org/schema/qudt/hasUnit
  </propertyURI>
  <valueURI label="GM-PER-M2">
    http://qudt.org/vocab/unit/GM-PER-M2
  </valueURI>
</annotation>
```

The label attribute of each propertyURI or valueURI is flexible and should be human-readable. The URI content is not expected to be human readable, but can be used to distinctly identify a relationship (property) and specific value. Chapter 7 of the EML specification document (7 Semantic Annotation Primer) provides additional information on how annotations are structured and where annotations can appear in EML metadata. Utility of annotations is increased when communities use the same sources for URIs. This avoids the need to crosswalk different ontologies.

Annotations always refer to a particular element of metadata, known as the *subject* of the annotation. To be the subject of an annotation, that element must have an **id** attribute with a unique value (e.g. <**dataset id="dataset-01"**>). Each <**annotation**> element has two required child elements - <**propertyURI**> and <**valueURI**> - that, together with the subject, form a full semantic statement. Annotations are allowed in five locations in the EML document:

- In <**dataset**>, <**attribute**>, or entity (<**dataTable**>, <**otherEntity**>, etc) elements
- in an <**annotations**> root element
- as a child of an <**additionalMetadata**> root element

In the <**dataset**>, <**attribute**>, or entity position, the subject of the annotation is the parent element within which the <**annotation**> element is placed, as in Example 7.8.

Example 7.8: An <**attribute**> element with two child <**annotation**> elements providing ontology references about the variable. Note that the annotated element must be given the **id="att.12"** attribute to become the subject of the two annotations.

```
<attribute id="att.12">
  <attributeName>biomass</attributeName>
  ...
  <annotation>
    <propertyURI label="of characteristic">
      http://ecoinformatics.org/oboe/oboe.1.2/oboe-core.owl#ofCharacteristic
    </propertyURI>
    <valueURI label="Mass">
      http://ecoinformatics.org/oboe/oboe.1.2/oboe-characteristics.owl#Mass
    </valueURI>
  </annotation>
  <annotation>
    <propertyURI label="of entity">
      http://ecoinformatics.org/oboe/oboe.1.2/oboe-core.owl#ofEntity
    </propertyURI>
    <valueURI label="Plant Material">
      http://purl.dataone.org/odo/ECSO_00000503
    </valueURI>
  </annotation>
</attribute>
```

The <**annotations**> element is a root-level (**/eml:eml/annotations**) container intended to hold one or more <**annotation**> elements. A @**references** attribute must be used to establish the subject of any <**annotation**> listed in <**annotations**>. The value of the @**references** attribute should point to the @**id** attribute for the intended element, as in Example 7.9.

Example 7.9: An <**annotations**> element with one <**annotation**> referencing the "CDR-biodiv-table" entity.

```
<annotations>
  <annotation references="CDR-biodiv-table">
    <propertyURI label="Subject">
      http://purl.org/dc/elements/1.1/subject
    </propertyURI>
    <valueURI label="grassland">
      http://purl.obolibrary.org/obo/ENVO_01000177
    </valueURI>
  </annotation>
</annotations>
```

Similarly, <**annotation**> elements within the root level <**additionalMetadata**> element (**eml:eml/additionalMetadata**) may describe elements elsewhere in the EML document.

Establish the subject of these elements using the associated <**describes**> element populated with the @**id** attribute value for the intended element, as in Example 7.10. More information on the structure of <**additionalMetadata**> is given in Chapter 12.

Example 7.10: An <**annotation**> within the <**additionalMetadata**> element used to provide parent organization information for a person identified with the `id=adam.sheperd` attribute.

```
<additionalMetadata>
  <describes>adam.sheperd</describes>
  <metadata>
    <annotation>
      <propertyURI label="member of">
        https://schema.org/memberOf
      </propertyURI>
      <valueURI label="BCO-DMO">
        https://ror.org/00vcb3m70
      </valueURI>
    </annotation>
  </metadata>
</additionalMetadata>
```

8.4 XPaths referenced in this chapter

Keyword set: **/eml:eml/dataset/keywordSet**

A keyword: **/eml:eml/dataset/keywordSet/keyword**

Keyword thesaurus name: **/eml:eml/dataset/keywordSet/keywordThesaurus**

Dataset geographic coverage: **/eml:eml/dataset/coverage/geographicCoverage**

Dataset temporal coverage: **/eml:eml/dataset/coverage/temporalCoverage**

Dataset taxonomic coverage: **/eml:eml/dataset/coverage/taxonomicCoverage**

Dataset annotations: **/eml:eml/dataset/annotations**

Annotation about an EML element: **/eml:eml/additionalMetadata/metadata/annotation**

9 Citations and references

Key points

- In practice, "cited by" information is most often identified after-the-fact and tracked by non-EML systems at repositories or other services.
- Include only one reference within each **bibtex** element, and enclose each **bibtex** element within a **citation** element.
- Use **literatureCited** and other **CitationType** elements as direct children of dataset rather than in lower-level elements (e.g., **methodStep**).

Scholarly research norms call for the attribution of sources in all research works. More recently established open science principles place high value on the reproducibility of research, with the expectation that citation-based links between published research products, such as between journal articles and datasets, will facilitate this. Citation metrics have also become increasingly important for gauging the re-use and impact of any research product, including datasets. The EML standard therefore supports extensive use-cases for citations and referencing in datasets, and we provide details on usage and examples below. These include several new elements that were introduced in EML version 2.2, as described in the EML schema documents.

9.1 Works cited in a dataset

As in other scholarly works (journal articles, theses, etc.), dataset authors should use citations and references to provide proper attribution and identification of sources used to create their dataset. EML provides the <**literatureCited**> element to include a list of references for external resources cited in the dataset, for example methods papers or other published protocols used to derive the data. The <**literatureCited**> element was introduced in EML version 2.2 and is implemented as a list of **CitationType** elements, either <**citation**> or <**bibtex**> (more information below, or in Appendix B). Before this element was introduced, references cited in an EML dataset were usually included as text within the methods description (**methods/methodStep/description,** see Chapter 9, or an EDI example), which is simple, and keeps the references in flow with the method description. However, whether or not references are described elsewhere in EML, **utilizing the <literatureCited> element, preferably at the <dataset> level, is the recommended method to list references**

cited in a dataset. This approach makes citation information more machine-interpretable and accessible to citation tracking systems.

9.2 Citations accrued by a dataset

Using EML to document "cited by" information (a.k.a. inbound citations or backlinks) can be useful when a dataset is referenced or used in other scholarly works, for example when it is described and cited in a publication (like a "data paper"), or when its data are used to generate research results with an appropriate citation in the resulting journal article. These use-cases are supported by EML's <**referencePublication**> and <**usageCitation**> elements, respectively. In many cases, the <**referencePublication**> and <**usageCitation**> elements are challenging or impossible to accurately apply in EML because the dataset is usually published before the work that cites it (the "cart before the horse" problem). In practice, "cited by" information is most often identified after-the-fact and tracked by non-EML systems at repositories or other services. Using these elements in EML is relatively rare and no examples are provided here, but for more information see the EML schema documentation (Section 5.1.4, eml-literature module).

9.3 Using CitationType elements and BibTeX

Tip: BibTeX is pronounced "bib-tek"

The <**literatureCited**> element is a list of one or more child elements, each containing the reference information for a resource the authors have cited in the dataset. Usually, <**citation**> elements are used as the items in the list. The <**citation**> element is built from the EML **CitationType** (see Appendix B), which describes a reference either with a set of component child XML elements, or with a <**bibtex**> child element that contains a BibTeX string, a structured text format for describing citations (more information here).

When not using <**bibtex**>, the source described in a <**citation**> element must be described with some generic resource description elements (<**title**>, <**creator**>, <**pubDate**>, and similar) and a "resource type" child element drawn from EML's list of available types, including <**article**>, <**book**>, <**thesis**>, <**report**>, and others. Each of these resource type elements has a corresponding set of possible child elements. The <**article**> element is the most common resource type and is covered in example 8.1; for others consult the EML schema documents (Section 5.1.4, eml-literature module and CitationType definition). Note that the <**referencePublication**> and <**usageCitation**> elements, if used, are also CitationTypes and are constructed just like <**citation**>.

Example 8.1: A <**citation**> element using the EML child elements to describe a reference publication. This is for a journal article, and therefore uses the <**article**> resource type element.

```
<citation>
  <title>Title of a paper</title>
  <creator>
    <individualName>
      <givenName>Author</givenName>
      <surName>McAuthorson</surName>
    </individualName>
  </creator>
  <pubDate>2017</pubDate>
  <article>
    <journal>EcoSphere</journal>
    <pageRange>158-168</pageRange>
    <publicationPlace>https://doi.org/10.0000/some_doi.123</publicationPlace>
  </article>
</citation>
```

When using <**bibtex**> within a <**citation**> element, all other child elements described above are not used because that information is already encoded in the BibTeX string. The resource types included in the BibTeX may vary depending on the reference management system used, but will be similar to those defined in EML (which are modeled on EndNote types). Example 8.2 gives a simple example.

Example 8.2: A <**citation**> element using BibTeX child elements to describe a reference publication. This is for the same article as in the previous example, with `@article` representing the resource type.

```
<citation>
  <bibtex>
    @article{mcauthorson_2017,
       title = {Title of a Paper},
       doi = {10.0000/some_doi.123},
       journal = {EcoSphere},
       author = {McAuthorson, Author},
       year = {2017},
       volume = {15},
       number = {11},
       pages = {158--168}
    }
  </bibtex>
```

```
</citation>
```

Since references can be added to <**literatureCited**> as <**citation**> elements either with or without <**bibtex**> child elements, dataset authors (or their EML preparation software) must decide which to use. **This document recommends using the citation format that is most convenient to the reader's workflow, and if one cannot decide, then go with BibTeX.** Using a reference manager software, such as Zotero, Mendeley, or EndNote, makes it convenient to export citations as BibTeX directly from a reference collection. Using the BibTeX format also enables users to easily import the citations into their own collection through their reference manager.

BibTex is a fully-developed standard independent of EML, so using the <**bibtex**> child element allows a high degree of flexibility, but can create challenges in interpretation and implementation. This is because

1. The BibTeX format allows chaining multiple citations into one text block, so instead of describing two articles with two <**bibtex**> elements, one could use a single <**bibtex**> element and include the BibTeX for two articles within it (as in Example 8.3).
2. EML allows <**bibtex**> elements to be placed as direct children of <**literatureCited**> (also as in Example 8.3)

As a current best practice, **we recommend including only one reference within each <bibtex> element, and enclosing each <bibtex> element within a <citation> element**. This is demonstrated in example 8.4, and is a more interpretable standard that matches the intent of the <**literatureCited**> and <**citation**> elements. When expressing reference information as BibTeX text, one must remember to escape XML special characters when they appear, such as a less-than sign (<) in the title of a paper, as in any XML element's content. If escaping is necessary in BibTeX, one can enclose the entire content of the <**bibtex**> element in a CDATA block. See Appendix B for more details about XML special characters.

Example 8.3: Example of <**literatureCited**> element with one child <**bibtex**> element describing two journal articles. This arrangement of references in <**literatureCited**> is valid, but not currently recommended.

```
<dataset>
  …
  <literatureCited>
    <bibtex>
      @article{mcauthorson_2017,
         title = {How To Collect Temperature Data},
         doi = {10.0000/some_doi.123},
         journal = {EcoSphere},
         author = {McAuthorson, Author},
```

```
        year = {2017},
        pages = {158--168}
      }
      @article{delacroix_2018,
        title = {How To Measure Animal Happiness},
        doi = {10.0001/other_doi.321},
        journal = {Ecosquare},
        author = {Delacroix, Eugenia and Lee, Vonda and Weiss, Ragnar},
        year = {2018},
        pages = {15--16}
      }
    </bibtex>
  </literatureCited>
  …
</dataset>
```

Example 8.4: Example of <**literatureCited**> element with child <**citation**> elements describing two journal articles with <**bibtex**>. This is the recommended way to use BibTex formatting for multiple references in <**literatureCited**>.

```
<dataset>
  …
  <literatureCited>
    <citation>
      <bibtex>
        @article{mcauthorson_2017,
          title = {How To Collect Temperature Data},
          doi = {10.0000/some_doi.123},
          journal = {EcoSphere},
          author = {McAuthorson, Author},
          year = {2017},
          pages = {158--168}
        }
      </bibtex>
    </citation>
    <citation>
      <bibtex>
        @article{delacroix_2018,
          title = {How To Measure Animal Happiness},
          doi = {10.0001/other_doi.321},
          journal = {Ecosquare},
          author = {Delacroix, Eugenia and Lee, Vonda and Weiss, Ragnar},
```

```
            year = {2018},
            pages = {15--16}
          }
        </bibtex>
      </citation>
    </literatureCited>
    …
</dataset>
```

9.4 Where (in EML) and when to use citation metadata

Citation elements can appear in several places in EML. For example, a <**methodStep**> can include one or more <**citation**> child elements as references for a dataset's methods section (**methods/methodStep/citation,** see Chapter 9), which was a common practice before <**literatureCited**> was released in EML 2.2. For better consistency and machine-readability, however, **use <literatureCited> and other CitationType elements as direct children of <dataset>**. To summarize the use cases for these once again:

- The <**literatureCited**> element is a list including one or more child <**citation**> or <**bibtex**> elements, each describing resources the authors have cited in the dataset. These might include research publications demonstrating a concept, methods papers (such as methods used to derive the data), published protocols, or other foundational works that are cited in the EML metadata.
- The <**referencePublication**> element is reserved for describing publications whose purpose is to describe the dataset and its use. For example, as the profile and importance of open data has risen, dataset descriptor articles (or "data papers") are often used to publicize newly released datasets. Typically there is only one reference publication for any given dataset.
- A <**usageCitation**> is used for publications that reference the dataset, such as a journal article presenting research results derived from the data. Data citation is important for proper attribution and reproducibility of research, and is increasingly required by publishers.

As a general rule, <**referencePublication**> and <**usageCitation**> elements are difficult to use in EML documents because the resource to be described is usually published after the EML document is created. If the resource is available with a DOIs at the time of EML creation, then they are fine to include, but alternative methods of linking these types of resources to a published EML dataset are typically preferred.

EDI context note

The EDI repository has a system for tracking citations of EDI datasets external to EML metadata documents. This allows an up-to-date listing of citations without requiring a new version of the EDI dataset be produced. Add dataset usage citations with the "Add Journal Citation" link at the bottom of the landing page for each dataset in EDI.

9.5 XPaths referenced in this chapter

Literature cited: **/eml:eml/dataset/literatureCited**

Literature cited citation child: **/eml:eml/dataset/literatureCited/citation**

Literature cited BibTex child: **/eml:eml/dataset/literatureCited/bibtex**

Usage citation: **/eml:eml/dataset/usageCitation**

Reference publication: **/eml:eml/dataset/referencePublication**

Citation in a methods step: **/eml:eml/dataset/methods/methodStep/citation**

10 Methods

Key points

- **methods** elements are intended to be human readable instead of machine-readable.
- It is preferable to include detailed methods in the EML document rather than to as links to external documents.
- It is no longer recommended to place **citation** elements within a **methodStep** - use the **literatureCited** element instead.
- A flowchart in this chapter outlines decision points for what children and content to include in a **methods** element.

The methods and procedures used to generate data objects should be described in the EML <**methods**> tree and its child elements. As a 'rule of thumb,' the content of the <**methods**> tree is intended to be human readable instead of machine-readable, and should fully describe all steps used when collecting and processing the data. When these steps follow an established protocol, references to external sources, such as a published article (with DOI) containing the methods, may be given, but in most cases it is preferable to include detailed methods descriptions that can be readily browsed, indexed for searching, and will remain available with the data even if links to external documents fail. In addition to content in EML <**methods**>, images, documentation files (e.g. txt or pdf), or code illustrating methods may be included as "other entities" in your dataset. Although the <**methods**> tree can appear at the <**dataset**>, entity, and <**attributes**> levels, it is recommended to describe all methods used at the dataset level. This recommendation assures that a data reuser can find all important information in one place.

A <**methods**> tree must contain at least one <**methodStep**> element, and may optionally have a <**sampling**> element and any number of <**qualityControl**> elements. The <**methodStep**> element may have the optional child elements <**dataSource**>, <**instrumentation**>, and <**software**>. Figure 9.1 outlines the likely decision points needed to choose what child elements to include in <**methods**>.

10.1 Methods steps (<methodStep>)

At least one <**methodStep**> element is required under <**methods**>, and each step should contain a logical portion of the methods used to create the published data; for example, sampling design, field data collection, laboratory procedures, quality control/assurance, and statistical analysis. A dataset <**methods**> element may contain as many <**methodStep**> elements as are necessary. All text describing each step must be placed within a <**description**> element, which is a **TextType** (see Appendix B). For <**description**> elements that require complex formatting or layouts, the standard **TextType** elements (<**section**>, <**para**>, etc.), Markdown formatting (<**markdown**>), and LaTex typesetting may be used. Below are additional, optional child elements to <**methodStep**>.

10.1.1 Data provenance (<dataSource>)

The optional <**dataSource**> element is for creating references to published data that was used as source data for the dataset being described (i.e. a dataset's provenance). For instance, if the data being described by EML is mean annual temperature values calculated from hourly temperature measurements published in a separate dataset, the <**dataSource**> element could be used to identify and provide a reference to these source data. This element may contain a fairly complete set of child elements describing the source data, including a <**title**>, <**creator**>, and <**contact**>, but the most important child element is a <**distribution**> element containing the source dataset's DOI or other URL (**distribution/online/url**). It is strongly recommended to include <**dataSource**> elements when publishing any dataset derived from other sources.

> EDI context note
>
> The <**dataSource**> element is used by the EDI repository's provenance tracking system for linking between derived and source data packages. For more information, see additional data repository resources from EDI.

Example 9.1: An example of a <**methods**> element with one <**methodStep**> element that includes a <**dataSource**> child element to indicate data provenance. The <**description**> element uses **TextType** <**section**>, <**title**>, and <**para**> formatting elements, but note that it is not sufficiently detailed for a derived dataset.

```
<methods>
  <methodStep>
    <description>
      <section>
        <title>Data collection</title>
        <para>
```

```
          We utilize NPP data collected from 1906 to 2006 from the FRS
          site. The FRS NPP data unit definition is kg/m\^2/yr.
        </para>
      </section>
    <section>
      <title>Quality control and analysis</title>
        <para>These data were cleaned before analysis.</para>
    </section>
    </description>
    <dataSource>
      <title>NPP data from FRS 1906 to 2006</title>
      <creator>
        <organizationName>Fictitious Research Site</organizationName>
      </creator>
      <distribution>
        <online>
          <onlineDescription>This DOI link references a source dataset that
↪  was used to create this derivative dataset.</onlineDescription>
          <url function="information">
            https://doi.org/10.6073/pasta/91789c93a8930c3091bc5849060ff672
          </url>
        </online>
      </distribution>
      <contact>
        <organizationName>Fictitious Research Site</organizationName>
        <positionName>Information Manager</positionName>
        <electronicMailAddress>
          data.manager@ficstate.edu
        </electronicMailAddress>
      </contact>
    </dataSource>
  </methodStep>
</methods>
```

10.1.2 Instrumentation and software

The <**instrumentation**> and <**software**> elements are both optional and only occasionally used. Both are most common in disciplines and projects that rely on extensive instrumentation and data post-processing, for example marine science, micrometeorology, sensor networks, or bioinformatics datasets. There are ongoing efforts to create standardized metadata and controlled vocabularies for these categories of metadata, which may improve their utility and

ease-of-use in the future.

The <**instrumentation**> element may contain a full description of the instruments used, including manufacturer, model, calibration dates and accuracy. Changes in instrumentation and dates of changes should be mentioned earlier under the <**description**>. The <**software**> element describes software used to process the data. Include relevant details such as software title, author, implementation (hardware, operating system), and version.

Example 9.2: An example of a <**methods**> element with one <**methodsStep**>. The <**description**> element uses a <**markdown**> element for complex layouts, and there are two <**instrumentation**> child elements that indicate the equipment used to collect the data. Note that descriptive markdown text has been abbreviated to save space where indicated by ellipses.

```
<methods>
  <methodStep>
    <description>
      <markdown>
## Sampling design and supplies for arthropod biodiversity monitoring

At each monitoring site, between 10 and 21 pitfall traps were installed in
 ↪  randomized locations...

**Supplies used:**

* pitfall traps - P-16 plastic Solo cups with lids
* metal spades and large bulb planters (to dig holes in which to put traps)
* 70% ethanol (to preserve specimens)
* Qorpak glass jars with lids from the VWR Corporation, 120ml (4oz), cap size
 ↪  58-400 (comes included), Qorpak no. 7743C, VWR catalog no. 16195-703.

## Data collection procedures

... On each collection date, all trapped taxa were retrieved from the traps
 ↪  and returned to the lab. In the lab arthropods were counted, measured
 ↪  (body length in mm), and taxa were identified to Family. Specimens were
 ↪  preserved...

## Tissue isotope analysis

... subsamples of arthropods were sorted by taxon, oven dried, ground in
 ↪  liquid nitrogen, and then analyzed in a combustion CHN elemental analyser
 ↪  coupled to an isotope ratio mass spectrometer (IRMS)...
      </markdown>
```

```
    </description>
    <instrumentation>Fisons 1110 CHN elemental analyzer; CE Elantech, Inc.,
 ↪  Lakewood, NJ, USA</instrumentation>
    <instrumentation>Finnigan DELTAplus Advantage mass spectrometer; Thermo
 ↪  Scientific Inc., West Palm Beach , FL, USA</instrumentation>
  </methodStep>
</methods>
```

10.1.3 Citations

Any <**methodStep**> element may contain an indeterminate number of <**citation**> child elements for documenting references to external sources. The <**citation**> element is a CitationType designed for bibliographic information, which can be useful when referencing, for example, a methods paper or published protocol used to collect the data. With the introduction of new citation and referencing features in EML 2.2, **it is no longer recommended to place <citation> elements within a <methodStep>** - use the <**literatureCited**> element instead, preferably at the <**dataset**> level. See Chapter 8 and Appendix B for more information.

10.2 Optional <methods> child elements

There are two other optional children of the <**methods**> element: <**sampling**> and <**qualityControl**>. These elements are designed to contain very specific information about sampling procedures and post-collection data handling. In most cases this information is easy to include in one or more <**methodsStep**> elements as described above, but if more detailed and structured descriptions of sampling and QA/QC procedures are needed these elements may be used. See Appendix A for further information.

10.3 Decision flowchart

Figure 9.1 This flowchart outlines decision points for what children and content to include in an EML <**methods**> element.

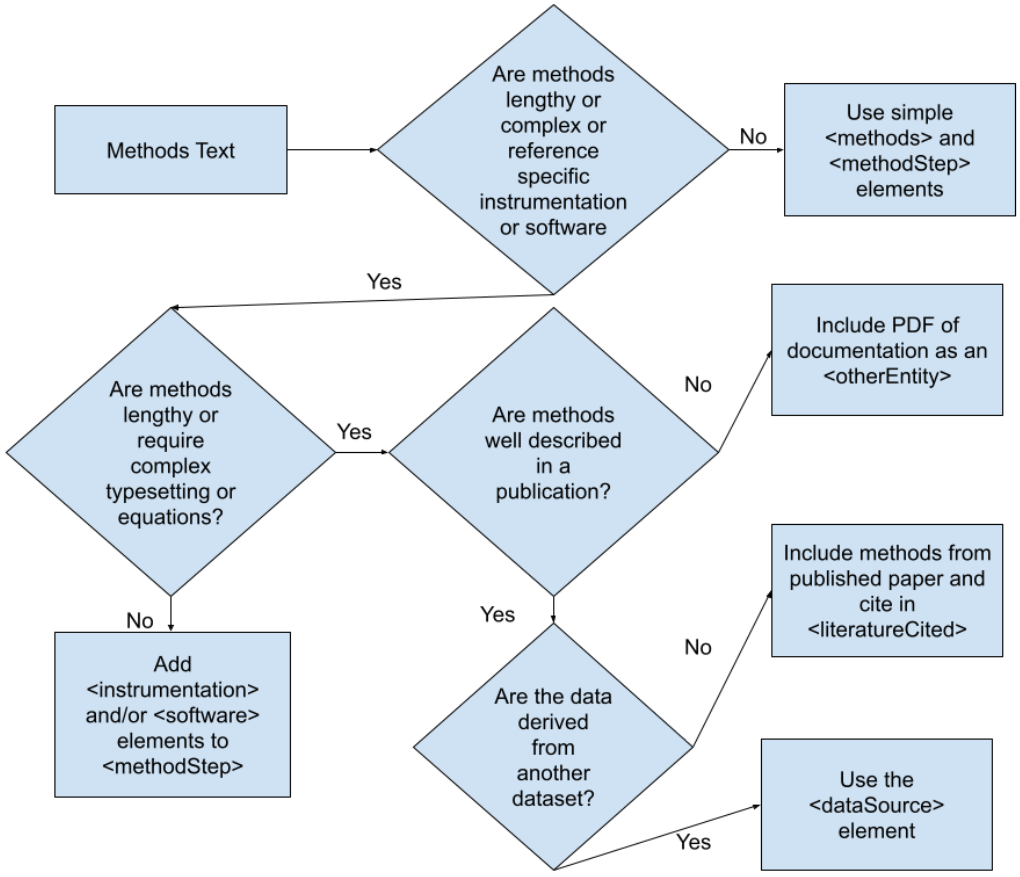

Figure 10.1: A flowchart for determining the structure and content of a methods element

10.4 XPaths referenced in this chapter

Dataset methods: **/eml:eml/dataset/methods**

Dataset method steps: **/eml:eml/dataset/methods/methodStep**

TextType method step: **/eml:eml/dataset/methods/methodStep/description/section/...**

Markdown method step: **/eml:eml/dataset/methods/methodStep/description/markdown**

Provenance metadata: **/eml:eml/dataset/methods/methodStep/dataSource**

Instrumentation: **/eml:eml/dataset/methods/methodStep/instrumentation**

Software: **/eml:eml/dataset/methods/methodStep/software**

11 Data entities

Key points

- Publish tabular data as plain delimited text files when possible, instead of proprietary or binary file formats.
- For non-tabular data in **otherEntity** elements, use open, non-proprietary file formats to the extent possible.
- Place standardized descriptive text in the **entityType** child element that complements the standardized file format information included in **physical/dataFormat/externallyDefinedFormat/formatName**.
- For **externallyDefinedFormat** we recommend using media type values (formerly known as MIME types).

An EML document is a container for metadata that describe the digital products of research or data collection activities, which can include a wide range of file types and other data objects. Data entities are a class of EML elements that directly describe these digital objects, and altogether, six data entity types have been defined for this purpose. This chapter provides thorough guidance on using <**dataTable**> and <**otherEntity**> data entity elements because they are very commonly used in EML documents and are capable of describing a wide range of data objects or files. The <**spatialRaster**> and <**spatialVector**> entity types are occasionally used for geospatial data objects and are addressed in depth in the spatial data chapter of the "Data Package Design for Special Cases" companion document. The <**view**> and <**storedProcedure**> entity types are only briefly described in Appendix A because they refer to specific data products derived from relational databases that are rarely used. To assist with selection of an EML entity type, table A.1 in Appendix A describes the purpose, example data objects, and metadata features of all six available types.

Every EML data entity has a set of potential child elements in common, called the **EntityGroup**, for storing general information about the data resource. Some of these are required and some are recommended. In addition, each data entity type has a set of child elements specific to the type with, again, some required and some recommended. Table 10.1 summarizes typical use cases for <**dataEntity**> and <**otherEntity**>, and their required and recommended child elements. More details about child elements come later in the chapter.

Table 10.1. Required and recommended child elements for <**dataTable**> and <**otherEntity**> types. Members of the **EntityGroup** are given, followed by elements specific to the entity

type. This is a subset of Table A.2 in Appendix A.

Entity Type	Typical Uses	Required and recommended elements
<**dataTable**>	Tabular data objects with a fixed structure (delimited text files, simple spreadsheet, etc.)	**EntityGroup** <**entityName**> (required) <**entityDescription**> <**physical**><**attributeL** (required)<**numberOfRecords**>
<**otherEntity**>	Data objects not described by other standard entity types (images, non-tabular data, binary files, etc.)	**EntityGroup** <**entityName**> (required) <**entityDescription**> <**physical**><**entityTyp** (required)<**attributeList**> (for tabular data)

EDI context note

Like many other EML elements, each of the entity elements has three attributes, @**id**, @**scope**, and @**system**. EDI will automatically fill them in with the actual entityID within the EDI repository system (a hash), `scope = "document"`, and `system = "https://pasta.edirepository.org"`. Hence, these attributes should not be used for local information as they will be overwritten upon submission. Use <**alternateIdentifier**> instead.

11.1 Recommendations for <dataTable>

Carefully consider what files or data objects should be described with a <**dataTable**> element. If at all possible, do not publish tabular data in dated, proprietary, or binary file formats such as Microsoft Excel files. When handling tabular data in such formats it is preferable to export to accessible, open data formats, such as plain delimited text files (such as CSV) for publication. The <**attributeList**> tree is required in <**dataTable**> entities to describe all attributes, or column variables, in the table. Best practices for <**attributeList**> are in Chapter 11.

Example 10.1: An example of a <**dataTable**> entity. The elements in the EntityGroup are shown, along with an abbreviated <**attributeList**>, which is required for a <**dataTable**>.

```
<dataTable>
  <entityName>Arthropod habitat table</entityName>
  <entityDescription>
    habitat description table for the sampling locations
  </entityDescription>
  <physical>
    <objectName>frs-1-arthro-hab.csv</objectName>
    <dataFormat>
```

```
      <textFormat>
        <numHeaderLines>1</numHeaderLines>
        <numFooterLines>0</numFooterLines>
        <recordDelimiter>\r</recordDelimiter>
        <numPhysicalLinesPerRecord>1</numPhysicalLinesPerRecord>
        <attributeOrientation>column</attributeOrientation>
        <simpleDelimited>
          <fieldDelimiter>,</fieldDelimiter>
          <quoteCharacter>"</quoteCharacter>
        </simpleDelimited>
      </textFormat>
    </dataFormat>
    <distribution>
      <online>
        <onlineDescription>frs-1 Arthro Habitat Data File</onlineDescription>
        <url function="download">
          http://www.ficstate.edu/lter/data/frs-1-arthro-hab.csv
        </url>
      </online>
    </distribution>
  </physical>
  <attributeList>
    …
  </attributeList>
</dataTable>
```

11.2 Recommendations for <otherEntity>

When publishing an <**otherEntity**>, we recommend using open, non-proprietary file formats to the extent possible, so that the published data objects can be used without restriction (such as requiring an expensive software license). Though <**otherEntity**> is something of a catch-all data entity type, it requires a free text <**entityType**> child element to describe whatever data object is being published. It is strongly recommended to place standardized descriptive text here that complements the standardized file format information included in **physical/dataFormat/externallyDefinedFormat/formatName**. This is demonstrated in Example 10.2, and Table 10.2 provides suggested <**entityType**> values for a variety of file formats. Further guidance on this, including sources for standard terms, is given below in the "Physical tree" section. Additional information needed to fully describe the file format, data structure, use, or other aspects of the data object can also be provided in the element's <**entityDescription**> as needed.

EDI context note

If archiving HTML files at EDI, you must use a <**formatName**> with the value "text/html" for the entity to be accepted.

Example 10.2: An example of an <**otherEntity**> element. The recommended elements in the **EntityGroup** are shown, as is a standardized value in the required <**entityType**> element. Note the corresponding IANA standard text in the <**formatName**> element.

```
<otherEntity>
  <entityName>Field and lab protocol for arthropod sampling</entityName>
  <entityDescription>
    An Adobe PDF (archival PDF/A) document describing the habitat, sampling
 ↪  locations, field collection protocols, and laboratory processing
 ↪  procedures for the FLS arthropod sampling program.
  </entityDescription>
  <physical>
    <objectName>frs-1-artho-protocol.pdf</objectName>
    <dataFormat>
      <externallyDefinedFormat>
        <formatName>application/pdf</formatName>
      </externallyDefinedFormat>
    </dataFormat>
    <distribution>
      <online>
        <onlineDescription>frs-1 Arthropod Protocol</onlineDescription>
        <url function="download">
          http://www.ficstate.edu/lter/protocols/frs-1-arthro-protocol.pdf
        </url>
      </online>
    </distribution>
  </physical>
  <entityType>Portable Document Format</entityType>
</otherEntity>
```

11.3 EntityGroup elements

In the **EntityGroup**, the <**entityName**> element is required, and <**entityDescription**> and <**physical**> (including optional <**access**>) are recommended. Optional additional **EntityGroup** elements include <**alternateIdentifier**>, <**additionalInfo**>, <**coverage**>, and <**methods**>. These are briefly described after the required and recommended elements,

but for better readability by data users, it is generally recommended that their content be provided at the dataset level and not distributed throughout the EML document.

11.3.1 Entity name (required)

The entity name (<**entityName**>) is a human readable name of the data object, whether that is a file, database table, document, or something else. The content should not exceed 100 characters.

> EDI context note
>
> The EDI repository requires that <**entityName**> elements be unique within the dataset. This element will be displayed on the dataset landing page as the name for the entity.

11.3.2 Entity description (recommended)

The <**entityDescription**> element should contain a longer, more descriptive explanation of the data in the entity. Like all descriptions, it is human-readable text, and should help determine if it is appropriate for a particular use. This element is an appropriate place to elaborate on the format of the data object if it cannot be easily described within the <**physical**> tree (see discussion under <**dataFormat**> below).

11.3.3 Physical tree (recommended)

The <**physical**> tree (**/eml:eml/dataset/[entity]/physical**) describes the physical format and location of the data object. It should contain sufficient detail to allow a data user to obtain and access the object with the correct software tools. There are a number of child elements in the <**physical**> tree that we recommend populating.

11.3.3.1 <characterEncoding>

The <**characterEncoding**> element defines the encoding of any text file data objects. For most English language based data, an encoding of UTF-8 is typically correct, with ASCII being another typical encoding. Whatever you choose, if you do provide an encoding, please be sure it is not an incorrect one, e.g., do not choose ASCII if your data include extended Latin characters.

11.3.3.2 <objectName>

The <**objectName**> is the filename of the object in a file system or wherever it is accessible via the internet. For example, "NPPdata_FRS_2006-2020.csv".

11.3.3.3 <dataFormat>

The <**dataFormat**> element defines the internal physical format of the data object. The three possible child elements to choose choose within this element are

1. <**textFormat**>, which describes a formatted text data object (a CSV file for example) using a range of child elements. This is commonly used to describe the data format of <**dataTable**> elements.
2. <**externallyDefinedFormat**>, which describes data objects that are in prescribed formats other than text format (e.g., NetCDF, KML, Excel). This element is commonly used for <**otherEntity**> elements.
3. <**binaryRasterFormat**>, which describes a raster data file such as a GeoTIFF, which is useful when publishing spatial data files (See the spatial data chapter of the "Data Package Design for Special Cases" companion document).

When using <**externallyDefinedFormat**>, the format should be named in the <**formatName**> child element. To promote machine interpretability, we recommend using media type values (formerly known as MIME types) from the **template** column, e.g., "application/zip", of IANA's Media Types list, which is the authoritative source for internet media types. Many media types are not registered with IANA, so if your format is not present there, you may use one of the *media types* from DataONE's format list, which includes many non-standard, but still commonly accepted media/MIME types. If you cannot find a matching IANA media type, specify your own, and consider adding it to IANA.

The <**otherEntity**> element commonly describes non-tabular data objects that require an **externallyDefinedFormat/formatName** element. In addition to selecting a standard IANA or DataONE media type value, it is also beneficial to populate the required <**entityType**> element with the corresponding *media name* from IANA or DataONE. This is demonstrated in Example 10.2, and Table 10.2 gives further examples. To give the reader even more clues about how to use or interpret the non-tabular data object, it is recommended to provide additional information about the data object in <**entityDescription**>.

Table 10.2. Recommended entity types and format names for some files that can be included in an <**otherEntity**>.

Common Name	<entityType> value(from DataOne or similar)	<formatName> value(IANA if possible, or DataOne)
R script	R programming language script	text/x-rsrc

Common Name	<entityType> value(from DataOne or similar)	<formatName> value(IANA if possible, or DataOne)
R markdown file	R Markdown file	text/markdown
Python script	Python programming language script	text/x-python
JPEG image	JPEG	image/jpeg
PDF document	Adobe Portable Document Format	application/pdf
Zip file	Zip file format	application/zip

EDI context note

If archiving HTML files at EDI, you must use a <**formatName**> with the value "text/html" for the entity to be accepted.

11.3.3.4 <distribution>

The <**distribution**> tree provides information on how the resource is distributed, and the contents of this tree is generally covered at the <**dataset**> level (refer to Chapter 6). However, there are a few points which will be reiterated. For submission of the dataset to a repository, the content of a <**url**> element at the entity level should deliver data, and not point to another application or use page. The <**url**>'s @**function** attribute should have the value "download" (i.e. <**url function="download"**>). This is implied if the @**function** attribute is omitted.

EDI context note

EDI provides a 'manual' upload option for smaller data entities directly from the user's desktop. For this option, the distribution URL does not need to be provided. Upon submission, EDI will replace the distribution URL with the repository download URL for the data object.

An optional <**access**> element in a data entity's <**distribution**> tree is intended specifically to control access to the data entity separately from the metadata. For more information on using the <**access**> tree, refer to the discussion in Chapter 6.

11.3.4 Optional EntityGroup elements

All of the EntityGroup elements below are optional. They may add value at the data entity level under some circumstances, but in general they are more useful at the <**dataset**> level.

- **Alternate identifier:** The primary identifier for a data entity belongs in the @**id** attribute of the entity element (e.g., <**dataTable id="xxx"**>) and may be filled in by the repository, but adding the <**alternateIdentifier**> element can accommodate additional identifiers that might be used in a local data management system. It is used similarly to the <**alternateIdentifier**> element at the <**dataset**> level, as described in Chapter 3.
- **Annotations:** The <**annotation**> element can be used at the entity level to establish links to semantic ontologies and other external resources that provide context or utility to the entity or its content. See Chapter 7 for more details.
- **Additional information:** The <**additionalInfo**> element is a text field for any material that cannot be characterized by the other elements for the data type.
- **Coverage and methods:** Entity-level <**coverage**> and <**methods**> elements can provide information on the geographic, taxonomic and temporal coverages, or data collection methods, for the data entity. The general recommendation is that these be placed at the <**dataset**> level instead, but there may be specific use cases that benefit from entity-level coverage or methods information. See more details about coverages in Chapter 7, and about methods in Chapter 9.

11.4 XPaths referenced in this chapter

Data table entity: **/eml:eml/dataset/dataTable**

Other entity: **/eml:eml/dataset/otherEntity**

Data table entity name: **/eml:eml/dataset/dataTable/entityName**

Other entity description: **/eml:eml/dataset/otherEntity/entityDescription**

Data table physical tree: **/eml:eml/dataset/dataTable/physical/**

Object name (filename): **/eml:eml/dataset/dataTable/physical/objectName**

Format name: …**physical/distribution/dataFormat/externallyDefinedFormat/formatName**

Other entity type: **eml:eml/dataset/otherEntity/entityType**

Other entity annotation: **eml:eml/dataset/otherEntity/annotation**

12 Variable descriptions (Attributes)

Key points

- Attribute metadata in EML should provide the name and definition of each attribute, its measurement domain, a description of coded values, definitions of missing values, and other pertinent information.
- Use of the optional **storageType**, **missingValueCode** and **annotation** elements are recommended.

The data entities described by an EML document are typically files containing some kind of research or monitoring data. As such, these data entities hold one or more variables, which are the amounts, characteristics, or other values being measured or controlled during data collection. In EML, a data entity's variables are referred to as "attributes," and the *eml-attribute* module provides EML structures to describe all attributes (variables) within a data entity. Attribute metadata in EML should provide the name and definition of each attribute, its measurement domain, a description of coded values, definitions of missing values, and other pertinent information. All <**dataTable**> entities are required to contain an <**attributeList**> element with this information, as are other, less-commonly used entity types.

12.1 Attribute lists and attributes

The <**attributeList**> tree is required for all data types except for <**otherEntity**>. It describes all attributes in a data entity as a list of individual <**attribute**> elements. Each <**attribute**> element must fully describe a particular attribute (variable) of the data entity using the range of required and optional child elements described below. An <**attribute**> element should have an **id** attribute (e.g., <**attribute id="att.01"**>) if it will be referenced by annotations or other elements of EML. Required and optional child elements of an <**attribute**> element are described below.

12.1.1 Attribute names (required)

The <**attributeName**> element contains the identifier or name of an attribute (variable) in a data table, such as the header name for a column in a CSV file. Values in <**attributeName**>

must precisely match the name or identifier of an attribute in the data entity object to which they refer, and must be composed of alphanumeric characters, typically in the Unicode set. Attribute names are usually a short name or abbreviation for the full attribute name or description. When preparing a tabular data file for use or publication, we recommend making attribute names (e.g. column headers) clear, concise, and easy to type. They should also give some indication of the meaning or content of the data field. We recommend NOT 1) starting the attribute name with a number, 2) using special characters and spaces and 3) incorporating units into the attribute name (because other metadata elements, e.g. <**unit**>, should be used for this). Following these recommendations makes the data file easier to describe and reuse.

EDI context note

In the EDI repository, <**attributeName**> elements must be unique within a data entity.

12.1.2 Attribute definitions (required)

The <**attributeDefinition**> element is a text field giving a precise and complete text definition of the attribute. It explains the contents of the attribute fully so that a data user can interpret the attribute accurately.

12.1.3 Measurement scale (required)

The <**measurementScale**> element describes the *scale* from which values of the attribute are drawn. Measurement scales assign meaning to the values of an attribute (or variable) in a data entity, examples being the units of a numeric variable, a list of identifiers or code names in a categorical variable, or a description of format used for date or time variables. One of five scale types defined in the EML schema must be inserted as a child element to <**measurementScale**>. Two of these are non-numeric (<**nominal**>, <**ordinal**>) and three are numeric (<**interval**>, <**ratio**>, <**dateTime**>). The significance of these <**measurementScale**> child elements are as follows:

- A nominal scale (<**nominal**>) is a non-numeric scale where named values are used to distinguish one observation or category from another. Nominal values have no intrinsic logical or ordered relationship to each other, they serve only as identifiers or descriptors. An attribute using a nominal scale might include simple text identifiers for individuals ("Gabe", "Corinna", "Tim"…), named groups or categories in the data ("control" or "treatment"), or plain text descriptions ("sample vial eaten by a bear"). Note that observations or groups in nominal data can be coded numerically (e.g. 1=male, 2=female), so nominal data CAN contain numbers, but these numbers are not intrinsically meaningful.
- Ordinal scales (<**ordinal**>) are also non-numeric, but ordinal values have a logical or ordered relationship to one another where the magnitude of the differences between the

values is not defined or meaningful. For example, qualitative data that is ordered (Low, Medium, High) or ranked (1=Good, 2=Fair, 3=Poor) uses the ordinal scale. Note that ordinal data can also be coded numerically.

- An interval measurement scale (<**interval**>) uses numeric values that have equal-sized increments (or units) between one another and an arbitrary zero point. For example, the Celsius temperature scale uses equally-spaced degrees, but zero does not represent "absolute zero" (i.e., the temperature at which molecular motion stops), and 20 C is not "twice as hot" as 10 C. Latitude and longitude are also intervals since their zero points are arbitrarily chosen.
- Ratio measurement scales (<**ratio**>) also use numeric values with equally-spaced increments, but with a meaningful zero point that allows legitimate ratio comparisons. For example, the Kelvin scale reflects the amount of kinetic energy of a substance (i.e., zero is the point where a substance transmits no thermal energy), and so temperature measured in kelvin units is a ratio measurement. Concentration is also a ratio measurement because a solution at 10 micromolePerLiter has twice as much substance as one at 5 micromolePerLiter.
- Date-time measurement scales (<**dateTime**>) use date and time values from the Gregorian calendar, including year, month, day, hour, minute, and second. Though these essentially are ordered, numeric data, dates and times may be represented in many different ways and must therefore be described with specialized metadata.

Each of the five <**measurementScale**> child elements described above requires its own child elements to fully describe the measurement scale.

12.1.3.1 Describing non-numeric scales (<nominal> and <ordinal>)

Both the <**nominal**> and <**ordinal**> measurement scale elements require a <**nonNumericDomain**> element with either a text domain element (<**textDomain**>) indicating plain text values, or an enumerated list of codes (<**enumeratedDomain**). The <**enumeratedDomain**> describes categorical data and requires one or more <**codeDefinition**> elements to describe each category or coded value present in the attribute using the <**code**> and <**definition**> child elements. Note that codes and their explanations can also be provided in a referenced data entity (<**entityCodeList**>) or an external source (<**externalCodeSet**>), but these are rarely used and not generally recommended. For <**textDomain**> an optional pattern provided in <**definition**> may describe the text, e.g., a ten-digit telephone number (area code and number) can be described with the format `\d\d\d-\d\d\d-\d\d\d\d`.

12.1.3.2 Describing numeric scales (<interval> and <ratio>)

The numeric <**interval**> and <**ratio**> measurement scales require child elements to describe measurement units (<**unit**>) and number type (<**numericDomain**>), and can optionally describe the floating point precision of attribute values (<**precision**>). Any measurement units

included in <**unit**> must be described in correct physical units that are clearly defined and linked to the International System of Units (SI). Terms which describe data but are not units, such as the substance or substrate being measured, should be placed in <**attributeDefinition**>. For example, for data describing "milligrams of carbon per square meter," "carbon" belongs in the <**attributeDefinition**>, while the <**unit**> is "milligramPerMeterSquared." A wide selection of units are already included in the EML schema's standard units dictionary (see the EML schema docs). To assign one of these to an attribute, include it by name in a <**standardUnit**> child element (**unit/standardUnit**).

If an appropriate unit is not available in the EML standard units dictionary, or if a unit from a different source is desired, a unit name should be placed in a <**customUnit**> element (**unit/customUnit**) and then defined in an <**additionalMetadata**> element, generally at the <**dataset**> level (refer to Example 12.2 and the Custom Unit part of Chapter 12 for details). There are many alternative unit dictionaries or ontologies that can be used as custom units in EML, such as UCUM.org ("kg/m2" or "kg.m-2") and QUDT.org ("kiloGM-PER-M2"). Each has some advantages with respect to interpretability, lack of ambiguity, use of special characters (e.g., " "), brevity, degree of standardization, and desirability as perceived by researchers. Most of these can also be coupled to <**annotation**> elements that link the unit to online resources from the dictionary/ontology itself. For general purposes, one may also define custom units independently of other standards, such as by naming custom units following the pattern set by EML standard units ("kilogramPerMeterSquared"), When doing so, the following guidelines from ISO recommendations apply to their naming: 1) Units should be written out, not abbreviated. 2) Unit modifiers, such as "squared," should follow the unit being modified. For example, meterSquared is preferred, while squareMeter is improper. 3) Units should be singular, such as "meter," and not plural, such as "meters." Again, all units placed in <**customUnit**> must be defined in the document's <**additionalMetadata**> elements using the conventions described in Chapter 12.

> Community context note
>
> The EDI repository and LTER Network are currently developing a new custom units framework for EML based on the QUDT ontology and annotations. Following the best practices in this document will allow an easy transition to this new system.

The <**numericDomain**> element describes the numeric values of an attribute using a required <**numberType**> element with a controlled vocabulary (*natural*, *whole*, *integer*, *real*, defined in the EML schema documentation), and, optionally, a <**bounds**> element describing the minimum and maximum allowable values of the numeric attribute. The <**bounds**> are theoretical or allowable minimum and maximum values (prescriptive), rather than the actual observed range in a data set (descriptive).

The optional <**precision**> element describes the number of decimal places for the attribute, which is useful to determine rounding criteria, estimate storage size, and choose system-specific data types (e.g. "float" vs "long" in C) for an attribute. Currently, EML does not allow more

than one precision value for a column, so variable attribute precision information should be described in a <**methodStep**> element.

12.1.3.3 Describing <dateTime> scales

Date and time variables can be expressed in many different formats, some of which are ambiguous. The <**formatString**> child element (**dateTime/formatString**) is required to to describe the format of any dateTime attributes in a data entity, and it is strongly recommended to use the ISO 8601 standard in data entities described with EML. An example of an allowable ISO date-time format is "YYYY-MM-DD," as in 2004-06-25, or, more fully, as "YYYY-MM-DDThh:mm:ssTZD" (eg 1997-07-16T19:20:30.45Z). Place whatever string is appropriate to describe the data directly into a <**formatString**> element. The ISO standard is quite strict, and legacy datasets or equipment outputs (e.g. from sensors) often contain non-ISO standard dates. The EML schema therefore provides additional allowable formats (see the EML documentation for a complete list). Note that a <**dateTime**> measurement scale cannot be used to describe time durations. In that case, use a ratio measurement scale with a unit such as seconds, nominalMinute or nominalDay (all from the EML standard units library) which are defined in relation to SI second. To describe the measurement precision of dates or times in a <**dateTime**> scales, using the <**dateTimePrecision**> element is recommended.

12.1.4 Attribute labels (optional)

The <**attributeLabel**> element contains a less ambiguous or less cryptic alternative identification than what is provided in <**attributeName**>. The value in <**attributeLabel**> is likely to be used as a column or row header in an HTML display.

12.1.5 Attribute storage type (optional)

A <**storageType**> element is optional, but is recommended and should contain a text field indicating the data type used to store the values of the attribute. For instance, decimal numbers may stored using data types called "float", "numeric", or "decimal" depending on the particular programming language (e.g. Python, R) or relational database system (e.g., PostgreSQL or MySQL) being used. The value in <**storageType**> may therefore be system- or application-specific if needed, such as to designate specific types for storage in a relational database system, but can also be used to provide a more general 'hint' to users or destination systems about how the attribute should be stored or represented. If there are no system-specific applications for the data (or they are unknown), we recommend choosing the best match from a small list of appropriate non-system-specific values, including *float*, *integer*, *string*, *boolean*, and *dateTime*. You may also choose from the values already provided by your EML preparation system if that is more convenient. Perhaps most importantly, do not indicate a type that is completely wrong, such as *integer* when the attribute clearly stores text values.

12.1.6 Missing value codes (optional)

The <**missingValueCode**> is optional, but it is strongly recommended to include it for any attribute that contains missing data values indicated with a particular code (e.g. NA, NaN, ND, -9999). Like enumerated domains above, this element should contain a <**code**> and <**definition**> child element to describe the missing value code and its meaning. The missing value code is a string, not a value, which means that the content of this field must exactly match what appears in place of data values for it to be correctly interpreted. For example, if data are output with precision .01 and with missing values formatted to "-9999.00," then the content of the <**missingValueCode**> element must be "-9999.00" not "-9999."

12.1.7 Annotation (optional)

Any <**attribute**> element may contain an optional <**annotation**> element, or can be referenced by annotations elsewhere in EML via their @**id** attribute. Annotations add additional context about the measurement or variable by linking to concepts or terms from an ontology. Annotations are described more fully in Chapter 7, and Example 7.7 gives an example of annotations used within an <**attribute**>. Though few use-cases have been fully developed for doing this to date, future custom units functionality in EML will almost certainly rely on annotations within <**attribute**> elements.

12.2 Examples

Example 11.1 has examples of many of the required, recommended, and optional metadata described above. For additional examples see those provided the spatialRaster and spatialVector sections of Appendix A.

Example 11.1: An <**attributeList**> element with six <**attribute**> children, as would be included to describe a tabular data entity. Variables described include a nominal text attribute (site identifier), a ratio attribute using dimensionless units (pH), a ordinal attribute with enumerated numeric values (a categorical variable), a dateTime attribute for year, an integer interval attribute for individual counts, and a ratio attribute with conductivity units. Note that several <**attribute**> elements have @**id** attributes that would be required for annotation. There are also @**typeSystem** attributes for some <**storageType**> elements that link the column data types to those defined in the XML schema (defined by www.w3.org). Also note that attribute 8, the "cond" column, uses a <**customUnit**>. This unit is described in the chapter on <**additionalMetadata**> (Chapter 12) in Example 12.1.

```
<attributeList>
  <attribute id="soil_chemistry.site_id">
    <attributeName>site_id</attributeName>
```

```
    <attributeDefinition>Site identifier as used in sites
↪   table</attributeDefinition>
    <storageType typeSystem="http://www.w3.org/2001/XMLSchema-datatypes">
      string
    </storageType>
    <measurementScale>
      <nominal>
        <nonNumericDomain>
          <textDomain>
            <definition>Site identifier text</definition>
          </textDomain>
        </nonNumericDomain>
      </nominal>
    </measurementScale>
  </attribute>
  <attribute id="soil_chemistry.pH">
    <attributeName>pH</attributeName>
    <attributeDefinition>ph of soil solution</attributeDefinition>
    <storageType typeSystem="http://www.w3.org/2001/XMLSchema-datatypes">
      float
    </storageType>
    <measurementScale>
      <ratio>
        <unit>
          <standardUnit>dimensionless</standardUnit>
        </unit>
        <precision>0.01</precision>
        <numericDomain>
          <numberType>real</numberType>
        </numericDomain>
      </ratio>
    </measurementScale>
  </attribute>
  <attribute id="pass2001.q110">
    <attributeName>q110</attributeName>
    <attributeDefinition>Q110-Preference for front yard
↪   landscape</attributeDefinition>
    <storageType typeSystem="http://www.w3.org/2001/XMLSchema-datatypes">
      integer
    </storageType>
    <measurementScale>
      <ordinal>
        <nonNumericDomain>
```

```
        <enumeratedDomain>
          <codeDefinition>
            <code>1</code>
            <definition>1-A desert landscape</definition>
          </codeDefinition>
          <codeDefinition>
            <code>2</code>
            <definition>2-Mostly lawn</definition>
          </codeDefinition>
          <codeDefinition>
            <code>3</code>
            <definition>3-Some lawn</definition>
          </codeDefinition>
        </enumeratedDomain>
      </nonNumericDomain>
    </ordinal>
  </measurementScale>
 </attribute>
 <attribute id="att.2">
  <attributeName>Year</attributeName>
  <attributeDefinition>Calendar year of the observation from years 1990 -
↪ 2010</attributeDefinition>
  <storageType>dateTime</storageType>
  <measurementScale>
    <dateTime>
      <formatString>YYYY</formatString>
      <dateTimePrecision>1</dateTimePrecision>
      <dateTimeDomain>
        <bounds>
          <minimum exclusive="false">1993</minimum>
          <maximum exclusive="false">2003</maximum>
        </bounds>
      </dateTimeDomain>
    </dateTime>
  </measurementScale>
 </attribute>
 <attribute id="att.7">
  <attributeName>Count</attributeName>
  <attributeDefinition>Number of individuals observed</attributeDefinition>
  <storageType>integer</storageType>
  <measurementScale>
    <interval>
      <unit>
```

```
          <standardUnit>number</standardUnit>
        </unit>
        <precision>1</precision>
        <numericDomain>
          <numberType>whole</numberType>
          <bounds>
            <minimum exclusive="false">0</minimum>
          </bounds>
        </numericDomain>
      </interval>
    </measurementScale>
    <missingValueCode>
      <code>NaN</code>
      <codeExplanation>value not recorded or invalid</codeExplanation>
    </missingValueCode>
  </attribute>
  <attribute id="att.8">
    <attributeName>cond</attributeName>
    <attributeLabel>Conductivity</attributeLabel>
    <attributeDefinition>measured with SeaBird Elecronics
↪   CTD-911</attributeDefinition>
    <storageType>float</storageType>
    <measurementScale>
      <ratio>
        <unit>
          <customUnit>siemensPerMeter</customUnit>
        </unit>
        <precision>0.0001</precision>
        <numericDomain>
          <numberType>real</numberType>
          <bounds>
            <minimum exclusive="false">0</minimum>
            <maximum exclusive="false">40</maximum>
          </bounds>
        </numericDomain>
      </ratio>
    </measurementScale>
  </attribute>
</attributeList>
```

12.3 XPaths referenced in this chapter

Attribute list: **/eml:eml/dataset/dataTable/attributeList**

An attribute: **/eml:eml/dataset/dataTable/attributeList/attribute**

An attribute's ID attribute: **/eml:eml/dataset/dataTable/attributeList/attribute/@id**

An attribute's name: **/eml:eml/dataset/dataTable/attributeList/attribute/attributeName**

An attribute definition: **/eml:eml/dataset/dataTable/attributeList/attribute/attributeDefinition**

Nominal attribute: **/eml:eml/dataset/dataTable/attributeList/attribute/nominal**

Standard numeric unit: **/eml:eml/dataset/dataTable/attributeList/attribute/ratio/unit/standardUnit**

Custom numeric unit: **/eml:eml/dataset/dataTable/attributeList/attribute/ratio/unit/customUnit**

Missing value code: **/eml:eml/dataset/dataTable/attributeList/attribute/missingValueCode**

13 Additional metadata

Key points

- The **additionalMetadata** element can contain **customUnit** specifications, annotations, and EML workflow indicators.
- The **additionalMetadata** element has no schema defined other than to contain well-formed XML, so EML creators have developed a range of applications for this element.

The <**additionalMetadata**> element is a flexible field for including any other relevant metadata that pertains to the resource being described with EML. It may be used for extending EML to include metadata that is not already available in another part of the EML specification, or to include site- or system-specific extensions that are needed beyond the core metadata. There may be any number of <**additionalMetadata**> elements included as children of the root of an EML document (**/eml:eml/additionalMetadata**) and each must contain valid XML within two available child elements. The <**describes**> element is optional, and, if used, should contain a string identical to the @**id** attribute of an EML element elsewhere in the document. The <**metadata**> child element is required and is a generic container for any valid (or "Well-formed") XML metadata to be included in the document. When both of these child elements are present, the content of a <**metadata**> element therefore describes the EML element referenced in the <**describes**> element preceding it. If the <**describes**> element is omitted, it is assumed that the <**additionalMetadata**> content applies to the entire EML document.

Though there is significant flexibility in how to create and use <**additionalMetadata**> elements, there are also some common use cases that require particular child elements. These use cases, and other considerations for using the <**additionalMetadata**> element are below.

13.1 Custom units

Any numerical variables (aka attributes) in a data entity that have a custom unit of measurement defined (such as in **attribute/measurementScale/ratio/unit/customUnit**), as described in Chapter 11, must have their custom unit described in the <**additionalMetadata**> tree. These descriptions must be created using XML that follows the convention laid out in the

STMML schema (defined in this paper). The schema requires that these EML custom units must appear in a list of <**stmml:unit**> elements nested within the <**stmml:unitList**> container element (**additionalMetadata/metadata/stmml:unitList/stmml:unit**). Each <**stmml:unit**> element in the list must have an @**id** attribute that matches the unit name in the associated <**customUnit**> element. A range of other attributes can also be used in <**stmml:unit**> including @**abbreviation**, @**parentSI**, @**multiplierToSI**, and @**unitType**. Each <**stmml:unit**> element also requires a description element (<**stmml:description**>) containing a text description of the unit. While the EML schema explicitly states that the STMML schema should be imported into an `stmml` namespace, this is a lax requirement that is not checked during validation, and it is commonly left out of EML documents.

Example 12.1: An <**additionalMetadata**> element containing an STMML <**unitList**> with one <**unit**>. This unit being described is the <**customUnit**> defined in Example 11.1 in Chapter 11.

```
<additionalMetadata>
  <metadata>
    <stmml:unitList>
      <stmml:unit id="siemensPerMeter" name="siemensPerMeter"
        unitType="conductance" parentSI="siemen" multiplierToSI="1">
        <stmml:description>
          conductivity over distance unit
        </stmml:description>
      </stmml:unit>
    </stmml:unitList>
  </metadata>
</additionalMetadata>
```

13.2 Annotations

Annotation elements (<**annotation**>) may be included within <**additionalMetadata**> and some annotation use cases may require placement in this location to function. Often, annotations placed here will be used to describe elements elsewhere in the EML document using referenced @**id** attributes in the <**describes**> child element. More on annotations, including placement in <**additionalMetadata**>, is covered in Chapter 7.

13.3 EML workflow indicators

Some EML creation tools or workflows insert XML metadata into <additionalMetadata> to indicate the origin and build status of the EML document. For example, the ezEML metadata

editor application inserts the application version and any EML templates used to create the EML document, as shown in Example 12.2.

Example 12.2: <**additionalMetadata**> elements added to an EML document by the ezEML metadata editor.

```
<additionalMetadata>
  <metadata>
    <importedFromXML dateImported="2023-01-30" filename="FRS_template.xml"/>
  </metadata>
</additionalMetadata>
<additionalMetadata>
  <metadata>
    <emlEditor app="ezEML" release="2023.11.29"/>
  </metadata>
</additionalMetadata>
```

13.4 Other use cases

Because the <**additionalMetadata**> element has no schema defined other than to contain well-formed XML, EML creators (researchers, data managers, etc.) have developed a range of applications for this element. Some of these use cases expand on existing elements of the EML schema, such as adding structured project information not easily added to the <**project**> tree. Some are data management related, including workflow indicators or dataset categorizations. Below are a few of these use cases with links to published datasets that serve as examples.

- Project management information specific to a research site or program, such as permitting information associated with the research. See examples in this FCE LTER dataset.
- Data-management related keywords or dataset classification indicators such as those seen in this VCR LTER dataset.
- Data usage information
- Extended site descriptions, research context, or data access instructions
- Anything else your heart desires (expressed in valid XML)

13.5 XPaths referenced in this chapter

Additional metadata element: **/eml:eml/additionalMetadata**

Additional metadata "describes" element: **/eml:eml/additionalMetadata/describes**

Additional metadata annotation: **/eml:eml/additionalMetadata/metadata/annotation**

STMML unit list: **/eml:eml/additionalMetadata/metadata/stmml:unitList**

Custom unit: **/eml:eml/additionalMetadata/metadata/stmml:unitList/stmml:unit**

Custom unit ID attribute: **...additionalMetadata/metadata/stmml:unitList/stmml:unit/@id**

Custom unit description: **...additionalMetadata/metadata/stmml:unitList/unit/description**

Appendix A. Specialized elements

This appendix covers "specialized" EML elements that are not recommended for most EML applications. These elements are not deprecated or obsolete, but they do have limited use-cases and are rarely used. Best practices for their use have not been determined. Much of the text and most examples come from earlier versions of the EML Best Practices document.

Geographic Coverage

There are multiple EML elements that can be used to describe the geographic coverage of a dataset. The highest priority is to define the geographic coverage at the dataset level as described in Chapter 7. Less commonly used are geographic coverage elements within methods and more complex boundaries described by datasetGPolygon, both described below.

methods/spatialSamplingUnits/<geographicCoverage>

In the dataset/methods element, individual sampling sites may be entered under <**spatialSamplingUnits**>, each site in a separate coverage element (see below).

Example A.1: <**geographicCoverage**> under <**spatialSamplingUnits**>

```
<spatialSamplingUnits>
  <coverage>
    <geographicDescription>sitenumber 1</geographicDescription>
    <boundingCoordinates>
      <westBoundingCoordinate>-112.2</westBoundingCoordinate>
      <eastBoundingCoordinate>-112.2</eastBoundingCoordinate>
      <northBoundingCoordinate>33.5</northBoundingCoordinate>
      <southBoundingCoordinate>33.5</southBoundingCoordinate>
    </boundingCoordinates>
  </coverage>
  <coverage>
    <geographicDescription>sitenumber 2</geographicDescription>
    <boundingCoordinates>
      <westBoundingCoordinate>-111.7</westBoundingCoordinate>
```

```
      <eastBoundingCoordinate>-111.7</eastBoundingCoordinate>
      <northBoundingCoordinate>33.6</northBoundingCoordinate>
      <southBoundingCoordinate>33.6</southBoundingCoordinate>
    </boundingCoordinates>
  </coverage>
  <coverage>
    <geographicDescription>sitenumber 3</geographicDescription>
    <boundingCoordinates>
      <westBoundingCoordinate>-112.1</westBoundingCoordinate>
      <eastBoundingCoordinate>-112.1</eastBoundingCoordinate>
      <northBoundingCoordinate>33.7</northBoundingCoordinate>
      <southBoundingCoordinate>33.7</southBoundingCoordinate>
    </boundingCoordinates>
  </coverage>
</spatialSamplingUnits>
```

<datasetGPolygon>

The <**datasetGPolygon**> element may be included when the required bounding box does not adequately describe the study location, for example, if an irregular polygon is necessary to describe the study area, or there is an area within the bounding box that is excluded. This element is optional, and has two child elements.

<**datasetGPolygonOuterGRing**>: This is the outer part of the polygon shape that encompasses the broadest area of coverage. It can be created either by a gRing (list of points) or 4 or more <**gRingPoint**>s. Documentation for an FGDC G-Ring states that four points are required to define a polygon, and the first and last should be identical. However this is not enforceable in XML Schema, and so in EML a minimum of three <**gRingPoint**>s is required to define the polygon, and it can be assumed that since a polygon is closed, the last point can be joined to the first.

The <**datasetGPolygonExclusionGRing**> is the closed, non-intersecting boundary of a void area (or hole in an interior area). This could be the center of the doughnut shape created by the <**datasetGPolygon**>. It can be created either by a gRing (list of points) or one or more <**gRingPoint**>s. This is used if there is an internal polygon to be excluded from the outer polygon, e.g, a lake to be excluded from the broader geographic coverage.

> EDI context note
>
> It is somewhat rare for repository systems to display complex geographic coverage elements. EDI, for example, does not display <**datasetGPolygon**> elements in dataset landing pages.

Taxonomic coverage

<taxonomicSystem> child elements

The optional **taxonomicCoverage/taxonomicSystem** trees may be used to detail the use of taxonomic identification resources and on the identification process. <**classificationSystem**> should be used to list authoritative taxonomic databases (such as ITIS, IPNI, NCBI, Index Fungorum, or USDA Plants) or classification systems used for taxonomic identification. Documentation and relevant literature regarding, used authoritative sources, including URL's pointing to these sources, should be listed in <**classificationSystemCitation**>. Exceptions to, or deviation from, used authoritative sources should be explained in <**classificationSystemModification**>.

Methods and protocols used for taxonomic classification should be detailed using the <**identifierName**> and <**taxonomicProcedures**> tags. Examples of methods that should be listed in <**taxonomicProcedures**> are details of specimen processing, keys, and chemical or genetic analyses. <**taxonomicCompleteness**> may be used to document the status, estimated importance, and reason for incomplete identifications.

Example A.2: Taxonomic system

```
<taxonomicSystem>
  <classificationSystem>
    <classificationSystemCitation>
      <title>Integrated Taxonomic Information System (ITIS)</title>
      <creator>
        <organizationName>
          Integrated Taxonomic Information System
        </organizationName>
        <onlineUrl>http://www.itis.gov/</onlineUrl>
      </creator>
      <generic>
        <publisher>
          <organizationName>
            Integrated Taxonomic Information System
          </organizationName>
          <onlineUrl>http://www.itis.gov/</onlineUrl>
        </publisher>
      </generic>
    </classificationSystemCitation>
  </classificationSystem>
  <identifierName>
    <references>pers-1</references>
```

```
  </identifierName>
  <taxonomicProcedures>
    All individuals were identified and stored in alcohol, except
    for one voucher specimen for each species, which was tagged and
    pinned.
  </taxonomicProcedures>
</taxonomicSystem>
```

Optional <methods> child elements

<sampling>

This optional tree can contain very specific information about the study site and associated sampling locations and frequencies. However, we recommend that descriptive geographic and temporal coverage elements should also be available at the dataset level to ensure dataset users have all information at their fingertips. The <**studyExtent**> child element provides specific information about the temporal and geographic extent of the study such as domains of interest in addition to geographic, temporal, and taxonomic coverage of the study site. This information can be provided as either simple text using <**description**> or by including detailed temporal or geographic <**coverage**> elements describing discrete time periods sampled or the sub-regions sampled within the overall geographic bounding box that was described at the <**dataset**> level. The <**samplingDescription**> element is an alternative TextType element available as a child to <**sampling**> that may be formatted similarly to the sampling methods section of a journal article.

<qualityControl>

The optional <**qualityControl**> element describes actions taken to either control or assess the quality of data resulting from the methods used to create the dataset. For a basic description under <**qualityControl**>, use the <**description**> element. The <**citation**> and <**protocol**> elements are also available to define detailed QA/QC protocols, but keep in mind that referencing external sources may fail in the future.

Example A.3: A methods element with some optional child elements <**sampling**> and <**qualityControl**>. Note the **sampling/spatialSamplingUnits/geographicCoverage** element.

```
<methods>
  <methodStep>
    <description>
      <section>
        <title>
          Pitfall trap sampling for ground arthropod biodiversity monitoring
        </title>
        <para>Supplies used: pitfall traps (P-16 plastic Solo cups with lids)
↪
          metal spades and large bulb planters (to dig holes in which to
          put traps) 70% ethanol (to preserve specimens) Qorpak glass jars
          with lids from the VWR Corporation, 120ml (4oz), cap size 58-400
          (comes included), Qorpak no. 7743C, VWR catalog no.
          16195-703.</para>
        <para>Between 10 and 21 traps are placed at each site in suitable
          location.</para>
        <para>All trapped taxa counted and measured (body length), most taxa
          identified to Family, ants to Genus</para>
      </section>
    </description>
    <instrumentation>SBE MicroCAT 37-SM (S/N 1790); manufacturer:
      Sea-Bird Electronics (model: 37-SM MicroCAT); parameter: Conductivity
      (accuracy: 0.0003 S/m, readability: 0.00001 S/m, range: 0 to 7 S/m);
      last calibration: Feb 28, 2001</instrumentation>
    <instrumentation>SBE MicroCAT 37-SM (S/N 1790); manufacturer:
      Sea-Bird Electronics (model: 37-SM MicroCAT); parameter: Pressure
      (water) (accuracy: 0.2m, readability: 0.0004m, range: 0 to 20m); last
      calibration: Feb 28, 2001</instrumentation>
    <instrumentation>SBE MicroCAT 37-SM (S/N 1790); manufacturer:
      Sea-Bird Electronics (model: 37-SM MicroCAT); parameter: Temperature
      (water) (accuracy: 0.002°C, readability: 0.0001°C, range: -5 to 35°C);
      last calibration: Feb 28, 2001</instrumentation>
  </methodStep>
  <sampling>
    <studyExtent>
      <description>
        <para>Arthropod pit fall traps are placed in three different
          locations four times a year</para>
      </description>
    </studyExtent>
    <samplingDescription>
      <para>Six traps were set in a transect at each location.</para>
    </samplingDescription>
```

```
    <spatialSamplingUnits>
      <coverage>
        <geographicDescription>site number 1</geographicDescription>
        <boundingCoordinates>
          <westBoundingCoordinate>-112.234566</westBoundingCoordinate>
          <eastBoundingCoordinate>-112.234566</eastBoundingCoordinate>
          <northBoundingCoordinate>33.534566</northBoundingCoordinate>
          <southBoundingCoordinate>33.534566</southBoundingCoordinate>
        </boundingCoordinates>
      </coverage>
      <coverage>
        <geographicDescription>site number 2</geographicDescription>
        <boundingCoordinates>
          <westBoundingCoordinate>-111.745677</westBoundingCoordinate>
          <eastBoundingCoordinate>-111.745677</eastBoundingCoordinate>
          <northBoundingCoordinate>33.64577</northBoundingCoordinate>
          <southBoundingCoordinate>33.64577</southBoundingCoordinate>
        </boundingCoordinates>
      </coverage>
      <coverage>
        <geographicDescription>site number 3</geographicDescription>
        <boundingCoordinates>
          <westBoundingCoordinate>-112.167899</westBoundingCoordinate>
          <eastBoundingCoordinate>-112.16799</eastBoundingCoordinate>
          <northBoundingCoordinate>33.76799</northBoundingCoordinate>
          <southBoundingCoordinate>33.76799</southBoundingCoordinate>
        </boundingCoordinates>
      </coverage>
    </spatialSamplingUnits>
  </sampling>
  <qualityControl>
    <description>
      <para>All specimens are archived for future reference. Quality
        control during data entry is achieved with standard database
        techniques of pulldowns that prevent typos and constraints.
        Scientists inspect standard data summary statistics after data
        entry.</para>
    </description>
  </qualityControl>
</methods>
```

Example A.4: <**methods**>, with <**dataSource**>

```
<methods>
  <methodStep>
    <description>
      <section>
        <para>We utilize NPP data collected from 1906 to 2006 from the ONL
          LTER site. The ONL NPP data unit definition is kg/m\^2/yr. This
↪
          unit does not require conversion.</para>
      </section>
    </description>
    <dataSource>
      <title>NPP data from ONL 1906 to 2006</title>
      <creator>
        <organizationName>ONL LTER</organizationName>
      </creator>
      <distribution>
        <online>
          <onlineDescription>This online link references an EML document that
↪  describes data used in the creation of this derivative data package.
          </onlineDescription>
          <url function="information">
            https://pasta.lternet.edu/package/metadata/eml/edi/15/5
          </url>
        </online>
      </distribution>
      <contact>
        <organizationName>ONL LTER</organizationName>
        <positionName>ONL Information Manager</positionName>
        <electronicMailAddress>im@onl.lternet.edu</electronicMailAddress>
      </contact>
    </dataSource>
  </methodStep>
</methods>
```

SpatialRaster & SpatialVector and their attributes

Some examples from the earlier attributes section:

The examples below show complete entity trees for <**spatialVector**> and <**spatialRaster**> converted via XSLT (stylesheet) from Esri metadata format.

Example A.5: Entity and attribute information for spatialVector

```
<spatialVector id="Landuse for Ficity in 1955">
  <entityName>Landuse for Ficity in 1955</entityName>
  <entityDescription>This GIS layer represents a reconstructed
    generalized landuse map for the area of current Ficity around the time
    period of 1955.</entityDescription>
  <physical>
    <objectName>frs-20.zip</objectName>
    <dataFormat>
      <externallyDefinedFormat>
        <formatName>Esri Shapefile (zipped)</formatName>
      </externallyDefinedFormat>
    </dataFormat>
    <distribution>
      <online>
        <onlineDescription>f1s-20 Zipped Shapefile File</onlineDescription>
        <url function="download">http://www.fsu.edu/frs/data/frs-20.zip</url>
      </online>
    </distribution>
  </physical>
  <attributeList>
    <attribute>
      <attributeName>FID</attributeName>
      <attributeDefinition>Internal feature number.</attributeDefinition>
      <storageType typeSystem="http://www.esri.com/metadata/esriprof80.html">
        OID
      </storageType>
      <measurementScale>
        <nominal>
          <nonNumericDomain>
            <textDomain>
              <definition>
                Sequential unique whole numbers that are automatically
                generated.
              </definition>
            </textDomain>
          </nonNumericDomain>
        </nominal>
      </measurementScale>
    </attribute>
    <attribute>
      <attributeName>Shape</attributeName>
      <attributeDefinition>Feature geometry.</attributeDefinition>
      <storageType typeSystem="http://www.esri.com/metadata/esriprof80.html">
```

```
      geometry
    </storageType>
    <measurementScale>
      <nominal>
        <nonNumericDomain>
          <textDomain>
            <definition>Coordinates defining the features.</definition>
          </textDomain>
        </nonNumericDomain>
      </nominal>
    </measurementScale>
  </attribute>
  <attribute>
    <attributeName>Z955</attributeName>
    <attributeDefinition>
      This field signifies the landuse value for each polygon.
    </attributeDefinition>
    <storageType typeSystem="http://www.w3.org/2001/XMLSchema-datatypes">
      string
    </storageType>
    <measurementScale>
      <nominal>
        <nonNumericDomain>
          <enumeratedDomain>
            <codeDefinition>
              <code>Agriculture</code>
              <definition>Agricultural land use</definition>
            </codeDefinition>
            <codeDefinition>
              <code>Urban</code>
              <definition>Urbanized area</definition>
            </codeDefinition>
            <codeDefinition>
              <code>Desert</code>
              <definition>Unmodified area</definition>
            </codeDefinition>
            <codeDefinition>
              <code>Recreation</code>
              <definition>Recreational land use</definition>
            </codeDefinition>
          </enumeratedDomain>
        </nonNumericDomain>
      </nominal>
```

```
      </measurementScale>
    </attribute>
  </attributeList>
  <geometry>Polygon</geometry>
  <geometricObjectCount>78</geometricObjectCount>
  <spatialReference>
    <horizCoordSysName>NAD_1927_UTM_Zone_12N</horizCoordSysName>
  </spatialReference>
</spatialVector>
```

Example A.6: Entity and attribute information for spatialRaster

```
<spatialRaster id="fi_24k">
  <entityName>fi_24k</entityName>
  <entityDefinition>
    Fictitious State 7.5 Minute Digital Elevation Model
  </entityDefinition>
  <physical>
    <objectName>frs-30.zip</objectName>
    <dataFormat>
      <externallyDefinedFormat>
        <formatName>Esri binary grid</formatName>
      </externallyDefinedFormat>
    </dataFormat>
    <distribution>
      <online>
        <onlineDescription>frs-30 zipped raster data File</onlineDescription>
        <url function="download">http://www.fsu.edu/frs/data/frs-30.zip</url>
      </online>
    </distribution>
  </physical>
  <attributeList>
    <attribute>
      <attributeName>ObjectID</attributeName>
      <attributeDefinition>Internal feature number.</attributeDefinition>
      <storageType typeSystem="http://www.esri.com/metadata/esriprof80.html">
        OID
      </storageType>
      <measurementScale>
        <nominal>
          <nonNumericDomain>
            <textDomain>
```

```
            <definition>
              Sequential unique whole numbers that are automatically
              generated.
            </definition>
          </textDomain>
        </nonNumericDomain>
      </nominal>
    </measurementScale>
  </attribute>
  <attribute>
    <attributeName>Cell Value</attributeName>
    <attributeDefinition>Elevation Value</attributeDefinition>
    <storageType typeSystem="http://www.esri.com/metadata/esriprof80.html">
      Integer
    </storageType>
    <measurementScale>
      <ratio>
        <unit>
          <standardUnit>meter</standardUnit>
        </unit>
        <precision />
        <numericDomain>
          <numberType>integer</numberType>
          <bounds>
            <minimum exclusive="true">-5193.000000</minimum>
            <maximum exclusive="true">14785.000000</maximum>
          </bounds>
        </numericDomain>
      </ratio>
    </measurementScale>
  </attribute>
  <attribute>
    <attributeName>Count</attributeName>
    <attributeDefinition>Count</attributeDefinition>
    <storageType typeSystem="http://www.esri.com/metadata/esriprof80.html">
      Integer
    </storageType>
    <measurementScale>
      <ratio>
        <unit>
          <standardUnit>number</standardUnit>
        </unit>
        <precision />
```

```
          <numericDomain>
            <numberType>whole</numberType>
          </numericDomain>
        </ratio>
      </measurementScale>
    </attribute>
  </attributeList>
  <spatialReference>
    <horizCoordSysName>NAD_1927_UTM_Zone_12N</horizCoordSysName>
  </spatialReference>
  <horizontalAccuracy>not available</horizontalAccuracy>
  <verticalAccuracy>not available</verticalAccuracy>
  <cellSizeXDirection>30.0</cellSizeXDirection>
  <cellSizeYDirection>30.0</cellSizeYDirection>
  <numberOfBands>1</numberOfBands>
  <rasterOrigin>Upper Left</rasterOrigin>
  <rows>21092</rows>
  <columns>18136</columns>
  <verticals>1</verticals>
  <cellGeometry>matrix</cellGeometry>
</spatialRaster>
```

Tables describing all entity types

Table A.1 Summary of the six entities in EML, including the type of data entity typically described with that element, how they are created and a brief description of its metadata.

Element name	Used for	Created from	Metadata features
dataTable	Fixed-structure tabular data objects (csv, database table, etc.)	export from code, RDBMS or spreadsheets	columns/rows named and defined, e.g., measurement and storage typing
otherEntity	Data objects not described by other standard entity types (images, non-tabular data, binary files, etc.)	applications	type of entity

Element name	Used for	Created from	Metadata features
spatialRaster	Gridded data, raster cell data, remote sensing data	applications, stylesheet conversions. See "Other Resources"	spatial organization of the raster cells, their data values, and if derived via imaging sensors, characteristics about the image and its individual bands
spatialVector	Lines, points polygons, KML (if converted), ESRI shape files	applications, stylesheet conversions. See "Other Resources"	information about the vector's geometry type, count, attributes, and topology level
view	Data returned from a database query	RDBMS	Recommended to handle as data table (above).
storedProcedure	Data returned from a stored procedure in a database	RDBMS	Recommended to handle as data table (above)

Table A.2. Elements specific to each of the six entity types.

Entity Type	Typical Uses	Data entity child elements
<**dataTable**>	Tabular data objects with a fixed structure (delimited text files, simple spreadsheet, etc.)	**EntityGroup** (<**entityName**> required, others recommended or optional)<**attributeList**> (required)<**constraint**><**caseSensitivity**><**numberOfRecor**
<**view**>	Data returned from a database query	**EntityGroup** (<**entityName**> required, others recommended or optional)<**attributeList**> (required)<**constraint**><**queryStatement**> (required)
<**storedProcedure**>	Data returned from a stored procedure in a database	**EntityGroup** (<**entityName**> required, others recommended or optional)<**attributeList**> (required)<**constraint**><**parameter**>
<**otherEntity**>	Data objects not described by other standard entity types (images, non-tabular data, binary files, etc.)	**EntityGroup** (<**entityName**> required, others recommended or optional)<**attributeList**><**constraint**><**entityType**> (required)

Entity Type	Typical Uses	Data entity child elements
<**spatialRaster**>	Gridded data, raster cell data, remote sensing data	**EntityGroup** (<**entityName**> required, others recommended or optional)<**attributeList**> (required)<**constraint**><**spatialReference**> (required)<**georeferenceInfo**><**horizontalAccuracy**> (required)<**verticalAccuracy**> (required)<**cellSizeYDirection**> (required)<**numberOfBands**> (required)<**rasterOrigin**> (required)<**rows**> (required)<**columns**> (required)<**verticals**> (required)<**cellGeometry**> (required)<**toneGradation**><**scaleFactor**><**offset**><**imageI**
<**spatialVector**>	Lines, points polygons, KML (if converted), ESRI shape files	**EntityGroup** (<**entityName**> required, others recommended or optional)<**attributeList**> (required)<**constraint**><**geometry**> (required)<**geometricObjectCount**><**topologyLevel**><**spat**

<constraint>

This element tree is found at (XPath):

/eml:eml/dataset/dataTable/constraint

/eml:eml/dataset/view/constraint

/eml:eml/dataset/spatialRaster/constraint

/eml:eml/dataset/spatialVector/constraint

/eml:eml/dataset/storedProcedure/constraint

The <**constraint**> tree is for describing any integrity constraints between entities within a data package (e.g. tables), as they would be maintained in a relational management system. Use of the <**constraint**> tree is encouraged when data elements contain integrity constraints from a relational database. Example TO-DO shows the constraints for the <**attributeList**> in Example TO-DO. If there are constraints in which several columns are involved, these should be described in methods/qualityControl, since EML is not currently equipped to handle keys defined by multiple columns. When the <**constraint**> tree is used, all of the entities that may be referenced should be in the same package. There are six child elements:

<**primaryKey**> is an element which declares the primary key in the entity to which the defined constraint pertains.

<**uniqueKey**> is an element which represents a unique key within the referenced entity. This is different from a primary key in that it does not form any implicit foreign key relationships to other entities; however it is required to be unique within the entity.

<**nonNullConstraint**> defines a constraint that indicates that no null values should be present for an attribute in this entity.

<**checkConstraint**> defines a constraint which checks a conditional clause within an entity.

<**foreignKey**> defines an SQL statement or other language implementation of the condition for a check constraint. Generally this provides a means for constraining the values within and among entities. It also provides the means to meaningfully link table for explanation of codes (de-normalization).

<**joinCondition**> defines a foreign key relationship among entities which relates this entity to another's primary key.

The <**primaryKey**>, <**uniqueKey**>, <**nonNullConstraint**> require an additional <**key**> tag defining the attribute to which this constraint applies, referenced by its id attribute (described in another area). All **ConstraintType** entities require additional <**constraintName**> and <**attributeReference**> tags.

Example A.7: constraint

```
<constraint id="soil_chemistry.PRIMARY">
  <primaryKey>
    <constraintName>PRIMARY</constraintName>
    <key>
      <attributeReference>soil_chemistry.ID</attributeReference>
    </key>
  </primaryKey>
</constraint>
<constraint id="soil_chemistry.FK_soil_chemistry_sites">
  <foreignKey>
    <constraintName>FK_soil_chemistry_sites</constraintName>
    <key>
      <attributeReference>soil_chemistry.site_id</attributeReference>
    </key>
    <entityReference>sites</entityReference>
  </foreignKey>
</constraint>
```

Appendix B. XML, schemas, and EML

What is XML?

The Extensible Markup Language, or XML, is a markup language for encoding documents in a format that is both human-readable and machine-readable. XML is used in all sorts of application domains. The Ecological Metadata Language, or EML, is one such application for working with ecological and environmental data.

Tags, elements, and attributes

Every XML document has a structure defined by the presence of XML *tags*. Tags have names that are enclosed in angle brackets, and all tags must be paired. The first tag in a pair is the start tag (<**tag**>) and the second, with its name prefixed by a backslash (</**tag**>), is a closing tag. A pair of tags is used to form an *element*, which is the most fundamental unit of information encoded in an XML document. An element consists of a start tag, a closing tag, and all content in between them. In this document, we frequently refer to XML elements (and implicitly to their content) using the start tag in boldface type with angle brackets.

To add further structure to the information in an XML document, elements can be nested within other elements, giving XML documents a hierarchical, or tree-like, structure (also known as the Document Object Model, or DOM). The first element that encloses all other elements is referred to as the "root" of the tree, and all XML documents must have one root element. For example, an XML element called <**individualName**> might define a person's full name as in example B.1.

Example B.1: a simple XML element

```
<individualName>
  <givenName>Grace</givenName>
  <surName>O'Malley</surName>
</individualName>
```

This is a simple, but complete, XML tree with <**individualName**> as the root element. Note that hierarchically nested elements are often referred to using a family tree language. In the

example above, <**givenName**> is the child of the parent element <**individualName**>, and a sibling to the <**surName**> element.

Besides having a name, any XML element may also be assigned *attributes*. Attributes are a key-value statement defined inside an element's start tag, after the element name. If we wished to assign a type attribute to our <**individualName**> element, we might do something like example B.2.

Example B.2: a simple XML element with a @**type** attribute

```
<individualName type="pirate">
  <givenName>Grace</givenName>
  <surName>O'Malley</surName>
</individualName>
```

Keep in mind that we could accomplish the same thing by including the type as an element nested in <**individualName**> as in example B.3.

Example B.3: a simple XML element with individual type as a child element

```
<individualName>
  <givenName>Grace</givenName>
  <surName>O'Malley</surName>
  <individualType>pirate</individualType>
</individualName>
```

There are no requirements or rules for using attributes in XML, but they can be useful for attaching additional data to an element. Attributes are used in several places in EML, and their use is defined in the EML schema (which we'll talk about in a moment).

Navigating XML documents with XPath

As noted above, the tree-like structure of XML documents means that any element can be a branch (if it contains more elements) or a leaf (if it is an atomic value). Elements in this hierarchy are often referred to as nodes and each can contain attributes, text, or other values. These nodes and associated values can be referenced and navigated using a convention called `XPath`, which is similar to the way we describe computer file system paths. Nodes are referred to by name and are separated by forward slashes (/), starting with the top-level, or most inclusive, note on the left. For example, in the <**individualName**> examples from the section above, if we wished to select the individual type we would use the XPaths "**/individualName/@type**" if it were defined as an attribute, and "**/individualName/individualType**" if it were defined as a child element if <**individualName**>.

Namespace attributes

Namespace attributes are a special kind of XML attribute that are used to make a defined vocabulary of elements, or namespace, available in an XML document. A namespace attribute usually also assigns a prefix that refers to the namespace in named elements. Namespace prefixes are commonly used to avoid conflicts between elements that have different content or purposes, but similar names. For instance, in an XML application that used both customer and product tables, the two table elements could be distinguished with namespaces: <**c:table**> and <**p:table**>. To use the **c** and **p** namespace prefixes, in an XML element, the namespaces must be imported and assigned those prefixes using a namespace attribute.

Namespace attributes have a two-part key that starts with **xmlns** (for XML namespace), followed by a colon and then the prefix that will identify elements from the given namespace. Most XML namespaces are designed for particular applications or user communities and the definition of the namespace and its elements are kept in an agreed upon location. Usually this location is a URL, which becomes the value for the namespace attribute. To make the EML namespace available under the **eml** prefix, for example, one must include the attribute **xmlns:eml="https://eml.ecoinformatics.org/eml-2.2.0"** in the top-level element of an EML document.

Escaping special characters

As a structured text markup language, XML must treat certain characters as special. Most importantly, the less-than sign (<) is special because it begins an XML tag and the ampersand (&) is special because it begins something called an entity reference (see here). When these special characters appear as content in XML (i.e., as text between start and end tags, or in an attribute value), most XML parsers will interpret them as part of the XML structure. For example, the less-than sign in the text expression "one < two" would be interpreted as the start of an XML tag when parsed. The quotation, apostrophe, and greater-than signs (", ', and >) are also special characters, but they are misinterpreted much less often. To avoid errors, special characters can be "escaped" in one of two ways.

First, special characters can be encoded with distinct character sequences that XML parsers can understand without ambiguity. Notice that these escape sequences begin with the special ampersand. Escape encodings for the five special characters are:

- `<` encoded as `<`
- `&` encoded as `&`
- `'` encoded as `&apos;`
- `"` encoded as `&quot;`
- `>` encoded as `>`

Second, blocks of text containing special characters can be enclosed in a CDATA section, which begins with the `<![CDATA[` sequence and closes with `]]>`, such as in example B.4. This is a handy way to escape text that contains many special characters at once. Escaping special characters is not needed in every case, but it is well-worth remembering to escape < and & most of the time (see this SE answer for a concise summary: https://stackoverflow.com/a/46637835/290085).

Example 2.4: A CDATA section in which all text content between the opening and closing CDATA sequences (`<![CDATA[` and `]]>`), including the <**greeting**> tags, will be interpreted as character data instead of XML markup. Example taken from W3.org documentation (link).

```
<![CDATA[<greeting>Hello, world!</greeting>]]>
```

Other XML features

It is common to find a *declaration* at the beginning of XML documents. Declarations are enclosed in angle brackets with question marks, and most declarations that you will see, including in EML, look like this:

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

Declarations are not tags. They are used to hold metadata about the XML document itself.

Resources

The XML standard is described in:

- https://www.w3.org/XML/

Tutorials on XML

- https://www.w3schools.com/xml/
- https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction

What is an XML Schema?

An XML *schema* is a description of a specific type of XML document that is defined by rules about its form and the content it contains. An XML schema is written in a subset of XML called XML Schema Definition, or XSD. Any other XML documents that are written as instances of a particular XML schema must be able to be "validated" against the rules laid out in that schema's XSD file. To make an XML document into an instance of a particular schema,

the schema location and XSD file must be referenced using the **schemaLocation** attribute from the XML Schema Instance namespace (http://www.w3.org/2001/XMLSchema-instance, usually given the **xsi** prefix). Details about this attribute are in the "Overview of the EML schema" section and the examples below.

The EML standard consists of a series of XSD files collectively defining the structure of a valid EML document and the minimum content that it needs to contain. You can look at this schema definition on GitHub (https://github.com/NCEAS/eml/tree/main/xsd) if you like. The EML standard uses an XML schema because doing so enables users and applications to ensure the consistency and completeness of a metadata document. Many data repositories that accept EML metadata documents check schema compliance when datasets are deposited, and it is highly recommended that data managers using EML know the schema location and how to validate their documents' adherence to it.

XML Data types

All XML schemas are built using a hierarchy of defined element types, starting with those types that are built into XML itself. Built into XML are several element types to contain particular kinds of data, such as text (`xs:string`), decimal numbers (`xs:decimal`), and dates (`xs:date`). Using a series of rules defined in an XSD file, a more complex data type, referred to as a "`complexType`", can be built from these simpler types. For example, an <**individualName**> element might be defined using an XSD rule stating that it must contain the <**givenName**> and <**surName**> child elements, in that order, and that both must be `xs:string` data types, as in the rule in example B.4.

Example B.4: an XSD rule defining a complex type for an <**individualName**> element

```
<xs:element name="individualName">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="givenName" type="xs:string"/>
      <xs:element name="surName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

This is an example of a `complexType` being defined in an XML schema, and with this definition, an `individualName` element could be re-used throughout any document following this schema. When the document was validated, any `individualName` elements that were formatted differently, such as those missing a `surName` element, or with child elements out of order, would violate the schema. In this way, XML data types can be defined, nested, and built into very complex, and useful data structures. A number of data types are defined in the EML schema to contain and validate particularly useful elements of metadata.

Validating against a schema

There are a number of tools that can be used to validate an XML document against a schema defined in an XSD file. If the XSD location is provided in the XML document root (as described in the EML root element section below), then most tools will compare the document against that XSD without any need to access it separately. Elements that break the rule of the schema can be identified using warning messages

Resources

XML Schemas and XSD

- https://www.w3schools.com/xml/xml_schema.asp

Tools for validating XML against any schema

- Oxygen - a full featured XML editing software with built in schema validation (paid license)
- XML Copy Editor - a fast, free, validating XML editor
- Freeformatter.com - A website offering free XML validation tools

Overview of the EML schema

Like all XML documents, EML has a hierarchical structure. Because the EML standard is ultimately defined by an XML schema (or XSD), there are rules about what elements must be included, in which locations in the hierarchy, and what content they may and may not hold. A valid EML document must follow these rules. In this section we define the XML elements that are placed at the highest level of an EML document, and how they should be structured. We start with the "root" element, which encloses all others, and then define several top-level elements that may be placed directly inside the root. Some of the top-level elements are required and some are optional, and many have required or recommended attributes to consider.

The root element (<eml:eml>)

This <**eml:eml**> element is the root element in all EML documents, meaning that it is required and encloses all other elements. Other than any declarations present, the opening tag of this root element (<**eml:eml**>) should always come first in an EML document. Notice that the EML namespace is often immediately defined using the first attribute in this element (**xmlns:eml="https://eml.ecoinformatics.org/eml-2.2.0"**) though this may not be required by all applications using EML documents. The EML root element has three other

important and required attributes that are described below. An example EML root starting tag, with all these elements, is shown in Example B.5.

Schema location attribute

The @**xsi:schemaLocation** attribute is required and tells a processor (or person) that the XML document is an instance of the EML schema and where to find the XML schema file (XSD) to validate against. For an EML 2.2.0 document, this attribute should contain two URIs, one pointing to the EML namespace (https://eml.ecoinformatics.org/eml-2.2.0) and one pointing to the EML schema XSD (https://nis.lternet.edu/schemas/EML/eml-2.2.0/eml.xsd).

Package identifier attribute

All metadata documents following the EML schema must be given a globally unique identifier that allows identification and citation of the dataset. This identifier should be placed in the required @**packageId** attribute, and if the dataset will be published, it is recommended that the @**packageId** attribute contain the same identifier as will be used by the repository. The content and structure of these identifiers may follow practices in place at a data manager's local level, the data repository's specifications, or a combination of the two.

> EDI context note
>
> At the EDI repository, the @**packageId** is entered into the repository software in a format that is standardized to three parts: *scope*, *accession number*, and *revision*. The scope should be "edi" unless another scope is justified by prior arrangement, such as in Example 1.

System attribute

The @**system** attribute is required to identify the data management or repository system that an EML document belongs to. This attribute provides the context needed to interpret other attributes of the EML, particularly @**packageId**. The value of this attribute should be recognizable to the EML preparer's local data management system, or to the repository the EML will be published in (see example B.5).

> EDI context note
>
> When publishing to the EDI repository, the @**system** attribute will be replaced with "https://pasta.edirepository.org."

Example B.5: A root EML element's starting tag, including required attributes **@schemaLocation**, **@packageId**, and **@system**. The **@system** attribute is set to "https://pasta.edirepository.org", indicating that this dataset is, or will be, published in the EDI repository and the **@packageId** attribute uses the EDI identifier format. Note that the three other namespace attributes (`xmlns:eml`, `xmlns:xsi`, `xmlns:stmml`) are not strictly required.

```
<?xml version="1.0" encoding="UTF-8"?>
<eml:eml xmlns:eml="https://ecoinformatics.org/dataset-2.2.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:stmml="http://www.xml-cml.org/schema/stmml"
   xsi:schemaLocation="https://ecoinformatics.org/eml-2.2.0
      https://ecoinformatics.org/eml-2.2.0/eml.xsd"
   packageId="frs.21.3"
   system="https://pasta.edirepository.org">
```

Top Level Elements

There are a number of potential top-level elements that can be nested directly below the EML root (<**eml:eml**>). Only one, a <**dataset**> element, is required for data packages, but several others are commonly used. We briefly describe the most common and useful top-level elements below, and then mention others that are more suitable for use within <**dataset**>. Many of these elements receive greater attention in later chapters.

The dataset element (<dataset>)

The <**dataset**> element is an EML document's flexible container for the vast majority of metadata describing the data file(s) being shared or published. Under <**dataset**>, many EML elements are available to describe the dataset. Some of these elements are required and some are optional, and some (such as people and organizations) are "repeatable" elements that may be nested at multiple levels and locations within a <**dataset**>. All must follow the order enforced by the EML schema. Refer to Chapter 2 for a list of the highest-priority metadata elements needed to meet FAIR data principles, in the order they should be included as child elements of <**dataset**>. Chapters 3-12 of this document are devoted to recommended placement, formatting and content of these sub-elements of <**dataset**>. Though not all are required, we highly recommend including them to facilitate re-use of the data resource when it is shared or published.

EDI context note

When publishing to the EDI repository, the PASTA+ system will add an <**alternateIdentifier**> element to the EML document that includes the unique Digital Object Identifier (DOI) generated for the published data package.

The descriptive metadata elements within <**dataset**> should be followed by one or more data entity elements that describe the actual data files being shared or published. There are several possible types of data entity elements that may be chosen depending on the file or data. The most commonly used are:

- <**dataTable**> describes a tabular data file, such as an delimited text or spreadsheet file
- <**otherEntity**> describes a file that doesn't fit into the data entity categories above
- <**spatialRaster**> describes a georeferenced raster data file, such as a geotiff
- <**spatialVector**> describes a data file describing georeferenced geometries, such as a shapefile

Both of the spatial data entity types are infrequently used and are primarily described in the "Data Package Design for Special Cases" companion to this document. The most infrequently used elements include <**storedProcedure**>, which describes a measurement or observation protocol, and <**view>,** which describes a query of a relational database or other structured data resource. We include no best-practices for these data entity types in this document.

Additional metadata (<additionalMetadata>)

The <**additionalMetadata**> element is a flexible field for including any other relevant metadata that pertains to the resource being described by EML. Its content must be valid XML. Though there is significant flexibility in how to create and use <**additionalMetadata**> elements, there are also some common use cases that require particular child elements. Several use cases and other considerations for using this element are described in Chapter 12.

Access elements (<access>)

Note that this element is deprecated in EML 2.2 (link)

An <**access**> element contains a list of rules defining access permissions for an EML document's metadata and any data files (or entities) that the metadata describes. In general, <**access**> trees precede other top-level EML elements, and any access rules must be specific to the system where the dataset is stored. Usually, that system is a research data repository.

This element is now deprecated, but is still in use by data repositories (including EDI) that are backward compatible with EML 2.1.0. Note that if <**access**> is omitted, the repository

may presume that only the dataset submitter will be allowed access. The <**access**> element is described more fully in Chapter 6.

Dataset annotations (<annotations>)

Annotations are a more recent addition to the EML schema and are used to describe the purpose and content of a dataset using precise semantics. An <**annotations**> element contains a list of child <**annotation**> elements, and can be included within many EML elements, at multiple levels within an EML document (including the root). Annotations are described in detail in Chapter 7.

Other top-level elements

The <**citation**>, <**software**>, and <**protocol**> elements may also be placed directly below <**eml:eml**>. When used at this level, the EML document is primarily being used to describe a bibliographic source (article, book, etc.), software package, or published scientific protocol instead of a dataset. EML metadata documents are only rarely used in this way and other, more widely accepted standards exist. These use cases for EML are therefore outside the scope of this document..

EML Complex Types and the module system

As described above, a range of useful XML `complexTypes` have been defined in the EML schema (they are referred to as Complex Types in the schema documentation). We have already briefly discussed several elements that are instances of these types, and each instance of a type must contain specific child elements and contents to be valid. Some commonly recurring and generally useful EML data types are listed below, but there are many more.

- **TextType** (EML schema definition) is a data type used to convey formatted or unformatted descriptive text. Formatted TextType elements use <**section**>, <**para**>, and <**markdown**> child elements to define sections, paragraphs, titles, and other formatting that makes long-form text more human-readable. Unformatted TextType elements do not use these child elements. Note that using the <**markdown**> child element allows the use of markdown formatting in descriptive metadata fields, but display of this formatting depends on support by repositories or other platforms. The <**abstract**> (Chapter 3), <**intellectualRights**> (Chapter 7), <**methodStep**> (Chapter 9), and many <**description**> elements found in EML are TextType elements.
- **CitationType** (EML schema definition) is used to assemble bibliographic information for citing published works like journal articles or books. The <**citation**> elements described in Chapter 8 (and above) are CitationTypes that may be used in several locations in

EML See Chapter 8 for usage information and the schema documentation (Section 5.1.4, eml-literature module) for additional details.

- **ResponsibleParty** (EML schema definition) is a data type defining a person, organization, or role associated with the EML dataset. They contain many possible child elements depending on the party being described. The <**creator**>, <**contact**>, <**metadataProvider**> and other elements are ResponsibleParty elements and are described in section Chapter 4.
- **ResearchProjectType** (EML schema definition) is a data type used to assemble information about the project under which the dataset was created. They contain child elements to describe the project, such as <**title**>, <**abstract**>, <**personnel**>, and <**award**> information. See Chapter 5 for more.

The EML schema and its associated Complex Types are organized into a system of modules. Complex Types and associated EML elements that serve similar purposes or share similar functions are grouped together into named modules. For instance, data types and elements having to do with citing sources (journal articles, books, etc.) are grouped in the `eml-literature` module. These modules are fairly regularly referred to within the EML schema documentation.

A very simple EML example

To create a valid EML document, there are a few required elements and attributes. In example B.6 we present a very simple EML document that contains all required elements and will successfully validate against the EML schema. However, the metadata this document contains is not very descriptive and wouldn't be sufficient to re-use a dataset if it were published this way. For this reason, the main chapters of this document elaborate on how to create rich, useful metadata and place it into an EML document. Beginning in Chapter 3, the highest priority EML elements to include with any published dataset are described, along with how to populate and structure them to make published datasets more compliant with FAIR principles.

Example B.6: A minimal example of a valid EML document, including a declaration, the required EML root and dataset elements, and any required attributes and child elements for each. Inclusion of a data entity (<**otherEntity**> in this case), is optional, but shown for clarity.

```
<?xml version="1.0" encoding="UTF-8"?>
<eml:eml xsi:schemaLocation="https://eml.ecoinformatics.org/eml-2.2.0
 ↪  https://eml.ecoinformatics.org/eml-2.2.0/eml.xsd" packageId="frs.1002.3"
 ↪  system="frs">
  <dataset>
    <title>Pirate attacks in the South Seas, 2000-2014</title>
    <creator>
```

```
      <individualName>
        <givenName>Grace</givenName>
        <surName>O'Malley</surName>
      </individualName>
    </creator>
    <contact>
      <organizationName>Fictitious Research Site</organizationName>
    </contact>
    <otherEntity>
      <entityName>An example data entity</entityName>
      <entityType>Shapefile</entityType>
    </otherEntity>
  </dataset>
</eml:eml>
```

Resources

About the EML schema

- The EML Schema XSD documents: https://github.com/NCEAS/eml/tree/main/xsd
- The EML development project's schema documentation: https://eml.ecoinformatics.org/
- Summary of EML validation rules in EML schema documentation, section 6

Tools for validating XML against the EML schema

- EML Parser - An online EML parser/validator maintained by EML developers
- emlvp - EML Validator/Parser is a simple python package written by the EDI repository that can validate EML documents
- EML - An R package that can help build and then validate EML documents
- A VScode editor extension maintained by redhat.

XPaths referenced in this chapter

Root EML element: **/eml:eml**

Schema location attribute: **/eml:eml/@schemaLocation**

Package identifier attribute: **/eml:eml/@packageId**

Dataset element: **/eml:eml/dataset**

Additional metadata: **/eml:eml/additionalMetadata**

Top-level annotations: **/eml:eml/annotations**

Top-level access element: **/eml:eml/access**

Dataset annotations: **/eml:eml/dataset/annotations**

Appendix C. Known issues and gotchas in EML

Just bullet points for now

- Custom units & varying capitalization of an attribute
- Validator problems
- Id and references - RP id cannot be re-used.
- Referring to citations multiple times
- EDI overwrites entity @**id** attribute

14 References

Bahim, Christophe, Carlos Casorrán-Amilburu, Makx Dekkers, et al. 2020. "The FAIR Data Maturity Model: An Approach to Harmonise FAIR Assessments." *Data Science Journal* 19 (1). https://doi.org/10.5334/dsj-2020-041.

ESIP, Data Readiness Cluster. 2022. "Checklist to Examine AI-Readiness for Open Environmental Datasets." Online resource. In *Figshare*. https://doi.org/10.6084/m9.figshare.19983722.v1.

Jones, Matthew, Peter Slaughter, and Ted Habermann. 2019. *Quantifying FAIR: Metadata Improvement and Guidance in the DataONE Repository Network*. https://doi.org/10.5063/F14T6GP0.

Wilkinson, Mark D., Michel Dumontier, IJsbrand Jan Aalbersberg, et al. 2016. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." *Scientific Data* 3 (1): 160018. https://doi.org/10.1038/sdata.2016.18.