Report to the IMC
EML Data Package Checks and the PASTA Quality Engine
July 2012

IMC EML Metrics and Congruency Checker Working Group
Sven Bohm (KBS), Emery Boose (HFR), Duane Costa (LNO), Jason Downing (BNZ), M. Gastil-Buhl (MCR), Corinna Gries (NTL), Margaret O'Brien (SBC, chair), Mark Servilla (LNO)

Introduction

To date, the LTER Network has not adopted a system of standards for data packages, although components of the LTER Network Information System (NIS) will require high quality in the EML data packages it ingests and processes with automated workflows. To be ingested by PASTA, a data package must include complete metadata, access to data, and agreement between data and metadata. Tools such as the EML Congruency Checker are being developed for reporting on the quality of data packages submitted by individual LTER research sites. These tools are based on open source software which is managed as part of the EML suite of schemas and libraries, and although current development is led by LTER, the eventual scope of use is expected to be very broad.

In 2010, the IMC EML Metrics and Congruency Checker (ECC) working group, the Data Manager Tiger Team, and NIS Developers began compiling a set of checks to address these needs. In 2011, five were implemented in an early version of the Quality Engine and tested against thousands of LTER data packages. To finalize the comprehensive set of checks for PASTA, a workshop was held in Spring 2012. The workshop was attended by a small community of experts from within and outside of LTER with extensive knowledge of EML and the issues exhibited by existing data packages. This document is the report of those efforts.

The challenges to the 2012 workshop participants were to
a. determine specifically what quality checks would be required to meet the criteria of the LTER community for high quality data packages;
b. consider the behavior of the Data Manager Library (core code for the Quality Engine);
c. consider Best Practice recommendations and EML construction currently in use; and
d. prioritize checks for the greatest return on investment.

The workshop also outlined a design for a web interface for data package evaluation software (which will be presented at a later date), the requirements for code behavior during evaluation vs. during PASTA-harvest, and a process for future check implementation. For most aspects of check-definition, current community practice was surveyed by querying the LTER metadata catalog (Metacat).

Products (quoted from 2012 workshop proposal):
1. Complete list of checks; organized by types (data, metadata, congruency), status response (valid, info, warn, error) and the criteria for each, Checks are prioritized based on a perceived return on investment for the community, with priorities justified.
2. Draft of a document describing the checks and Quality Engine behavior for comment by stakeholders and NISAC, and which can be further developed to inform those engaged in manual evaluation of datasets (requested by IMExec).

This report describes Workshop Product #2, the description, summary and justification of checks and Quality Engine behavior, and parts of it will form the basis of that draft. Workshop Product #1, the complete list of checks with all characteristics, is in the accompanying spreadsheet.

List of PASTA checks
As of May 2012 a total of 72 checks have been logged. Of these, 51 were planned to be included in PASTA, and have been fully described. Twenty (20) were implemented by May 2012, and 31 will be implemented in later releases of PASTA. The remaining 21 checks (72 minus 51) were designated as "deprecated" or "postponed". These may be reconsidered in the future, and additional new checks may be logged at the request of the community. All the checks and their definitions can be found in the accompanying Excel workbook. This document provides a summary and discussion.

Categorization
The checks were categorized according to several criteria (scope, priority, type, use, response status), and those terms and their definitions are summarized in Table 1. The checks themselves are summarized by response status in Table 2.

1. Scope: the Quality Engine is based on EML-community software called the Data Manager Library. This software is being extended by LTER as part of PASTA's Quality Engine, but it is expected that this new capability will be used by other communities with systems based on EML. Not all communities are expected to employ the same criteria for data package acceptance, so each check can be categorized with a 'scope', to indicate the community applying it. General checks that would apply to any data package (e.g., presence of working URLs) were given the scope "*knb*", and checks that are specific to the LTER community are labeled "*lter*" (e.g., features recommended by our EML Best Practices). Other values for scope may be added by interested communities.

2. Priority: each check was given one of 3 priority levels (*high*, *medium*, *low*) depending on its importance to PASTA and to the IMC. Priority levels have helped guide the implementation, but are not the only factor used to determine the schedule.

3. Type: reflects what part of a data package a check pertains to. "*Metadata*" checks are concerned purely with metadata presence or content, e.g., a check that looks for the presence of an element, such as "<methods>". "*Data*" checks are concerned only with the data entity, e.g., a check that simply returns a row of data or examines a data record for possible delimiters. "*Congruency*" checks examine the agreement between metadata and data, for example, to compare the number of attributes listed in metadata to the number of columns in a data table.

4. Use: the IMC specifically asked that checks not cause undue burden for data package submitters, and so each check was justified by stating how the EML feature was important to the NIS. "*Discovery*" use applies to those elements used by search tools, during human evaluation, or for building data citations. "*Workflow*" was applied to data package features essential to workflow software. "*PASTA*" and "*DAS*" refer to data package features specifically required by those two NIS software components. "*Good practice*" was gleaned from EML Best Practices documents (LTER, 2010), the EML specification, and requests from the LTER Executive Board or NSF. In some cases, a check may belong to more than one category; for example, a "*good practice*" may be defined as such to promote "*discovery*".

5. Response status: each check returns a status response, and some responses affect the insertion of the package into PASTA. *'Info*" checks do not affect the acceptance of the data package into PASTA in any way; an "*info*" check is for informational purposes only; for example the check that displays content of a URL has response status "*info*". A check that affects the acceptance of the dataset will return a response of "*valid*", "*warn*" or "*error*". Any "*error*" response during the checking process means that the entire data package will be rejected by PASTA. "*Warn*" means that the criteria of the check were not met and that there may be some problem needing attention, but that the data package is still acceptable to PASTA. "*Valid*" means that all criteria of the check were met. So there are a total of four possible response

statuses: either "*info*", or one of "*valid*" | "*warn*" | "*error*". See below for further discussion of data package evaluation and code behavior.

6. Implementation: 'yes' indicates that the check has been implemented as of May 2012 (V1.0 PASTA).

Table 1. Definition of classification terms for checks in Table 2, discussed in text, and in the accompanying spreadsheet.

Scope	KNB	Check applies to general features of a data package
	LTER	Check is used by LTER community
Priority	1	High
	2	Medium
	3	Low
Type	Metadata	Check applies only to EML metadata
	Data	Check applies only to the data entity (table) itself
	Congruency	Check applies to the agreement of the data with its metadata
Response status	valid	The data package meets the conditions of the check
	warn	Highlights bad practice or possible problem needing attention, but will not stop processing
	error	An error response will prevent loading of data into PASTA
	info	Check is descriptive, or for informational purposes only
Use	Workflow	Automated processing of data entities requires this feature
	PASTA	PASTA will require this feature
	DAS	The LTER Data Access System uses this feature
	Discovery	This feature is used by tools which 'discover' or return result-sets from data searches
	Good practice	This feature is recommended by an authority
Implemented	Yes	Implemented in Quality Engine code V1.0 (as of May 2012)
	[blank]	Not yet implemented

Table 2. Summary of data package checks by response status. A) checks producing an “*error*” response, B) “warn”, and C) “*info*”. Those marked as “implemented” as of May 2012 comprise V1.0. See the Excel workbook for the full description of each check.

A. Checks producing an "*error*". Failure to meet a check will stop insertion into PASTA.

Priority	Check name	Implemented
1	EML is version 2.1.0 or beyond	Yes
1	Document is schema-valid EML	Yes
1	Document is EML parser-valid	Yes
1	All entity-level data URLs are live	Yes
1	The packageId pattern matches "scope.identifier.revision"	Yes
1	There are no duplicate entity names (entityNames are unique)	Yes
1	An entity-level URL which is not set to “information” returns data	Yes
1	Data table does not have more fields than metadata attributes	Yes
1	Data table does not have fewer fields than metadata attributes	Yes
1	Database table can be created from EML metadata	Yes
2	enumeratedDomain codes are unique	
2	Field delimiter in metadata is a single character	Yes
3	Document is schema-valid after dereferencing	Yes

B. Checks producing a "warn", which will not stop processing. At some time in the future, some of these checks may be elevated to “error” and data packages could be required to meet them.

Priority	Check name	Implemented
1	Data can be loaded into the database	Yes
1	Length of entityName is not excessive (less than 100 char)	Yes
1	A methods element is present	Yes
1	Record delimiter is present in metadata	Yes
1	Data examined and possible record delimiters returned	Yes
1	Date format in metadata is a preferred format [LIST TBD] and data matches	
1	Date format in metadata is a non-preferred format [LIST TBD] and data matches	
1	Numeric fields in data do not have string content	
2	Dates are within stated bounds	
2	Attribute Names are unique	
2	An entity description is present	
2	Dataset title length is at 5 least words	
2	One of dataTable, view, spatialRaster or spatialVector is present	
2	pubDate element is present	

2	Integer fields do not include floats	
2	objectName is valid file name	
2	Dataset abstract element is a minimum of 20 words	
2	Coverage element is present	
2	If attribute had enumeratedDomain data values are within domain	
2	Numeric values are within stated bounds	
2	At least one controlled vocabulary term (preferred or non-preferred) is in keywords	
3	Number of records in metadata matches number of rows loaded	Yes
3	If constraint specified notNull, then data has no null fields	
3	Controlled vocabulary preferred term is present	

C. Checks are for "information" purposes.

Priority	Check name	Implemented
1	Display downloaded data	Yes
1	Display first insert row	Yes
2	numFooterLines element is present	
2	numHeaderLines element is present	
2	Header row is displayed alongside attribute names	
2	publisher element is present	
3	Duplicate data rows are displayed	
3	attributeName is not in a reserved word [LIST TBD]	
3	temporalCoverage element is present	
3	geographicCoverage element is present	
3	taxonomicCoverage element is present	
3	temporalCoverage endDate is not in the future	
3	intellectualRights element is present	
3	pubDate is not in future	

Data Package Quality Report

An XML schema was designed to contain the output of the Quality Engine, and an instance document is called a Data Package Quality Report. Using XML to house the report means that output can be transformed for a variety of purposes, e.g., an individual report can be transformed into HTML for web presentation during evaluation of a single data package, or results from a group of reports can be aggregated and statistics computed.

The expectation is that different communities will want to control the behavior of some checks. So every check can be configured in an XML template (an instance document), and the XML

template controls behavior of the checker code. An example of the report template is shown in Figure 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<qr:qualityReport
  xmlns="eml://ecoinformatics.org/qualityReport"
  xmlns:qr="eml://ecoinformatics.org/qualityReport"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="eml://ecoinformatics.org/qualityReport http://svn.lternet.edu/svn/NIS/documents/schemas/quali
  <creationDate>2011-12-01T12:00:00</creationDate>
  <packageId></packageId>
  <includeSystem>knb</includeSystem>
  <includeSystem>lter</includeSystem>
  <datasetReport>
    <qualityCheck qualityType="metadata" system="lter" statusType="error" >
      <identifier>emlVersion</identifier>
      <name>EML version 2.1.0 or beyond</name>
      <description>Check the EML document declaration for version 2.1.0 or higher</description>
      <expected>eml://ecoinformatics.org/eml-2.1.0 or eml://ecoinformatics.org/eml-2.1.1</expected>
      <found></found>
      <status>notChecked</status>
      <explanation>Validity of this quality report is dependent on this check being valid.</explanation>
      <suggestion>Use an approved namespace.</suggestion>
      <reference></reference>
    </qualityCheck>
    <qualityCheck qualityType="metadata" system="knb" statusType="error" >
      <identifier>schemaValid</identifier>
      <name>Document is schema-valid EML</name>
      <description>Check document schema validity</description>
      <expected>schema-valid</expected>
```

Figure 1. Example XML quality report template.

Quality Engine Behavior

A NIS Data Package Manager Web service component called the "Quality Engine" codifies checks during data package analysis. The Quality Engine is based on the open source EML Data Manager Library, and so the EML community beyond LTER is involved in its design and behavior. The Quality Engine uses the XML template to apply each check, and can be run in one of two modes: harvest (upload) and evaluation. Each data package harvested (uploaded) by PASTA will have its quality report associated and available when data are downloaded. A full introduction to the Quality Engine will be available at the 2012 All Scientists Meeting.

In addition to controlling data contributions to PASTA, when run in evaluation-mode, the Quality Engine code also forms the basis of a tool for evaluating data packages prior to submission. Typically, software evaluating XML stops at the first error, and repeated submissions are required until all errors have been exposed. The workshop participants requested that, when run in evaluation-mode, the Quality Engine should continue after an error, thus exposing as many errors as possible in one run. This will save package submitters considerable time. Of course, some errors will prevent future processing, e.g., if a data-URL does not return a data entity, that entity's delimiters cannot be examined. When run in harvest-mode, the Quality Engine will halt on the first error.

We expect that the community will wish to add checks to the system as the NIS matures. Also, existing checks may require modification in the future; for example, a check with a response status of "*warn*" in 2012 may require reclassification to return an "*error*", or conversely, a check's response status may be relaxed. It is imperative that changes to the check configuration be implemented with adequate notice to sites and without causing undue burden. We propose that a timeline be adopted in which the checks are reviewed periodically by an IMC sub-committee and changes or modifications announced. After a comment period, new checks would be implemented after 6 months. This schedule would mean that changes to the checks might be

expected to occur approximately annually. This system will be complemented by a centralized reporting mechanism in which all available data packages are examined and their compliance with current checks summarized. It has been suggested that the LTER Network Office conduct such reporting quarterly, and report results individually to each site. Aggregated results could be made available for other uses, such as reports requested by site PIs, the Executive Board or NSF. This proposed system will be presented to the IMC for their discussion, and approval at their annual meeting in 2012.

The Quality Engine is not yet complete and has certain limitations. Most notably:

- Of the more than fifty planned quality checks, twenty have been implemented and are being actively applied by the PASTA prototype system
- Of the six types of EML data entities, the Quality Engine can process “dataTable”, “spatialRaster”, “spatialVector”, and “otherEntity” entities. However, programming logic has not yet been developed and tested to process “storedProcedure” and “view” entities.
- When uploading data tables to a relational database, the Quality Engine (due to limitations in the underlying Data Manager Library) is unable to process datetime fields reliably. To compensate for this, the specific quality check that ensures that data tables can be successfully loaded into a database is currently set at the ‘warn’ level. We might prefer to have it set at the ‘error’ level were we to have full confidence in the Quality Engine’s ability to process datetime fields.

Terms and references used in this document

EML, Ecological Metadata Language: Specification used by the LTER network for data exchange. http://knb.ecoinformatics.org/software/eml

Metadata: data about data. In the LTER Network, this is represented by an EML document.

EML Best practices for LTER Sites: Recommendations for EML metadata content in LTER documents. Most recently updated in 2010.
http://im.lternet.edu/sites/im.lternet.edu/files/emlbestpractices-2.0-FINAL-20110801_0.pdf

Data entity: the data object described by EML. Usually a data file.

Data package: a data entity plus its EML metadata intended for submission

Congruency (also, Congruence): agreement between a data entity and its metadata

NIS, Network Information System: Informatics software framework intended for network use, http://nis.lternet.edu

PASTA, Provenance Aware Synthesis Tracking Architecture: NIS components intended to house and deliver data packages for synthesis, http://nis.lternet.edu/pasta

Tiger Team: a group of LTER members who specify requirements for, comment on, and test NIS software. https://nis.lternet.edu/NIS/?q=node/5

Quality Engine: PASTA component that assesses compliance of metadata and data submitted to the NIS

Data Manager Library (DML): software that reads a data table (type of data entity) described by EML-metadata and loads the table into a relational database. Part of the EML family of tools, and used by the PASTA Quality Engine

Quality report: XML output from the PASTA Quality Engine containing responses from the checks applied to an EML data package.

Resource map: A data structure describing a set of resources associated with a data package, including its EML metadata, data entities, and the quality report document generated from the data package at the time of its insertion into PASTA

ECC, EML Congruency Checker: code which interfaces with the PASTA Quality Engine to report on EML data package congruency (originally), and general metadata quality.