

# Open Geospatial Consortium

Submission Date: 2022-04-05

Approval Date: <yyyy-mm-dd>

Publication Date: <yyyy-mm-dd>

External identifier of this OGC® document: <http://www.opengis.net/doc/{doc-type}/{standard}/{m.n}>

Internal reference number of this OGC® document: 19-002

Version: 0.5

Category: OGC® Implementation

Editor: Mark Hedley, Jonathan Yu

## OGC Encoding Linked Data Graphs in NetCDF Files

### Copyright notice

Copyright © 2022 Open Geospatial Consortium

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>

### Warning

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OGC® Standard

Document subtype:

Document stage: Draft

Document language: English

## License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

# Table of Contents

1. Scope .....	8
2. Conformance .....	9
3. References .....	10
4. Terms and Definitions .....	11
4.1. <b>RDF</b> .....	11
4.2. <b>RDF statement (or triple)</b> .....	11
4.3. <b>RDF graph</b> .....	11
4.4. <b>URI</b> .....	11
4.5. <b>Literal</b> .....	11
4.6. <b>CDL</b> .....	11
5. Conventions .....	13
5.1. Identifiers .....	13
5.2. UML model .....	13
6. Clauses not Containing Normative Material .....	14
6.1. Rationale and Design principles .....	14
6.1.1. Design principles .....	15
6.2. Resource Description Framework .....	15
6.3. NetCDF File .....	16
6.4. RDF Vocabularies .....	17
6.5. Metadata RDF Graph .....	19
6.5.1. Uniform Resource Identifiers within netCDF files .....	20
6.5.2. Identity .....	20
6.5.3. Containment .....	22
6.5.4. Prefix Definition .....	22
6.5.5. Alias Definition .....	26
6.5.6. Attribute Names .....	29
6.5.7. Variable-to-Variable References .....	30
6.5.8. Attribute Values .....	32
6.5.9. Reference Exceptions .....	32
6.6. NetCDF Dimensions .....	33
6.6.1. Broadcasting .....	34
6.6.2. Shape and Reshape .....	36
6.6.3. Variable References and Dimensions .....	36
6.6.4. NetCDF Coordinate Variables .....	41
6.6.5. Worked Example .....	42
6.6.6. Worked Example - Climate and Forecasting Conventions .....	44
6.7. NetCDF Linked Data on Schema.org .....	47
6.8. Implementation Adaptation .....	48

7. Clause containing normative material .....	50
7.1. Requirement Class A: URIs .....	50
7.1.1. Requirement A-1 .....	50
7.1.2. Requirement A-2 .....	51
7.1.3. Requirement A-3 .....	52
7.1.4. Requirement A-4 .....	52
7.1.5. Requirement A-5 .....	52
7.2. Requirement Class B: Prefixing .....	53
7.2.1. Requirement B-1 .....	53
7.2.2. Requirement B-2 .....	54
7.2.3. Requirement B-3 .....	54
7.2.4. Requirement B-4 .....	55
7.2.5. Requirement B-5 .....	55
7.2.6. Requirement B-6 .....	55
7.2.7. Requirement B-7 .....	56
7.2.8. Requirement B-8 .....	56
7.3. Requirement Class C: Aliasing .....	56
7.3.1. Requirement C-1 .....	57
7.3.2. Requirement C-2 .....	57
7.3.3. Requirement C-3 .....	57
7.4. Requirement Class D: Attribute names .....	58
7.4.1. Requirement D-1 .....	58
7.4.2. Requirement D-2 .....	58
7.4.3. Requirement D-3 .....	58
7.4.4. Requirement D-4 .....	58
7.5. Requirement Class E: Variable-to-variable referencing .....	59
7.5.1. Requirement E-1 .....	59
7.5.2. Requirement E-2 .....	59
7.5.3. Requirement E-3 .....	60
7.5.4. Requirement E-4 .....	61
7.5.5. Requirement E-5 .....	63
7.5.6. Requirement E-6 .....	63
7.5.7. Requirement E-7 .....	63
7.5.8. Requirement E-8 .....	64
7.6. Requirement Class F: NetCDF Coordinate Variables .....	64
7.6.1. Requirement F-1 .....	64
7.6.2. Requirement F-2 .....	64
8. Media Types for any data encoding(s) .....	66
Annex A: Conformance Class Abstract Test Suite (Normative) .....	67
A.1. Conformance Class A: URIs .....	67
A.1.1. Abstract Test for Requirements A-1, A-2, A-3, A-4, A-5 .....	67

A.2. Conformance Class B: Prefixing .....	68
A.2.1. Abstract Test for Requirement B-1, B-2, B-3, B-4, B-5, B-6, B-7, B-8 .....	68
A.3. Conformance Class C: Aliasing .....	69
A.3.1. Abstract Test for Requirement C-1, C-2, C-3 .....	70
A.4. Conformance Class D: Attribute Names .....	71
A.4.1. Abstract Test for Requirement D-1, D-2, D-3, D-4 .....	71
A.5. Conformance Class E: Variable-to-variable Referencing; Class F: Coordinate Variables .....	73
A.5.1. Abstract Test for Requirement E-1, E-2, E-3, E-4, E-5, E-6, E-7, E-8, F-1, F-2 .....	73
Annex B: Relationship to other OGC Standards (Informative) .....	78
Annex C: Revision History .....	79
Annex D: Bibliography .....	80

## **i. Abstract**

netCDF is a format for encoding array-oriented scientific data, particularly in many Earth & Space Science domains, and is an Open Geospatial Consortium (OGC) standard. It is common practice to aid interoperability of netCDF files in the Earth sciences by using the Climate and Forecasting (CF) convention and also the Attribute Convention for Data Discovery (ACDD). Several communities are defining additional netCDF conventions for describing the semantics relating to their different domains. As the concurrent use of multiple, and possibly clashing or conflicting, netCDF conventions spreads, the challenge of finding a common mechanism to validate and interpret metadata embedded inside netCDF files grows.

Linked Data (LD) is a World Wide Web Consortium best practice for encoding and publishing metadata to expose and connect data both within individual datasets and across multiple datasets. LD also allows dataset descriptions to become more useful through the ability to run semantically-enabled queries across datasets. Publishing netCDF file descriptions using LD methods presents the user community with an opportunity to address this exposure and connection and, further, to enhance data findability and re-use.

The netCDF Linked Data standard (netCDF-LD) provides the encoding standard for encoding linked data semantics into netCDF files and interpreting netCDF files as RDF graphs.

## **ii. Keywords**

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, netCDF, Linked Data

## **iii. Preface**

This encoding standard standardises a W3 RDF encoding for a graph derived from the metadata payload of a netCDF file. This provides a mechanism to summarise and index individual netCDF files and to aggregate summaries across file sets and structured collections of identified files.

In providing these encoding mechanisms and rules, this standard provides a framework for publishers to define namespace qualified attributes for new metadata standards, enabling extended interoperability for using multiple complementary metadata scheme within a file and for existing standards to continue with their standardised terms. The standard also provides opportunity for publishers of metadata standards for netCDF files to publish standards specifications as machine parsable content for use within the parsing process.

The mechanisms standardised are deliberately generic, with detailed implementation to netCDF provided. This gives the opportunity for future work, beyond the scope of this standard, to reuse aspects of the approach for content metadata extraction and aggregation for other file formats; for example HDF5 and GeoTIFF may be suitable targets for future standardisation work.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any

relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

#### **iv. Submitting organizations**

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- UK Met Office
- CSIRO
- Marine Institute, Ireland

#### **v. Submitters**

All questions regarding this submission should be directed to the editor or the submitters:

<b>Name</b>	<b>Affiliation</b>
Mark Hedley	UK Met Office
Jonathan Yu	CSIRO
Adam Leadbetter	Marine Institute, Ireland

# Chapter 1. Scope

NetCDF-LD is an approach for constructing Linked Data descriptions using the metadata and structures found in netCDF files. Linked Data is a set of World Wide Web Consortium (W3C) best practices to allow structured data to be published so that it can be interlinked on the web and support semantic queries. Linked Data uses the W3C Resource Description Framework (RDF) to express the information and relationships.

NetCDF-LD enhances netCDF metadata, enabling information found in netCDF files to be linked with published conventions and controlled vocabularies used to express the content.

The netCDF Linked Data standard: netCDF-LD provides the encoding standard for encoding linked data semantics into netCDF files and interpreting netCDF files as RDF graphs.

NetCDF files may use the HDF5 based netCDF-4 encoding, which is limited to the netCDF data model, or the legacy netCDF-3 encoding.



## Chapter 2. Conformance

This standard defines RDF Metadata graphs interpreted from netCDF metadata encoded within netCDF files.

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site.

All requirements-classes and conformance-classes described in this document are owned by the standard(s) identified.

# Chapter 3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

Hayes, P. J., Patel-Schneider, P. F., RDF 1.1 Semantics. W3C Recommendation, 25 February 2014. URL: <http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/>. The latest edition is available at <http://www.w3.org/TR/rdf11-mt/>

Prud'hommeaux, E., Carothers, G., RDF 1.1 Turtle: Terse RDF Triple Language. W3C Recommendation, 25 February 2014. URL: <http://www.w3.org/TR/2014/REC-turtle-20140225/>. The latest edition is available at <http://www.w3.org/TR/turtle/>

Harris, S., Seaborne, A., SPARQL 1.1 Query Language, W3C Recommendation, 21 March 2013, URL: <http://www.w3.org/TR/sparql11-query/>, (2013).

Binary Array Linked Data ontology (BALD), <https://opengis.net/def/binary-array-ld>

OGC: NetCDF Core Encoding Standard version 1.0, [http://portal.opengeospatial.org/files/?artifact\\_id=43732](http://portal.opengeospatial.org/files/?artifact_id=43732), (2011).

ESIP: Attribute Convention for Data Discovery (ACDD) Version 1.3, [http://wiki.esipfed.org/index.php/Attribute\\_Convention\\_for\\_Data\\_Discovery](http://wiki.esipfed.org/index.php/Attribute_Convention_for_Data_Discovery), (2015).

W3C: RDF 1.1 Turtle, <https://www.w3.org/TR/turtle/>, (2014).

W3C: RDF 1.1 XML Syntax, <https://www.w3.org/TR/rdf-syntax-grammar/>, (2014).

W3C: Notation3 (N3): A readable RDF syntax, <https://www.w3.org/TeamSubmission/n3/>, (2011)

W3C: SKOS Simple Knowledge Organization System Reference, (2009).

W3C: Resource Description Framework (RDF): Concepts and Abstract Syntax, (2004).

Python Numpy library: Broadcasting, <https://numpy.org/devdocs/user/theory.broadcasting.html>.

# Chapter 4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

For the purposes of this document, the following additional terms and definitions apply.

## 4.1. RDF

The Resource Description Framework (RDF) is a standard model for data exchange on the World Wide Web. RDF is specifically designed to allow the exchange of data, even where underlying schemas are different, through the use of a normalised data model of subject-predicate-object triples and using Uniform Resource Identifiers to define terms within the triples.

## 4.2. RDF statement (or triple)

A triple is the atomic data entity in RDF. It is a set of three entities used to make a statement about the subject of the triple. The triple takes the form subject-predicate-object (e.g. "The sea - has colour - blue"). In a Linked Data system, as many of the portions of the expression as possible are defined by URIs. RDF statements are often called "triples".

## 4.3. RDF graph

An RDF graph is a combination of a number of triples or RDF statements. An RDF graph may be thought of in terms of the concepts from the branch of mathematics known as graph theory. In an RDF graph, the subject and object portions of the triple form the nodes (or vertices) of the graph, while the predicate is used to label the edges (or connectors between the nodes) of the graph.

## 4.4. URI

A Uniform Resource Identifier (URI) is a string of characters which identifies a logical or physical resource. Uniformity of URIs comes through a set of predefined syntactic rules. The most common form of URI is the Uniform Resource Locator (URL), which is frequently referred to in common parlance as a 'web address.'

## 4.5. Literal

Within RDF, literals are the lexical representations of values such as numbers or dates, and literals are therefore not defined by URIs.

## 4.6. CDL

The Common Data Language (CDL) is a human readable notation for describing netCDF objects and

data.

# Chapter 5. Conventions

This sections provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

## 5.1. Identifiers

The normative provisions in this specification are denoted by the URI

<http://www.opengis.net/spec/netcdf-ld/1.0>

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

## 5.2. UML model

UML diagrams are included in this standard to illustrate the conceptual model that underpins netCDF-LD. The UML model is not normative. The UML profile used is specified in ISO 19103:2015.

Resources are modelled as UML interfaces.

# Chapter 6. Clauses not Containing Normative Material

## 6.1. Rationale and Design principles

netCDF is used widely in the environmental sciences for data capture, analysis and archival. netCDF as a format is used for encoding array-oriented earth- and space-science data. netCDF is an open Geospatial Consortium (OGC) standard. As netCDF can be, and is, used to store and disseminate a wide-range of multi-dimensional data types it is a natural format for encoding digital geospatial information. A number of widely used conventions exist in order to promote interoperability across netCDF files. The Climate and Forecast (CF) conventions in particular are a widely used in the encoding of geospatial data in netCDF and are also an OGC standard. Whilst this standard uses concepts which have grown out of the World Wide Web Consortium (W3C), namely the Linked Data approach, this standard builds specifically on netCDF and the outputs of OGC's netCDF Standards Working Group, hence this standard belongs in the OGC suite.

While netCDF allows metadata to be specified in each file, it lacks scalable mechanisms for specifying metadata that conforms to community or project conventions or standards. Conformance to metadata conventions is usually achieved by one-off tools for validation or limited to manual checking. Linking to external resources on the web (e.g. controlled vocabulary terms, related files) can be implemented as simple strings in attribute values, however, the mechanism for implementing this is not standardised, nor are the tools to form knowledge graphs from these resources.

The netCDF-LD standard enables the annotation metadata in netCDF files in a way that is interoperable and reusable, and with minimal changes to current netCDF syntax. This means existing files can be interpreted using netCDF-LD tools. netCDF-LD proposes mechanisms such as encoding rules that allow users to precisely reference shared attribute names (e.g. property names) and shared values (e.g. controlled vocabulary terms). This facilitates consistency in specifying metadata across netCDF files. These netCDF-LD mechanisms thus allow existing netCDF files to be interpreted as Linked Data, and this facilitates interoperability between netCDF files (and potentially non-netCDF data files). Therefore, data providers adopting netCDF-LD are able to interpret existing standard netCDF metadata as well as annotate additional metadata in netCDF files that conform to one or more metadata conventions in a consistent way.

A key aspect of netCDF-LD is the introduction of rules for mapping identifiers for netCDF global and variable attributes and variables to Uniform Resource Identifiers (URIs) via *prefix* and *alias* mechanisms. URIs or identifiers are the building blocks for defining unambiguous semantics via concept definitions and relationships between concepts. Using these mechanisms, netCDF-LD provides a way to disambiguate attribute names and values shared among different netCDF files via identifiers.

netCDF's current use allows up to petabytes of data to be written out, sometimes organised as data files in series across thematic (observable properties), temporal (daily, monthly, yearly) or geographic regions. Users of the data may want to query or discover from those files. The current tools for handling this include software such as THREDDS and ERRDAP that provide APIs and functionality for searching and discovery.

netCDF-LD enables the potential for search and discovery in a different way to current netCDF software tools by leveraging the Linked Data representation of netCDF metadata as knowledge graphs. netCDF-LD allows summary metadata from any netCDF file to be extracted to provide an external description of the file in RDF. This enables references to externally published resources in RDF-specified metadata conventions. These descriptions can be used with other Linked Data technologies including the RDF query language, SPARQL (Harris and Seaborne 2013), and RDF triple stores. These descriptions can also be used to support data fusion from other data sources that have RDF representations also, i.e. non-netCDF data sources.

In summary, netCDF-LD provides mechanisms that allow interpreting and annotation of metadata to conform to community or project conventions/standards in a consistent way leveraging Linked Data approaches. In doing so, netCDF files are able to be interpreted into RDF graphs, which in turn can be used with other Linked Data technologies for data fusion, powerful querying and data discovery.

### 6.1.1. Design principles

- **Compatible with existing netCDF encoded files.** netCDF-LD encoded files should be compatible with existing netCDF encoded files. This means existing netCDF libraries can be used to parse files which have been encoded as netCDF-LD. This also ensures that netCDF-LD encoded files can be used as simply as netCDF, with no knowledge of RDF.
- **Existing netCDF encoded files can be used with netCDF-LD tools.** Existing netCDF files can be used with netCDF-LD files. That is, existing netCDF files can be represented as RDF using netCDF-LD tools.
- **Default behaviour while preserving flexibility.** netCDF-LD mechanisms and implementations should have consistent and useful default behaviour/configuration, but allow for user-defined configuration if needed.
- **Explicit semantics.** netCDF-LD promotes explicitly specifying semantics even if they implicit in netCDF files, e.g. aliases for well-known netCDF attribute names are made with explicit web-resolvable identifiers.
- **Leverage existing Linked Data approaches and tools.** The netCDF-LD standard should promote and adopt existing web and Linked Data approaches tools rather develop bespoke functionality, e.g. query language for netCDF-LD RDF content should use SPARQL.

## 6.2. Resource Description Framework

The *Resource Description Framework (RDF)* is a W3C standard (Hayes and Patel-Schneider 2014). Information elements consist of RDF statements. An RDF statement consist of three parts: *Subject*, *Predicate*, and *Object*. This is called a *triple*.

Identifiers are a key building block of RDF and *Uniform Resource Identifiers (URIs)* are used to identify a resource. URIs can appear in each part of a triple, i.e. *Subjects*, *Predicates* and *Objects*. Where an *Object* is a *URI*, this may also be a *Subject* for another statement.

Literals are another key part of RDF and are used to capture basic values other than a URI. e.g. strings such as "Gerald", dates such as "1 April 2019", and numbers such as "4897". Literals are associated with a data-type so that they can be parsed. Literals may only appear in the *Object* part

of a RDF statement.

RDF statements are collected together within an *RDF Graph*. Figure 1 below show an example of an RDF graph encoding information about the singer B.B. King.

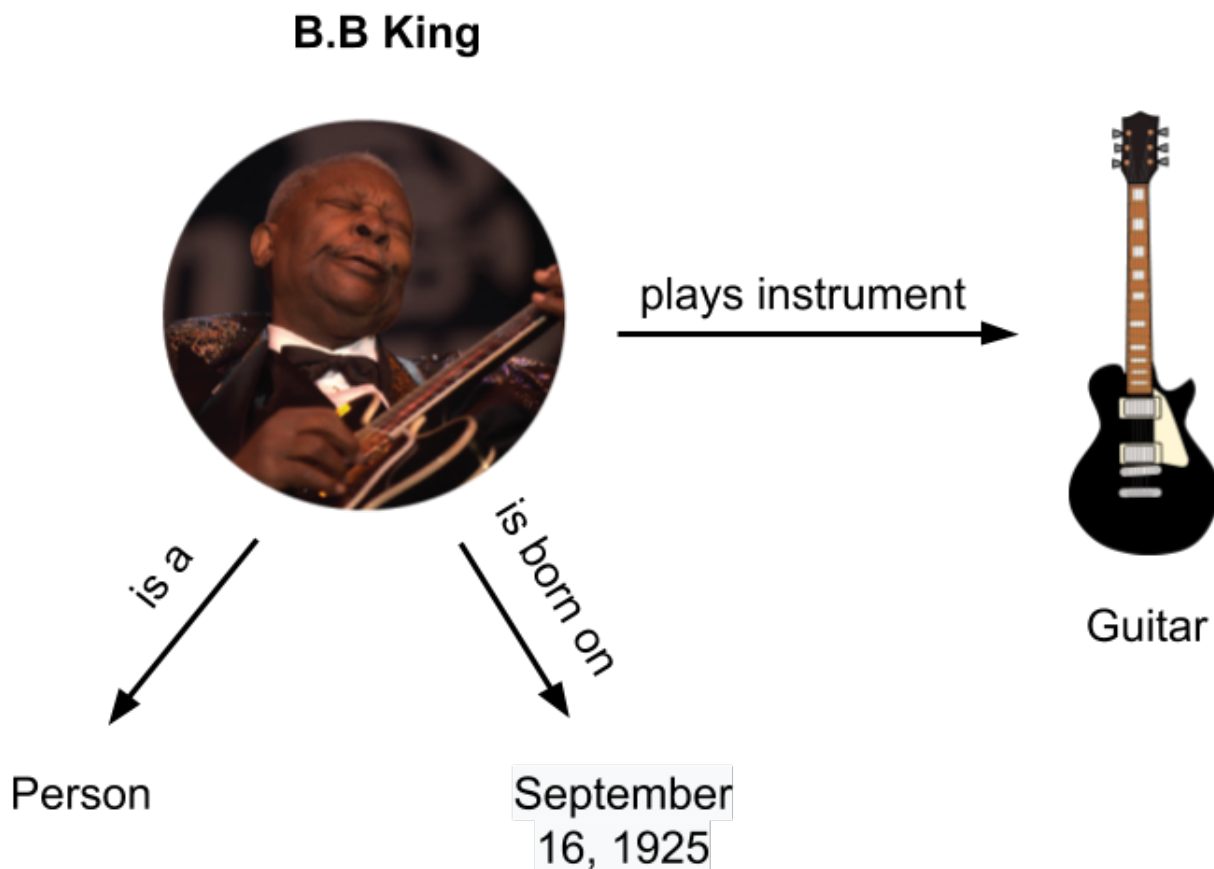


Figure 1. RDF Example: BB King

```
<B.B. King> <is a> <person>.  
<B.B. King> <is born on> <the 16th of September 1925>.  
<B.B. King> <plays instrument> <guitar>
```

The netCDF-LD standard maps the netCDF data model encoding onto the RDF information model.

### 6.3. NetCDF File

The Unidata network Common Data Form (netCDF) is an interface for scientific data access and a freely-distributed software library that provides an implementation of the interface. The netCDF library also defines a machine-independent format for representing scientific data (<https://github.com/Unidata/netcdf-c>), which may consist of multiple dimensions (Figure 2 and Figure 3).

NetCDF is an encoding format with open semantics and inferred referencing. Metadata may be defined on all variables and groups within a netCDF file, with each entity having its own name space for locally unique attribute names and each attribute containing a single value.



For targeted metadata specifications, such as defining referenced spatio-temporal coordinates, metadata conventions are used.

Metadata conventions and internal referencing enable a detailed representation of semantics describing the arrays of data values encoded within the file.

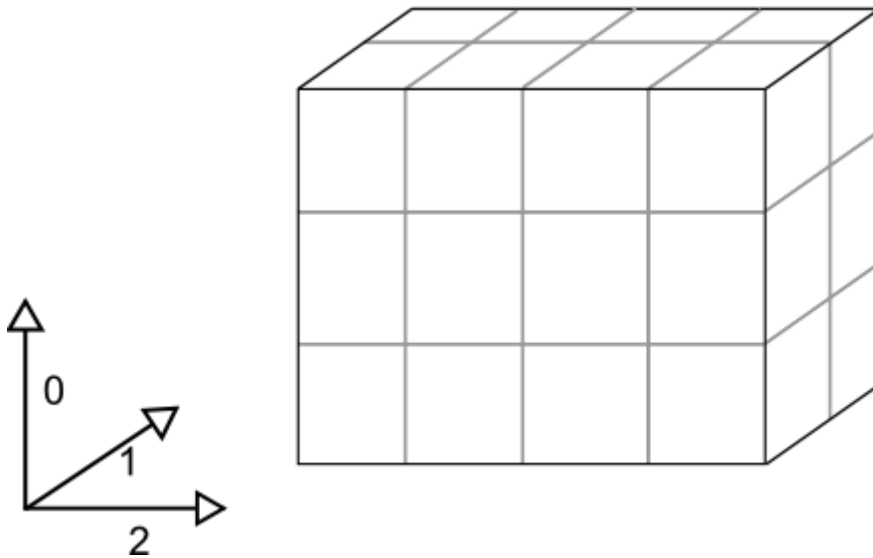


Figure 2. Multi-dimensional Array

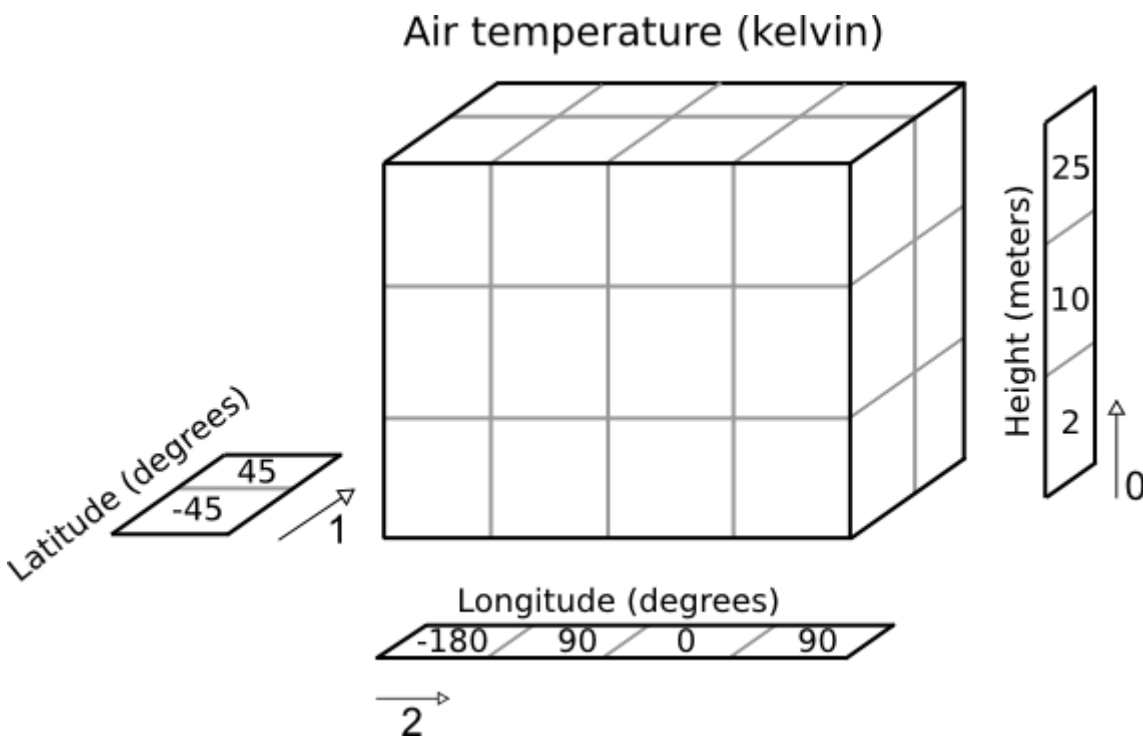


Figure 3. Array with Arrays Defining Dimensions

## 6.4. RDF Vocabularies

A set of RDF vocabularies are used within this standard to allow the representation of structural and semantic content of netCDF metadata within the context of an RDF graph. Where possible widely adopted RDF vocabularies are used before creating new vocabularies. This allows the ability to combine data graphs from different sources, promotes data harmonisation, and the ability to query them in an interoperable way (i.e. using the same vocabularies and predicates for like-for-

like semantics).

A new vocabulary is defined by the Open Geospatial Consortium, the Binary Array Linked Data vocabulary: **bald** (<https://www.opengis.net/def/binary-array-ld/>). The **bald** vocabulary (Figure 4) defines a small set of concepts which enable the structural elements of generic binary array containers, including netCDF, to be represented within an RDF graph.

This standard also makes use of the core RDF vocabularies, the Dublin Core vocabulary, the schema.org vocabulary and the Data Catalogue Vocabulary (DCAT):

- rdf (<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
- rdfs (<http://www.w3.org/2000/01/rdf-schema#>)
- dct (<http://purl.org/dc/terms/>)
- sdo (<http://schema.org>)
- dcat (<http://www.w3.org/ns/dcat#>)

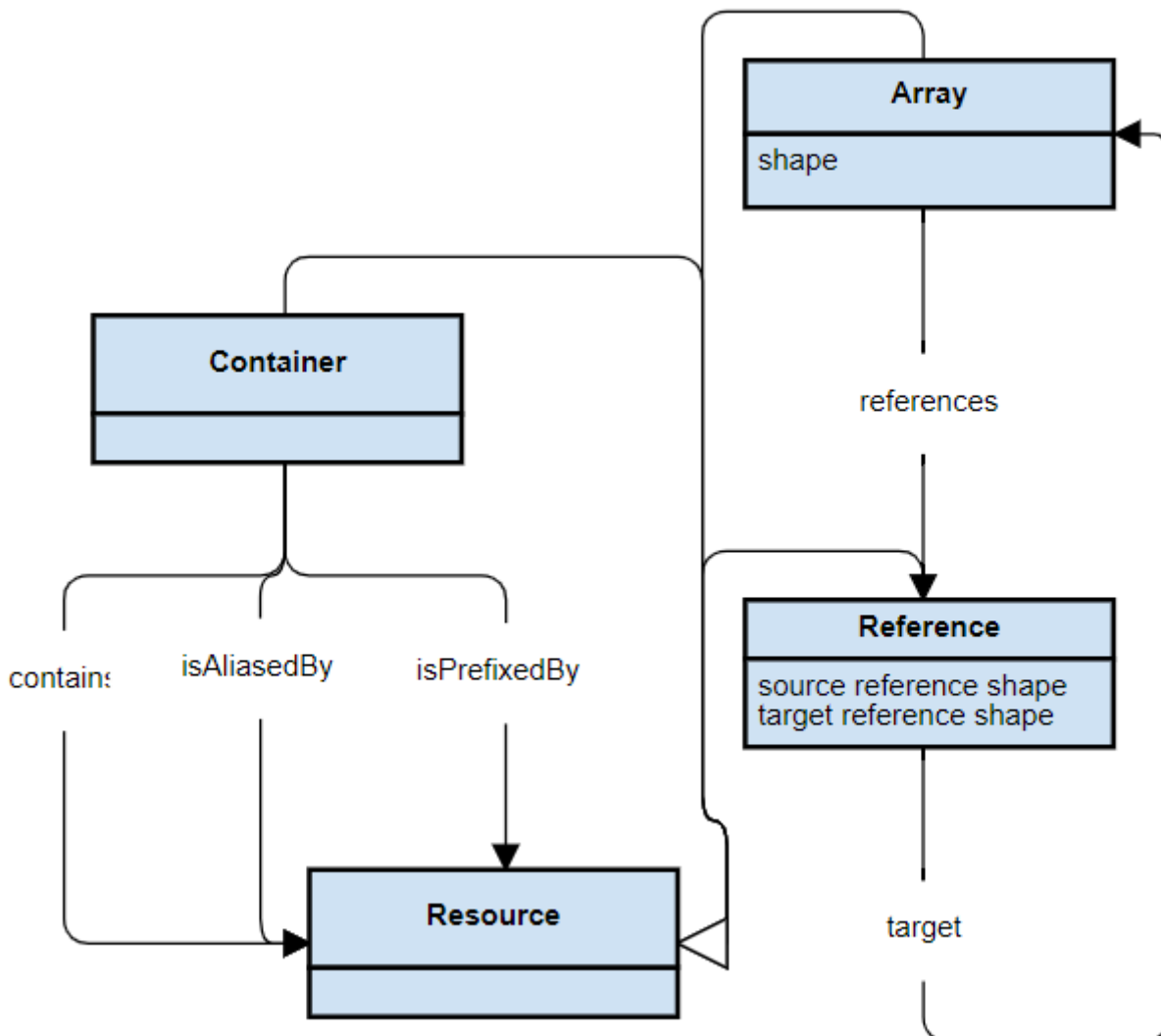


Figure 4. Binary Array Linked Data (BALD) vocabulary

## 6.5. Metadata RDF Graph

An *RDF graph* representing the metadata of multiple files provides a valuable resource for summarising the file collection.

The numerical data within the file is not encoded within the graph, only the metadata. In general, this means that the graph is orders of magnitude smaller than the data file it summarises.

Having an *RDF graph* of metadata across a file collection enables the use of Semantic Web technologies to explore, interrogate and visualise the metadata from that file collection in a variety of ways. In Figure 5, an example is given showing the CF standard name attribute value across multiple files.

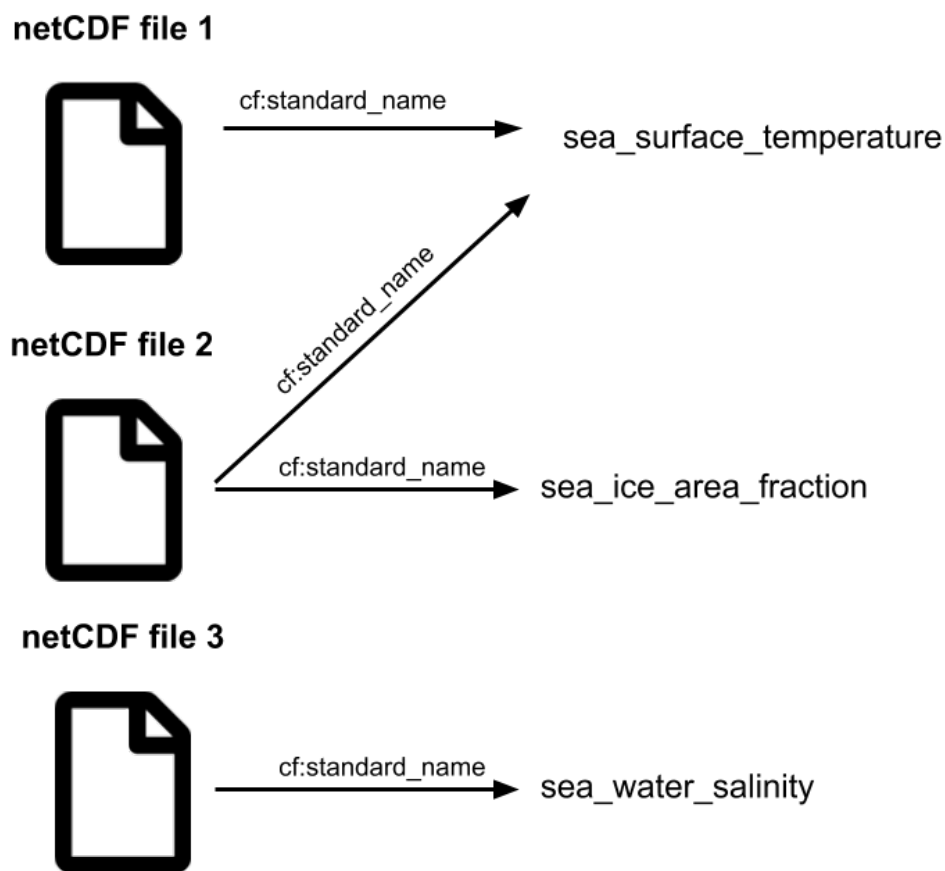


Figure 5. Example showing `cf:standard_name` property values across different files

Representing this and other metadata enables queries across netCDF files, e.g. find netCDF files that have data values on `sea_surface_temperature`.

The following sections of this standard define how to interpret a netCDF file as an RDF metadata graph.

Multiple graphs may be combined into a single multi-file metadata graph providing individual file identities are unique across the multi-file metadata graph.

### 6.5.1. Uniform Resource Identifiers within netCDF files

At the core of this standard is the mechanism for encoding and interpreting Uniform Resource Identifiers (URIs) within netCDF files.

These URIs may be declared explicitly within the files, or interpreted from file metadata and external information sources.

### 6.5.2. Identity

In netCDF-LD, a *netCDF file* has an identity. This identity provides the mechanism to obtain the file. As it is mutable, it is dependent on how the file is provided. Two systems may provide an identical file, but in different ways, and decide to use different identities. The identity is not inherent to the metadata payload of the file.

An explicit identity, i.e. a URL or URI, may be provided during file interpretation (i.e. by netCDF-LD parsers).

If no identity is provided, then the file path (**file:**) or remote location (**http:** / **https:**) used by a library to access the file is used as a default identity. If used, a local file URI will always use the '/' forward slash as an element separator, even on systems where local identifiers use back slash separators.

The identifier for the netCDF file is the identity of the root group, that is, the base entity within the netCDF file. This provides an identity root for the contents of the file. For this reason, this standard mandates that the identifier string will always terminate in a '/' character. In this way the root group's identity is distinguished as a different conceptual entity from the file itself.

Two examples are provided below showing a file URI from a location, via a URL, which also serves as a URI for that file object (Example 1). Example 2 shows the root group, within that file, may reuse the identity string, as a compound part, with the addition of the extra '/' character, thereby differentiating itself from the file object identity.

1. Example 1. [https://www.unidata.ucar.edu/software/netcdf/examples/test\\_hgroups.nc](https://www.unidata.ucar.edu/software/netcdf/examples/test_hgroups.nc) (the URI identity of the netCDF file object)
2. Example 2. [https://www.unidata.ucar.edu/software/netcdf/examples/test\\_hgroups.nc/](https://www.unidata.ucar.edu/software/netcdf/examples/test_hgroups.nc/) (the URI identity of the root group contained within that netCDF file object)

#### Variable Identity

Each netCDF variable within the root group, each netCDF group and each netCDF variable within each group shall take its identity as a URI, relative to the file identity.

This identity is defined using the full variable name, including any owning groups, appended to the file identity. Each element separation uses the / character.

Thus, each variable and group within that file has its own identity, defined relative to the file identity.

## Variable Type Declaration

Each netCDF variable shall declare a type statement, and RDF triple of the form

```
<variableURI> rdf:type <type>
```

There are two recognised type identifiers within the standard.

### Array Variable Type Identifiers

If the variable contains an array data payload (which may be missing data) then the type shall be defined as bald:Array and shall include a statement defining the shape of the data payload array as a Literal encoded tuple of integer values, using bald:shape.

The shape Literal is the shape as defined and ordered within the netCDF file, using the numerical values of the dimensions to define the actual shape.

The names of dimensions within the netCDF file encoding are not stored within the metadata summary graph.

```
<variableURI> rdf:type bald:Array ;  
                bald shape (1,1,16,9) .
```

### Resource Variable Type Identifiers

If there is no array payload and the variable is single valued (which may be missing data) then the type shall be defined as bald:Resource.

In the bald vocabulary, bald:Resource is the general type, bald:Array is a specialisation of this type.

```
<variableURI> rdf:type bald:Resource .
```

If the bald:Array instance has a single dimension, then the first and last values from the data payload, as ordered within the netCDF file, shall be encoded as Literals within the metadata graph, using bald:firstValue and bald:lastValue.

For example:

```
<variableURI> rdf:type bald:Array ;  
                bald shape '(72,)' ;  
                bald:firstValue -90 ;  
                bald:lastValue 90 .
```

## Download URL

The identity is conceptually distinct from the resolvable location of a file.

Identity and Download URL are optional inputs to the metadata graph creation process.

If a download URL is provided, but no Identity is provided, then an identity shall be inferred by the appending of a '/' character to the location.

If a download URL is provided, then it shall be recorded as part of the `dcat:distribution` definition.

As a `bald:Container` is a subclass of `dcat:Distribution`, DCAT is used to describe the file type and to provide an optional statement to specify the resolvable location of the file object, using `dcat:downloadURL`.

```
@prefix this: <https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract_tests/netCDF/ogcClassEF.nc/> .

this: a bald:Container ;
  dcat:distribution [
    a dcat:Distribution;
    dcat:downloadURL <https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract_tests/netCDF/ogcClassEF.nc>;
    dcat:mediaType [
      a dct:MediaType;
      dct:identifier "application/netcdf"
    ];
    dct:format [
      a dct:MediaType;
      dct:identifier <http://vocab.nerc.ac.uk/collection/M01/current/NC/>
    ]
  ].
```

netCDF-LD implementations shall enable the file location to be provided at run time, separate from the identity, to define from where the file object may be obtained.

netCDF-LD implementations shall not provide a `dcat:downloadURL` statement unless specified at run time.

### 6.5.3. Containment

`bald:Container` instances use a simple containment behaviour, provided by the `bald:contains` property to represent the containment of variables and groups within groups and files.

### 6.5.4. Prefix Definition

In netCDF-LD, the *prefix* is a mechanism to encode explicit URIs from elements in the *netCDF file*. It is the most straightforward and the most powerful way of doing so. It has two parts: the first part is the declaration of a prefix (or a set of prefixes), the second part is the use of the prefix.

Prefixes are in wide use in a number of domains, including XML and RDF. They allow a URI to be expressed in a compact fashion that saves space, enhances human readability and can mitigate issues with reserved characters.

## Prefix Declaration in File

Prefixes may be declared inside the file using a name-value-pair that associates a short name (e.g. `cf`, `ba1d`), with a URI.

The attributes defining prefixes shall be in a separate netCDF group or variable, as attributes.

The prefixes group shall not be interpreted as part of the graph, it is used only in the interpretation of URIs, which will be encoded as explicit RDF prefixes in RDF encodings.

A single prefix declaration is an attribute and a value: the attribute name is the prefix name and the attribute value is the full URI for that prefix. e.g.

```
bald__ = https://www.opengis.net/def/binary-array-ld/
```

The 'double underscore' character pair: `__` is used as an identifier and as the termination of the prefix; the double underscore is part of the prefix string within the netCDF file.

The double underscore is interpreted as a special character by this standard: the first use of a double underscore from the start of any string shall be interpreted as a prefixed entity and shall be unpacked into full URIs by aware software if a full URI is defined.

A prefix defined within a file shall only be used as a prefix if it ends in a double-underscore character pair, `__`.

Prefixes are applied across the file they are declared within. A single dedicated group shall be used to contain all of the internally defined prefixes applied to that file.

The prefix group is optional.

If included the prefixes group shall be identified within the file by a single global attribute, using the attribute name `bald__isPrefixedBy`.

If included, the prefixes group shall include the `bald` prefix declaration.

The definition of multiple prefix resources within a single file is invalid in this standard. netCDF-LD implementations may choose to combine prefix collections in invalid cases, but no precedence is implied, and prefix conflict is unresolved. netCDF-LD implementations may treat this condition as a warning condition and as a validation error.

## The Double Underscore Syntax

This standard introduces a special text syntax, a double underscore `'__'` as a separator between the prefix and suffix.

The choice of special character has been chosen as a compromise of various factors. RDF uses a `:` (colon) as a separator between prefix and suffix, but the netCDF CDL specification uses the colon as a variable attribute separator. As such, the use of a colon as a separator requires escape characters to be used.

Older versions of the netCDF libraries precluded the use of colon at all, the ability to use the escape character was introduced, but some use of double underscore was already in use in ad-hoc file encodings. Maintaining backwards compatibility is deemed a useful facet of this syntax.

The use of attribute names with colon separators is a limiting factor in writing code, as the attribute name can only be used within a string in many programming languages (e.g. python), and not as a variable name or identifier.

Early adopters of the approaches being standardised in this document have been using the double underscore to namespace locally defined attribute names for a significant period of time, in the hope that this syntax would become standardised. There is fairly widespread use of this namespace syntax in existing files across a number of archives of earth system geospatial data.

Given the ad-hoc usage of the double underscore and its utility in programming languages and data specifications, this standard adopts and standardises the double underscore as the prefix separator for this standard.

### Externally Defined Prefixes

Prefixes may be defined at runtime, by providing parse-able JSON-LD context files or contents.

Prefixes will be interpreted during parsing from all context files and internally defined prefixes in combination.

Prefixes in context files shall be defined as RDF prefixes in JSON-LD. This means that there is no prefix separator within the JSON-LD context file. The prefix, defined in the JSON-LD context file shall be interpreted as the prefix string appended by a double-underscore `__` within the netCDF-LD contextual interpretation.

For example, the prefix `bald` would be defined within a JSON-LD context file as:

```
{'@context': {'bald': 'https://www.opengis.net/def/binary-array-ld/'}}
```

The parsing library would interpret this as equivalent to the definition of a prefix within a file in CDL:

```
group: prefix_list {
  :bald__ = "https://www.opengis.net/def/binary-array-ld/" ;
}
```

and thus match to attributes within the file, such as in CDL:

```
// global attributes:
      :bald__isPrefixedBy = "prefix_list" ;
```



## Prefix Conflict

If a prefix string is defined multiple times in JSON-LD context files, with different URI interpretations, then implementations shall ignore that prefix and treat the prefix as locally unresolved. Implementations may choose to raise warnings, validation errors, etc. in these cases.

It is expected that files will be able to be parsed, even if prefix conflicts exist. Conflict in prefix definitions is not a violation of this standard, the fall-back position is to ignore conflicting prefixes as not well defined at runtime.

Prefix definitions provided explicitly within a file shall not be over-ridden by context files. Prefixes defined within a file have precedence.

If a prefix defined within a file is also defined within provided JSON-LD context files with different URI interpretations, then implementations shall ignore that JSON-LD context definition and treat the prefix as locally resolved.

This standard does not consider this as a parsing error.

Implementations may treat this condition as a warning condition and as a validation error within the provided context.

## Prefix use

A prefix is used with netCDF file elements as:

```
<prefix><name>
```

This is interpreted within this standard as a URI, a concatenation of the matching value within the prefix variable and the remainder of the attribute name or value.

The attribute name

```
bald__isPrefixedBy
```

together with the prefix definition

```
bald__ = https://www.opengis.net/def/binary-array-ld/
```

is interpreted as

```
bald__isPrefixedBy = https://www.opengis.net/def/binary-array-ld/isPrefixedBy
```

Prefixes shall end in a URI separator, either a / or a #.

The following example, in netCDF Common Data Language (CDL), uses the <https://www.opengis.net/def/binary-array-ld> and the [w3.org rdf-syntax-ns](http://www.w3.org/rdf-syntax-ns) vocabularies to describe a reference relationship

between two variables.

```
netcdf tmpMwXy8U {
dimensions:
    pdim0 = 11 ;
    pdim1 = 17 ;
variables:
    int a_variable(pdim0, pdim1) ;
        a_variable:rdf__type = "bald__Array" ;
        a_variable:bald__references = "b_variable" ;
    int b_variable(pdim0, pdim1) ;
        b_variable:rdf__type = "bald__Reference" ;
        b_variable:bald__array = "b_variable" ;

// global attributes:
    :bald__isPrefixedBy = "prefix_list" ;
group prefix_list {
    :bald__ = "https://www.opengis.net/def/binary-array-ld/" ;
    :rdf__ = "http://www.w3.org/1999/02/22-rdf-syntax-ns#" ;
}
}
```

In this example:

- `rdf__type` is interpreted as <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
- `bald__array` is interpreted as <https://www.opengis.net/def/binary-array-ld/array>

### 6.5.5. Alias Definition

Alongside the definition of prefixes, explicit aliases may be defined via netCDF-LD conventions within the *netCDF file*, or as a scope for a *netCDF file* during parsing. Aliases enable controlled attribute names to be interpreted as URIs.

Alias definitions do not exist with file CDL. They are provided as input (either as RDF or JSON) to a parsing process at parse time, from external vocabularies.

Alias URIs are interpreted from a reverse lookup from the file into the graph. Consequently there is no protection against name clashes.

For an entity in an alias graph to be considered as an alias, the entity will define a RDF statement:

```
<$entity> <http://purl.org/dc/terms/identifier> "$Literal" .
```

The Literal object of this RDF statement is the alias name.

An alias that may be used as an attribute name alias shall define its Type as

```
<$entity> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  
<http://www.w3.org/1999/02/22-rdf-syntax-ns#Property> .
```

or

```
<$entity> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  
<http://www.w3.org/2002/07/owl#ObjectProperty> .
```

The alias mechanism is less flexible than the prefix mechanism. It does enable interpretation of attribute names directly, making it useful for existing standards and existing files.

Aliases are applied across the file for which they are declared.

Aliases are declared as a set of RDF graphs. These RDF graphs are commonly provided as URIs, to be obtained during parsing and file metadata interpretation.

The RDF graphs shall be combined and treated as a single alias scope for the file.

An example of alias definitions in an external file is shown below ,encoded as TTL in a separate file for names.

```
# Names Aliases  
@prefix dct: <http://purl.org/dc/terms/> .  
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .  
@prefix acdd: <https://def.scitools.org.uk/ACDD/> .  
@prefix cfterms: <https://def.scitools.org.uk/CFTerms/> .  
@prefix nc: <https://def.scitools.org.uk/NetCDF/> .  
@prefix op: <http://environment.data.gov.au/def/op#> .  
  
acdd:id                dct:identifier "id" .  
acdd:title            dct:identifier "title" .  
cfterms:standard_name  dct:identifier "standard_name" .  
cfterms:units         dct:identifier "units" .  
nc:valid_min          dct:identifier "valid_min" .  
nc:valid_max          dct:identifier "valid_max" .  
op:matrix             dct:identifier "matrix" .  
op:objectOfInterest   dct:identifier "substanceOrTaxon" .  
skos:prefLabel        dct:identifier "prefLabel" .
```

The following shows the RDF example as a JSON-LD serialization:

```

{
  "@context": {
    "dct": "http://purl.org/dc/terms/",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "acdd": "https://def.scitools.org.uk/ACDD/",
    "cfterms": "https://def.scitools.org.uk/CFTerms/",
    "nc": "https://def.scitools.org.uk/NetCDF/",
    "skos": "http://www.w3.org/2004/02/skos/core#",
    "op": "http://environment.data.gov.au/def/op#"
  },
  "@graph": [
    {
      "@id": "skos:prefLabel",
      "dct:identifier": "prefLabel"
    },
    {
      "@id": "acdd:id",
      "dct:identifier": "id"
    },
    {
      "@id": "acdd:title",
      "dct:identifier": "title"
    },
    {
      "@id": "cfterms:standard_name",
      "dct:identifier": "standard_name"
    },
    {
      "@id": "cfterms:units",
      "dct:identifier": "units"
    },
    {
      "@id": "op:matrix",
      "dct:identifier": "matrix"
    },
    {
      "@id": "op:objectOfInterest",
      "dct:identifier": "substanceOrTaxon"
    },
    {
      "@id": "nc:valid_max",
      "dct:identifier": "valid_max"
    },
    {
      "@id": "nc:valid_min",
      "dct:identifier": "valid_min"
    }
  ]
}

```

Alias definitions, provided as input as a simple dictionary representation, are available as JSON. The alias dictionary representation encoded as JSON shall consist of only an object with name/value pairs for the alias and the mapped URI respectively. The value for an item in the JSON object shall be a string value for the URI.

An example of aliases for both names and values, expressed as a dictionary encoded as a JSON file, is shown below:

```
{
  "id"           : "https://def.scitools.org.uk/ACDD/id",
  "title"        : "https://def.scitools.org.uk/ACDD/title",
  "standard_name" :
  "https://def.scitools.org.uk/CFTerms/standard_name",
  "units"        :
  "https://def.scitools.org.uk/CFTerms/units",
  "valid_min"    :
  "https://def.scitools.org.uk/NetCDF/valid_min",
  "valid_max"    :
  "https://def.scitools.org.uk/NetCDF/valid_max",
  "matrix"       :
  "http://environment.data.gov.au/def/op#matrix",
  "substanceOrTaxon" :
  "http://environment.data.gov.au/def/op#objectOfInterest",
  "prefLabel"    :
  "http://www.w3.org/2004/02/skos/core#prefLabel"
}
```

### 6.5.6. Attribute Names

In order to map netCDF metadata to RDF, all global and variable attributes are interpreted as RDF statements. This requires that all attribute names are interpreted as URIs.

A parsing process shall map attribute names to URIs using prefix definitions first, then map attribute names to URIs aliases.

An attribute name shall be mapped to an alias URI if, and only if, there is an exact match for the full attribute name as a `dct:identifier` (expand to full URI) for an entity within the alias graph where that entity declares a RDF statement within its defining graph.

```
<entity> <rdf:type> <rdfs:ObjectProperty> .
```

That defining graph needs to be provided to netCDF-LD aware software at the time of parsing the file, so that it can be interpreted.

An error is created if multiple aliases match an attribute name in a *netCDF file* due to a conflict in unambiguously identifying the declared alias scope.

All remaining attribute names shall be mapped to local identifiers, using the file identity and

variable identity (`ref{}`) to form a locally applicable URI.

In the examples, the prefix `this:` is used within the graphs as the file identifier.

### 6.5.7. Variable-to-Variable References

The value of an attribute may be a reference to another variable, or multiple variables. The process of establishing identity for each variable within the file enables this reference to be interpreted as a URI. In this way, the RDF approach to having objects that are links to subjects, chaining RDF statements into graphs, is implemented.

For a reference to be identified, the predicate that defines that reference must identify itself as suitable for variable-to-variable referencing. No references will be inferred for predicates that do not identify themselves in this way.

To identify a predicate as a variable-to-variable reference predicate, that predicate shall provide a RDF statement that explicitly opts into this behaviour. The simplest way to do this is to include the RDF statement.

```
<{predicate}> rdfs:range bald:Resource .
```

Alternatively, it is permissible to provide an `rdfs:range` statement targeting another resource and for that resource to declare that it is an `rdfs:subClassOf` `bald:Resource`:

```
<{attribute}> rdfs:range <{AClass}> .  
<{AClass}> rdfs:subClassOf <https://www.opengis.net/def/binary-array-ld/Resource> .
```

This standard supports direct `rdfs:subClassOf` relations to enable variable to variable references. This standard does not support nested derivation, thus an implementation is not required to search the inheritance tree of a defined Class to see if any of the classes, which it inherits, define themselves as `rdfs:subClassOf` `bald:Resource`.

The definition of the owl:objectProperty `bald:references` uses this `rdfs:range`, `rdfs:subClassOf` definition within the bald ontology: <https://www.opengis.net/def/binary-array-ld>

```

<https://www.opengis.net/def/binary-array-ld/references>
  a owl:ObjectProperty ;
  rdfs:domain <https://www.opengis.net/def/binary-array-ld/Array> ;
  rdfs:label "references" ;
  rdfs:range <https://www.opengis.net/def/binary-array-ld/Reference> ;
  dct:description "This Array references that Reference. " .

<https://www.opengis.net/def/binary-array-ld/Reference>
  a owl:Class ;
  rdfs:label "Reference" ;
  rdfs:subClassOf <https://www.opengis.net/def/binary-array-ld/Resource> ;
  dct:description "The definition of a reference from one Array to another." .

```

References to variables are implemented in netCDF files by defining the value of an attribute as the name of a variable, or as a space-separated set of names of variables, or as a parenthesis bound, space-separated, list of names of variables.

A set of references is explicitly unordered whilst a list of references is explicitly ordered.

CDL defining a set of references:

```

int set_collection ;
  set_collection:bald__references = "data_variable1 data_variable2" ;

```

will be interpreted into RDF(turtle) as:

```

ns1:set_collection a bald:Resource ;
  bald:references ns1:data_variable1_ref,
                 ns1:data_variable2_ref .

```

CDL defining a list of references:

```

int list_collection ;
  list_collection:bald__references = "( data_variable1 data_variable2 )" ;

```

will be interpreted into RDF(turtle) as:

```

ns1:list_collection a bald:Resource ;
  bald:references ( ns1:data_variable1 ns1:data_variable2 ) .

```

If matching fails, the fall-back option is to ignore the potential for references and leave the attribute value as a Literal.

## 6.5.8. Attribute Values

In RDF, objects may be Literals or URIs, therefore attribute values are conditionally interpreted as Literals or as URIs. A parsing process shall map attribute values to URIs using identified prefixes first.

### Attribute Variable References

The value of a variable attribute may be an internal reference to another variable within the file.

For an in-file variable reference to be declared, three conditions shall be met.

1. Condition one: the value is a string which exactly matches the name of a variable within the file.
  - netCDF LD uses the rules for referencing variables across groups defined by the CF Conventions for netCDF Files:
    - (<http://cfconventions.org/cf-conventions/cf-conventions.html#groups>);
    - 'search by absolute path' and 'search by relative path' shall be implemented;
    - 'search by proximity' is deprecated by CF, and may be implemented or not.
2. Condition two: the attribute name is already interpreted as a URI, defining an entity, external to the file.
3. Condition three: the attribute name entity declares an `<rdfs:range>` of `<bald:Resource>` (or uses `rdf:subClassOf` linkage).

An identified attribute reference shall map the attribute value to the identify of the matched variable within the file.

This identification takes place after prefixes are identified and mapped.

### Attribute Value Aliases

After prefix and reference interpretation, remaining attribute values are mapped to URIs using the alias graph.

An attribute value shall be mapped to an alias URI if and only if there is an exact match for the full attribute value as a `dct:identifier` (expand to full URI) for an entity within the alias graph.

If multiple aliases match an attribute name, this is an error condition, the declared alias scope cannot be uniquely applied to the file.

### Attribute Value Literals

All remaining attribute values shall be left unchanged and declared as instances of `<rdf:Literal>`.

## 6.5.9. Reference Exceptions

The BALD vocabulary elements:



```
bald:isPrefixedBy  
bald:isAliasedBy
```

are exceptions within the standard.

They are variable to variable reference definitions, to provide capability within this standard. They are not metadata content for the file metadata summary.

```
bald:isPrefixedBy  
bald:isAliasedBy
```

statements shall not be represented within the standard output metadata graph. Target variables and groups from these variable to variable references shall be omitted from the output metadata graph.

## 6.6. NetCDF Dimensions

NetCDF makes strong inferences regarding how variables are defined by dimensions.

NetCDF files define named dimensions, `dims`. Each dimension defines a size, which is used as an array dimension. NetCDF Variables use dimensions to define their size and shape and to define some implicit references between each other (Figure 6).

In this way the netCDF variables are defined with respect to shared dimensions (Figure 7).

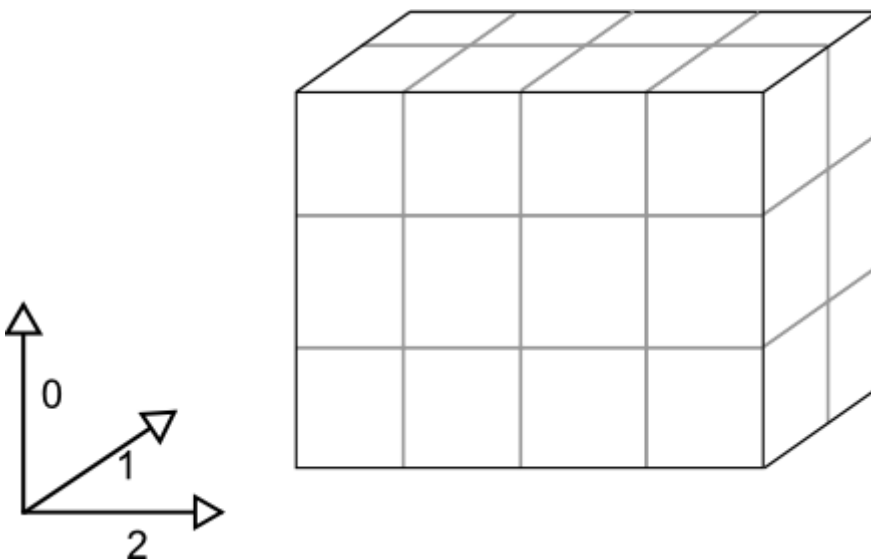


Figure 6. Multi-dimensional Array

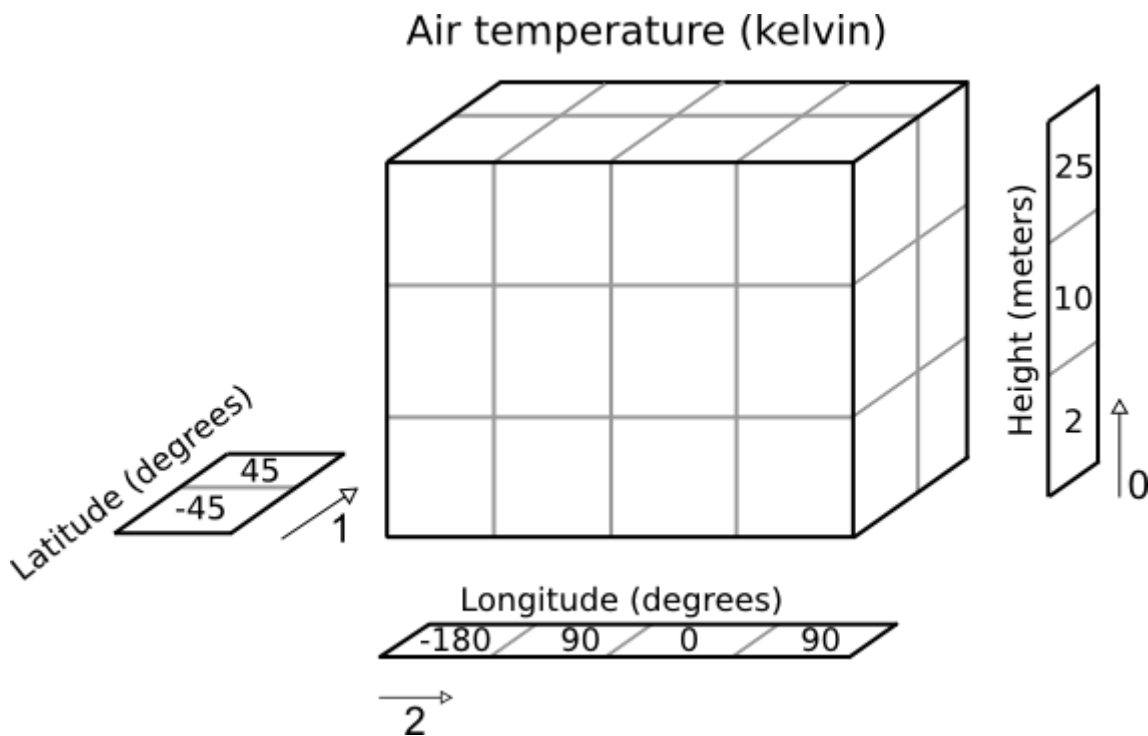


Figure 7. Array with Arrays Defining Dimensions

NetCDF-LD uses the dimensions to interpret the size and shape of a variable array.

NetCDF-LD does not explicitly encode the dimensions: only the sizing and referencing information. In cases where dimensions do not have a netCDF coordinate variable defined, this results in the name of the dimension being lost.

Extensive Variables are variables defined with respect to one or more dimensions.

In netCDF-LD, the size and shape of each extensive variable is explicitly stored as a RDF statement made with respect to that variable. The predicate `bald:shape` is used. Objects of this predicate shall be `rdf:Literal` instances.

Each extensive variable shall be described by a RDF statement.

```
<$entity> <https://www.opengis.net/def/binary-array-ld/shape> ($d0 $d1 $d2)
```

where `$dn` is an integer, taken from the defined dimension size in the netCDF file, and the count of the number of values is the dimensionality of the variable.

The object of this statement is an RDF List.

### 6.6.1. Broadcasting

In order to interpret netCDF dimensions within RDF graphs, the concept of Broadcasting is presented here, and used to implement array referencing.

Broadcasting enables arrays which share some dimensions, but have different overall dimensionality, to be interpreted together. Two arrays may be broadcast if the dimensions they share are ordered the same and extra dimensions can be interpreted unambiguously.

The result of broadcasting is an array shape which can represent the contents of each of the two input arrays, with extra dimensions containing copies of the defined values. In other words, an array may be stretched

In this way, an array location in the broadcast result array can interpret one, and only one, value from each of the input arrays.

The concept of broadcasting is defined in the Python Numpy library <https://numpy.org/devdocs/user/basics.broadcasting.html>, from where Figures 8-11 and the associated descriptions are sourced.

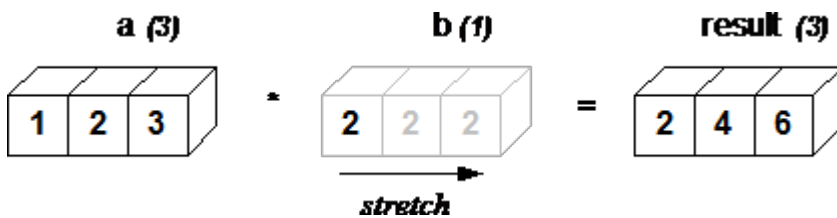


Figure 8. Scalar *b* is stretched to become an array of same shape as *a* so the shapes are compatible for element-by-element multiplication

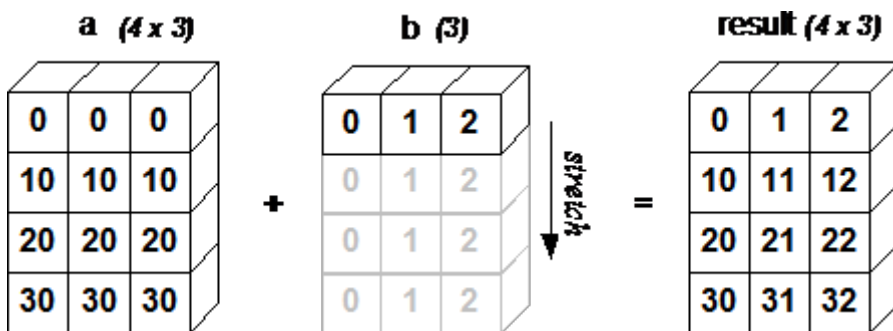


Figure 9. A one dimensional array added to a two dimensional array results in broadcasting if number of 1D array elements matches the number of 2D array columns

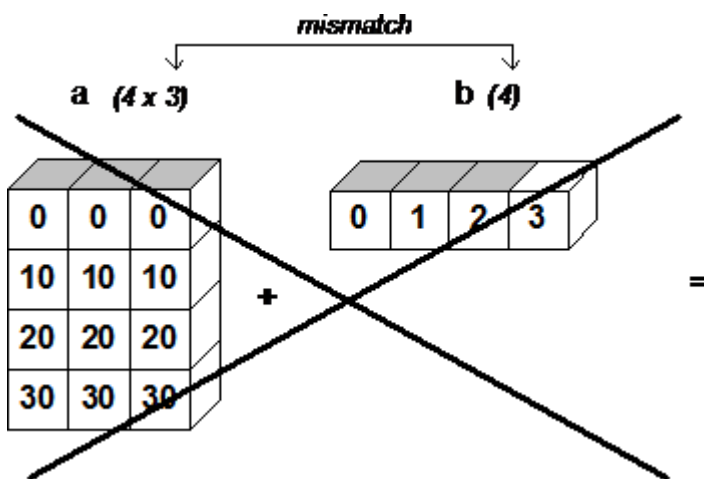


Figure 10. When the trailing dimensions of the arrays are unequal, broadcasting fails because it is impossible to align the values in the rows of the 1st array with the elements of the 2nd arrays for element-by-element addition

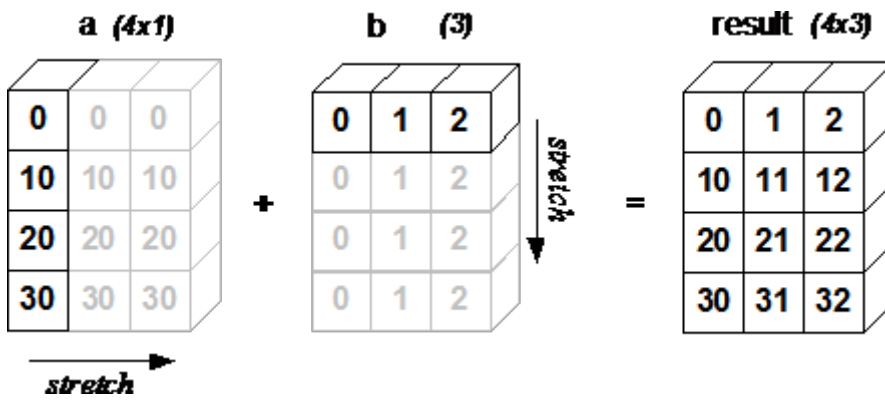


Figure 11. In some cases, broadcasting stretches both arrays to form an output array larger than either of the initial arrays

### 6.6.2. Shape and Reshape

In order to describe the results of the broadcast concepts in an implementation neutral fashion, this standard uses the shape and the concept of reshape.

The shape of an array is an ordered list of integers, representing the dimensionality of a multi-dimensional array and the size of each dimension.

Reshape is an operation which changes the dimensionality of an array, whilst preserving the number of values within the array.

Any array shape may be reshaped by adding elements to the array shape list, where the added items are of size 1. In situ elements of the shape that are not size 1 may not be reordered by a reshape.

For example:

```
bald:sourceRefShape (5, 9)
bald:targetRefshape (1, 1, 5, 9, 1)
```

### 6.6.3. Variable References and Dimensions

NetCDF-LD uses the defined netCDF dimensions to interpret references between variables and to interpret how the shapes of the variable arrays relate.

Each variable reference between variables defined with respect to netCDF dimensions is assumed to be an array-to-array relationship and that the arrays can be broadcast to enable a common indexing approach.

Broadcast is defined as extending an array along each of a set of defined dimensions, each of size 1, by copying the contents of the array for each array index up to the defined size. This enables an array to match shape with another array.

This interpretation means that it can be expected that the target in the reference relationship can be viewed with a dimensionality consistent with the source. A new shape is defined, with reshape, for the source and target, with size one values for dimensions that are to be broadcast. This imposes

ordering of the dimension for the broadcast operation.

Note: this is a key feature of netCDF dimensions: defining the size and commonality of array dimensions for variables.

In each case where a variable-to-variable reference is inferred within a netCDF file and both of the variables are defined with respect to one or more dimensions, the nature of that reference from the perspective of the arrays will be explicitly encoded within the RDF graph.

The inferencing of how array dimensions are matched and how this enables the interpretation of array broadcasting is subtle and implicit in netCDF, and specific to netCDF.

This information is unpacked and stored in a general fashion within the RDF graph.

All extensive variables have a shape encoded in the RDF graph. In order to interpret references, it is commonly required that a RDF statement, similar to the shape, is encoded, showing the reshaped shape that an array needs to be in order to properly broadcast. NetCDF-LD explicitly includes all reference RDF statements, even where the broadcast relationship can be inferred, for clarity and to aid comprehension.

A reshape array shape has the same total number of elements as the original array, but includes extra dimensions, of size 1, defining the order which the extensive dimensions are handled in.

To hold this information, netCDF-LD creates a new entity within the graph, representing this relationship, a `<$referenceEntity>`.

This `<$referenceEntity>` is referenced by the `<$sourceEntity>` using the predicate <https://www.opengis.net/def/binary-array-ld/references>, i.e.:

```
<$sourceEntity> <https://www.opengis.net/def/binary-array-ld/references>  
<$referenceEntity>
```

The `<$referenceEntity>` is defined to be of type <https://www.opengis.net/def/binary-array-ld/Reference> and shall define a single statement defining the target entity in the relationship, another variable in the file, using the predicate `bald:target`. The `<$referenceEntity>` shall define a `bald:targetShape` statement defining the reshape expansion of the target array. The `<$referenceEntity>` may define a `bald:sourceRefShape`, where that source shape is required to be different from the defining shape of the source array. The `<$referenceEntity>` shall define a `bald:targetRefShape`, whether that source shape is required to be different from the defining shape of the source array or whether the shape is the same.

In this manner, the source array and the target array are defined with a common dimensionality enabling the shape to be unambiguously defined for broadcasting; i.e.:

```

<$referenceEntity> a <https://www.opengis.net/def/binary-array-ld/Reference> ;
    <https://www.opengis.net/def/binary-array-ld/sourceRefShape> ($d1 <$d2 $d3 ...>)
;
    <https://www.opengis.net/def/binary-array-ld/targetRefShape> ($d4 <$d5 $d6 ...>)
;
    <https://www.opengis.net/def/binary-array-ld/target> <$targetEntity> .

```

$d_n$  are all defined to be integers. Each object of these RDF statements is an RDF list.

The object of the `bald:sourceRefShape` statement and `bald:targetRefShape` shall be an RDF list with equal numbers of elements.

The `sourceRefShape` is optional, it is assumed to be the same as the source array `bald:shape`, unless overridden by an explicit `sourceRefShape` statement. If the `sourceRefShape` is not provided, this is inferred to be the same as the defined `bald:shape`.

RefShapes never change the number of elements in an array nor the ordering of extensive (greater than size 1) array shape elements. A refShape array is only a higher dimensionality shape of a defined array shape, with dimensions of size 1 added to produce the desired dimensionality and dimension ordering.

The lengths of these two RefShape lists shall be the same, they represent the same dimensionality.

The product of the `sourceRefShape` (if provided) elements shall match the product of the `bald:shape`, the number of elements in the described arrays shall be the same.

The product of the `targetRefShape` elements shall match the product of the `bald:shape` of the `<$targetEntity>`, the number of elements in the described arrays shall be the same.

### Example Shapes and RefShapes

Shapes and RefShapes explicitly define the array dimension relationships that are implicit in the netCDF structure through the named dimensions.

If an array is defined with respect to named  $nc$  dimensions ( $da$ ,  $db$ ,  $dc$ ) giving it a shape of (13,17,7) and it references an array defined with respect to dimensions ( $db$ ) a shape of (17) then the effective reference shape for the relation is a `targetRefShape` of (1,17,1). If the same array of shape (13,17,7) references an array defined with respect to dimensions ( $da$ ) a shape of (13) then the effective reference shape for the relation is a `targetRefShape` of (13,1,1).

```

dimensions:
  da = 13 ;
  db = 17 ;
  dc = 7 ;
variables:
  int avar(da,db,dc) ;
  int db(db) ;
  int da(da) :

```

is represented as references in the metadata graph as:

```
this:avar a bald:array ;
  bald:shape ( 13 17 7 ) ;
  bald:references [ a bald:Reference ;
    bald:target this:db ;
    bald:targetRefshape ( 1 17 1 ) ],
  [ a bald:Reference ;
    bald:target this:da ;
    bald:targetRefshape ( 13 1 1 ) ] ;
.
this:db a bald:array ;
  bald:shape ( 17 ) .
this:da a bald:array ;
  bald:shape ( 13 ) .
```

This explicit reference is crucial for cases where there are unique dimensions of the same size, where inference could not be used. Given the generality of these, the standard mandates that `targetRefShape` shall always be defined, even if it could be inferred by the dimensionality size matching. This is to aid implementations using this information.

This reflects the explicit dimension naming within netCDF files. The references are stated explicitly in the file using these named dimensions. This standard does not preserve dimension names within the summary metadata graph but does represent these explicit array to array relationships and their cardinality.

If an array, defined with respect to named nc dimensions (de, df, dg), of shape (13,13,13), references an array defined with respect to dimensions (df), of shape (13), then there is no way to infer the correct `targetRefShape`, it must be specified, e.g. (1,13,1)

```
dimensions:
  de = 13 ;
  df = 13 ;
  dg = 13 ;
variables:
  int bvar(de,df,dg) ;
  int df(df) ;
```

is represented as references in the metadata graph as:

```

this:bvar a bald:array ;
  bald:shape ( 13 13 13 ) ;
  bald:references [ a bald:Reference ;
    bald:target this:df ;
    bald:targetRefshape ( 1 13 1 ) ] ;
.
this:df a bald:array ;
  bald:shape ( 13 ) .

```

If an array defined with respect to  $nc$  dimensions ( $di,dj,dk,dl$ ), of shape (13,17,13,7), references an array defined with respect to dimensions ( $dk, dm$ ), of shape (13,3) then both the source and target reference shape must be defined, e.g. `sourceRefShape (13,17,13,7,1)` `targetRefShape (1,1,13,1,3)`

```

dimensions:
  di = 13 ;
  dj = 17 ;
  dk = 13 ;
  dl = 7 ;
  dm = 3 ;
variables:
  int cvar(di,dj,dk,dl) ;
  cvar:bald__references = "lvar" ;
  int lvar(dk,dm) ;

```

is represented as references in the metadata graph as:

```

this:cvar a bald:array ;
  bald:shape ( 13 17 13 7 ) ;
  bald:references [ a bald:Reference ;
    bald:target this:lvar ;
    bald:sourceRefshape ( 13 17 13 7 1 ) ],
    bald:targetRefshape ( 1 1 13 1 3 ) ],
.
this:lvar a bald:array ;
  bald:shape ( 13 3 ) .

```

Where any ambiguity in ordering exists in the netCDF file for explicitly defined `sourceRefShape` and `targetRefShape` instances, due to partial overlap of named dimensions for both netCDF variables, then source reference dimensions shall be ordered before target reference dimensions.

In the above example the size 7 dimension is ordered before the size 3 dimension in the `RefShape` instances.

### Mismatched reference

Where metadata schemes provide further variable reference predicates, it is possible for variable-to-variable references to be defined where a broadcast relationship does not exist.



Applications may treat mismatches between reference definitions and the ability to broadcast as warning conditions, and skip the creation of RDF statements. In this case, it is recommended to continue to create a graph and omit references. It is accepted that an implementation may treat this as an error condition, and fail to create a graph. This is an implementation detail.

### Variable Reference Utility

The definition of the variable reference with its cardinality delivers specific capability to the resulting metadata graph.

References between arrays in a file are provided primarily to support partial access patterns from individual file objects. Where an element of interest is identified from a metadata graph, this element may be accessed directly, if supported by a suitable data supply interface, without the need to obtain the entire file object.

In order to support partial retrieval, it is essential that the metadata graph provides details on which array variables depend on other array variables and the nature of that dependency.

If two values from an array variable containing millions of elements in a file are requested, which other arrays within the file provide explicit metadata and need to be retrieved, and which elements within those arrays provide locational metadata? How should the two requested data values be described by metadata?

Without numerical array referencing, the only option for the user is to obtain the whole file and use an alternative netCDF tool to extract the values required with their associated metadata elements.

### 6.6.4. NetCDF Coordinate Variables

NetCDF defines a special type of variable, called a *Coordinate Variable*, which is identified by being one-dimensional and having the same name as the single dimension used to size the variable.

NetCDF-LD interprets Coordinate Variables as a case of variable referencing and includes entries as **bald:references** statements. There is an explicit exception to this, which is that self-references, from a coordinate variable to itself, shall not be added to the graph.

NetCDF-LD adds information on the Coordinate Variables, providing the first value, and optionally the last value, if there is more than one value. The first and last are taken from the order as presented within the file. If the value is encoded as a missing value, then no statement is included within the metadata graph.

NetCDF-LD uses the terms **bald:arrayFirstValue** and **bald:arrayLastValue** to identify these extracts from the Coordinate Data payload within individual statements in the resulting graph.

In this way, searchable information on the location in geospatial and conceptual coordinate space is extracted from the files, whilst the graph size increases by two statements per coordinate variable. The full coordinate array is not encoded within the metadata graph.

### 6.6.5. Worked Example

Here the definition of a netCDF file, in CDL, with all data array elements set as missing, is presented. It is followed by an RDF graph interpretation of the netCDF, illustrating many of the interpretation features described in this chapter.

```
netcdf tmpMwXy8U {
  dimensions:
    pdim0 = 11 ;
    pdim1 = 17 ;
    xy = 2 ;
  variables:
    int data_variable1(pdim0, pdim1) ;
      data_variable1:bald__references = "location_variable" ;
      data_variable1:long_name = "Gerald";
      data_variable1:obtype = "metce__SamplingObservation";

    int data_variable2(pdim0, pdim1) ;
      data_variable2:bald__references = "location_variable" ;
      data_variable2:long_name = "Imelda";
      data_variable2:obtype = "metce__SamplingObservation";

    int pdim0(pdim0) ;

    int pdim1(pdim1) ;

    int location_variable(pdim0, pdim1, xy) ;
      location_variable:bald__references = "location_reference_system" ;

    int location_reference_system;
      location_reference_system:pcode = "4897";

    int set_collection ;
      set_collection:bald__references = "data_variable1 data_variable2" ;

    int list_collection ;
      list_collection:bald__references = "( data_variable1 data_variable2 )"
;

  // prefix group
  group: prefix_list {
    :bald__ = "https://www.opengis.net/def/binary-array-ld/" ;
    :metce__ = "http://codes.wmo.int/common/observation-type/METCE/2013/" ;
    :rdf__ = "http://www.w3.org/1999/02/22-rdf-syntax-ns#" ;
  }
  // global attributes:
    :bald__isPrefixedBy = "prefix_list" ;

}
```

According to this standard, the netCDF file as defined above is interpreted into RDF, in the terse triple language (TTL) as

```
@prefix bald: <https://www.opengis.net/def/binary-array-ld/> .
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix metce: <http://codes.wmo.int/common/observation-type/METCE/2013/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix this: <file://CDL/multi_array_reference.cdl/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

this: a bald:Container ;
    dct:format [ a dct:MediaType ;
        dct:identifier <http://vocab.nerc.ac.uk/collection/M01/current/NC/> ] ;
    dcat:distribution [ a dcat:Distribution ;
        dcat:mediaType [ a dct:MediaType ;
            dct:identifier "application/netcdf" ] ] ;
    bald:contains this:data_variable1,
        this:data_variable2,
        this:list_collection,
        this:location_reference_system,
        this:location_variable,
        this:pdim0,
        this:pdim1,
        this:set_collection ;
    bald:isPrefixedBy "prefix_list" .

this:list_collection a bald:Resource ;
    bald:references ( this:data_variable1 this:data_variable2 ) .

this:set_collection a bald:Resource ;
    bald:references this:data_variable1,
        this:data_variable2 .

this:location_reference_system a bald:Resource ;
    this:pcode "4897" .

this:data_variable1 a bald:Array ;
    this:long_name "Gerald" ;
    this:obtype metce:SamplingObservation ;
    bald:references [ a bald:Reference ;
        bald:target this:pdim1 ;
        bald:targetRefshape ( 1 17 ) ],
        [ a bald:Reference ;
            bald:sourceRefshape ( 11 17 1 ) ;
            bald:target this:location_variable ;
            bald:targetRefshape ( 11 17 2 ) ],
        [ a bald:Reference ;
```

```

        bald:target this:pdim0 ;
        bald:targetRefshape ( 11 1 ) ] ;
    bald:shape ( 11 17 ) .

this:data_variable2 a bald:Array ;
    this:long_name "Imelda" ;
    this:obtype metce:SamplingObservation ;
    bald:references [ a bald:Reference ;
        bald:target this:pdim0 ;
        bald:targetRefshape ( 11 1 ) ],
        [ a bald:Reference ;
            bald:sourceRefshape ( 11 17 1 ) ;
            bald:target this:location_variable ;
            bald:targetRefshape ( 11 17 2 ) ],
        [ a bald:Reference ;
            bald:target this:pdim1 ;
            bald:targetRefshape ( 1 17 ) ] ;
    bald:shape ( 11 17 ) .

this:pdim0 a bald:Array ;
    bald:shape ( 11 ) .

this:pdim1 a bald:Array ;
    bald:shape ( 17 ) .

this:location_variable a bald:Array ;
    bald:references [ a bald:Reference ;
        bald:target this:pdim1 ;
        bald:targetRefshape ( 1 17 1 ) ],
        [ a bald:Reference ;
            bald:target this:pdim0 ;
            bald:targetRefshape ( 11 1 1 ) ],
        this:location_reference_system ;
    bald:shape ( 11 17 2 ) .

```

### 6.6.6. Worked Example - Climate and Forecasting Conventions

Variable to variable referencing is a pattern used extensively within the Climate and Forecasting conventions for netCDF files (CF-netCDF).

CF-netCDF attributes such as `coordinates` can be defined within a specification graph as variable reference attributes, i.e.:

```

cf:coordinates a owl:ObjectProperty ;
    rdfs:range bald:Resource .

```

Given a specification graph definition and an input CF-netCDF file:

```

netcdf orca2_votemper {
dimensions:
    dim0 = 148 ;
    dim1 = 180 ;
    time = 4 ;
    deptht = 4 ;

variables:
    float votemper(time, deptht, dim0, dim1) ;
        votemper:_FillValue = 9.96921e+36f ;
        votemper:standard_name = "sea_water_potential_temperature" ;
        votemper:long_name = "Temperature" ;
        votemper:units = "degC" ;
        votemper:coordinates = "deptht nav_lat nav_lon time" ;
    float deptht(deptht) ;
        deptht:units = "m" ;
        deptht:standard_name = "depth" ;
        deptht:long_name = "Vertical T levels" ;
        deptht:positive = "down" ;
        deptht:title = "deptht" ;
    float nav_lat(dim0, dim1) ;
        nav_lat:units = "degrees" ;
        nav_lat:standard_name = "latitude" ;
        nav_lat:long_name = "Latitude" ;
        nav_lat:nav_model = "Default grid" ;
    float nav_lon(dim0, dim1) ;
        nav_lon:units = "degrees" ;
        nav_lon:standard_name = "longitude" ;
        nav_lon:long_name = "Longitude" ;
        nav_lon:nav_model = "Default grid" ;
    int time(time) ;
        time:units = "hours since 2001-01-01 00:00:00" ;
        time:standard_name = "time" ;
        time:long_name = "Time axis" ;
        time:calendar = "360_day" ;
        time:time_origin = "2001-JAN-01 00:00:00" ;
        time:title = "Time" ;
// global attributes:
    :Conventions = "CF-1.5" ;

```

then the `coordinates` attribute defined on the `votemper` variable is interpreted as a variable to variable reference, creating four explicit reference statements with blank nodes representing the bald:Reference instances. These explicit bald:references statements are included within the graph alongside the CFTerms:coordinates statements.

This can be seen within the interpreted metadata graph for this file

```

@prefix CFTerms: <http://def.scitools.org.uk/CFTerms/> .
@prefix NetCDF: <http://def.scitools.org.uk/NetCDF/> .

```

```

@prefix bald: <https://www.opengis.net/def/binary-array-ld/> .
@prefix cf_sname: <http://vocab.nerc.ac.uk/standard_name/> .
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix this: <file://CDL/minVotemper.cdl/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

this: a bald:Container ;
  NetCDF:Conventions "CF-1.5" ;
  dct:format [ a dct:MediaType ;
    dct:identifier <http://vocab.nerc.ac.uk/collection/M01/current/NC/> ] ;
  dcat:distribution [ a dcat:Distribution ;
    dcat:mediaType [ a dct:MediaType ;
      dct:identifier "application/netcdf" ] ] ;
  bald:contains this:deptht,
    this:nav_lat,
    this:nav_lon,
    this:time,
    this:votemper .

this:votemper a bald:Array ;
  CFTerms:coordinates this:deptht,
    this:nav_lat,
    this:nav_lon,
    this:time ;
  CFTerms:standard_name "sea_water_potential_temperature" ;
  NetCDF:FillValue "9.96921e+36" ;
  NetCDF:long_name "Temperature" ;
  NetCDF:units "degC" ;
  bald:references [ a bald:Reference ;
    bald:target this:nav_lat ;
    bald:targetRefshape ( 1 1 148 180 ) ],
  [ a bald:Reference ;
    bald:target this:nav_lon ;
    bald:targetRefshape ( 1 1 148 180 ) ],
  [ a bald:Reference ;
    bald:target this:deptht ;
    bald:targetRefshape ( 1 4 1 1 ) ],
  [ a bald:Reference ;
    bald:target this:time ;
    bald:targetRefshape ( 4 1 1 1 ) ] ;
  bald:shape ( 4 4 148 180 ) .

this:deptht a bald:Array ;
  CFTerms:positive "down" ;
  CFTerms:standard_name "depth" ;
  NetCDF:long_name "Vertical T levels" ;
  NetCDF:title "deptht" ;

```

```

NetCDF:units "m" ;
bald:shape ( 4 ) .

this:nav_lat a bald:Array ;
  this:nav_model "Default grid" ;
  CFTerms:standard_name "latitude" ;
  NetCDF:long_name "Latitude" ;
  NetCDF:units "degrees" ;
  bald:shape ( 148 180 ) .

this:nav_lon a bald:Array ;
  this:nav_model "Default grid" ;
  CFTerms:standard_name "longitude" ;
  NetCDF:long_name "Longitude" ;
  NetCDF:units "degrees" ;
  bald:shape ( 148 180 ) .

this:time a bald:Array ;
  this:time_origin "2001-JAN-01 00:00:00" ;
  CFTerms:calendar "360_day" ;
  CFTerms:standard_name "time" ;
  NetCDF:long_name "Time axis" ;
  NetCDF:title "Time" ;
  NetCDF:units "hours since 2001-01-01 00:00:00" ;
  bald:shape ( 4 ) .

```

## 6.7. NetCDF Linked Data on Schema.org

This standard recognises that while domain specific vocabularies and detailed RDF representations of datasets in NetCDF are important, there is a growing movement to make use of standardised vocabularies, such as Schema.org, to harmonise the higher-level descriptions of digital assets such as datasets. The goal of this activity is often the promotion of datasets in portals through Search Engine Optimisation and also to provide a common syntax and understanding of the datasets in a consistent language.

To this end, NetCDF-LD also supports a limited description of a netCDF file as Schema.org Dataset through mapping a bald:Container to the Schema.org Dataset class.

Within the Schema.org representation, the DCAT usage to describe the file type and to provide an optional statement to specify the resolvable location of the file object described above is mapped to Schema.org's DataDownload class and its associated properties (Figure 12).

*A sample NetCDF-LD graph cast to the Schema.org vocabulary*

image::netcdf\_ld\_as\_schema\_dot\_org.png

A worked example follows in RDF, encoded using Turtle, the terse triple language (TTL):

```

@prefix so: <http://schema.org/> .

<https://www.ngdc.noaa.gov/thredds/dodsC/arctic/Polar-APP-
X_v01r01_Nhem_1400_d20160801_c20160803.nc> a so:Dataset;
  so:name "Extended AVHRR Polar Pathfinder Fundamental Climate Data Record (APPx
CDR)" ;
  so:description "The Extended AVHRR Polar Pathfinder (APP-x) version-2 Thematic
Climate Data Record (CDR) includes surface temperature, surface albedo, surface and
the Top Of the Atmosphere (TOA) shortwave and longwave radiative fluxes, cloud
properties (amount, phase, particle size, optical depth, top pressure and temperature,
surface and TOA radiative effect), and ice thickness and age. The APP-x CDR has twice
daily data at local solar times of 14 and 04(02) for the Arctic(Antarctic) at a
spatial resolution of 25 km for both poles." ;
  so:identifier "Polar-APP-X_v01r01_Nhem_1400_d20160801_c20160803.nc" ;
  so:keywords:
    "EARTH SCIENCE > ATMOSPHERE > ATMOSPHERIC RADIATION > RADIATIVE FLUX",
    "EARTH SCIENCE > TERRESTRIAL HYDROSPHERE > SNOW and ICE > ALBEDO" ;
  so:license "No restrictions on access or use" ;
  so:url <https://www.ngdc.noaa.gov/thredds/dodsC/arctic/Polar-APP-
X_v01r01_Nhem_1400_d20160801_c20160803.nc/>;
  so:distribution [
    a so:DataDownload ;
    so:contentUrl <https://www.ngdc.noaa.gov/thredds/dodsC/arctic/Polar-APP-
X_v01r01_Nhem_1400_d20160801_c20160803.nc/> ;
    so:encodingFormat "application/netcdf",
    "http://vocab.nerc.ac.uk/collection/M01/current/NC/"
  ] .

```

## 6.8. Implementation Adaptation

This standard recognises that there are myriad opportunities for optimisation of large collections of graphs of netCDF files contents.

For example, a series, or collection of netCDF files may share the vast majority of metadata, differing in only one or two key elements, such as a different value for a temporal coordinate, along with different data values.

Implementations may implement many different strategies, reducing the volume of data stored, constructing shared entities to aid query construction or optimisation.

As this standard is targeted at standardising the representation of a single file, it is important that a method for validating consistency is provided which does not preclude useful optimisation steps.

On this basis, it is suggested that an implementation may provide a transformation, such that the graph for an individual file may be derived from a compressed or engineered storage format.

The data storage shall be deemed compliant with this standard if a graph representing a single file may be transformed by the provided transformation into a single file graph that meets the validation rules within this standard.



The format for such transformations is not specified by this standard, it need only provide a suitable RDF graph output.

# Chapter 7. Clause containing normative material

## 7.1. Requirement Class A: URIs

Requirements for Universal Resource Identifiers definition and interpretation.

Requirements Class	
<a href="http://www.opengis.net/spec/netcdf-ld/1.0/req/req-class-a">http://www.opengis.net/spec/netcdf-ld/1.0/req/req-class-a</a>	
Target type	Core

### 7.1.1. Requirement A-1

<b>Requirement A-1</b>	<p><code>/req/req-class-a/requirement-1</code></p> <p>A netCDF file content root group shall be identified by an identity URI, external to the file contents.</p> <p>The identity of the root group shall always terminate in a '/' character.</p> <p>By convention, the prefix 'this:' for the identity URI may be used by implementations representing a single netCDF file within a graph.</p> <p>Implementations shall provide an optional 'uri' option to specify the identity of the file contents. This identity URI shall conform to the W3 URI specification (<a href="https://datatracker.ietf.org/doc/html/rfc3986">https://datatracker.ietf.org/doc/html/rfc3986</a>). Note, this includes 'http:', 'https:', 'file:', 'urn:' URIs.</p> <p>Implementations shall provide an optional 'download-url' option. If no 'uri' is provided, but a 'download-url' is provided, then this shall be used as the file contents identity, ensuring that a trailing '/' is added if it does not exist in the 'download-url'.</p> <p>If no identity is provided, then the file path (<b>file:</b>) or remote location (<b>http:</b> / <b>https:</b>) used by a library to access the file is used as a default identity. If used, a local file URI will always use the '/' forward slash as an element separator, even on systems where local identifiers use back slash separators.</p>
------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 7.1.2. Requirement A-2

### Requirement A-2

/req/req-class-a/requirement-2

The identified netCDF root group shall be described as a dcat:Distribution

```
this: a bald:Container ;
  dcat:distribution [
    a dcat:Distribution;
    dcat:mediaType [
      a dct:MediaType;
      dct:identifier "application/netcdf"
    ];
    dct:format [
      a dct:MediaType;
      dct:identifier
<http://vocab.nerc.ac.uk/collection/M01/current/NC/>
    ]
  ].
```

Implementations shall provide an optional 'download-url' option, providing the ability for a user to provide an explicit downloadURL for the file at run time.

If provided, then the 'download-url' value shall be included in the graph, with respect to the identified root group's **dcat:distribution** blank node:

```
this: a bald:Container ;
  dcat:distribution [
    a dcat:Distribution;
    dcat:downloadURL <{download-url}>;
    dcat:mediaType [
      a dct:MediaType;
      dct:identifier "application/netcdf"
    ];
    dct:format [
      a dct:MediaType;
      dct:identifier
<http://vocab.nerc.ac.uk/collection/M01/current/NC/>
    ]
  ].
```

### 7.1.3. Requirement A-3

<b>Requirement A-3</b>	<p>/req/req-class-a/requirement-3</p> <p>Each netCDF variable within the root group, each netCDF group and each netCDF variable within each group shall take its identity as a URI, relative to the file identity.</p> <p>This identity is defined using the full variable name, including any owning groups, appended to the file identity. All element separation uses the / character.</p>
------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 7.1.4. Requirement A-4

<b>Requirement A-4</b>	<p>/req/req-class-a/requirement-4</p> <p>Each netCDF variable shall declare a type statement, and RDF triple of the form</p> <pre>&lt;variableURI&gt; rdf:type &lt;type&gt;</pre> <p>There are two recognised type identifiers within the standard.</p> <p>If the variable contains an array data payload (which may be missing data) then the type shall be defined as bald:Array</p> <pre>&lt;variableURI&gt; rdf:type bald:Array .</pre> <p>If there is no array payload and the variable is single valued (which may be missing data) then the type shall be defined as bald:Resource.</p> <pre>&lt;variableURI&gt; rdf:type bald:Resource .</pre>
------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 7.1.5. Requirement A-5

<p><b>Requirement A-5</b></p>	<p>/req/req-class-a/requirement-5</p> <p>Each netCDF variable that is a bald:Array shall declare a shape statement.</p> <p>A bald:array within the metadata graph shall include a statement defining the shape of the data payload array as a Literal encoded tuple of integer values, using bald:shape. Integers are derived from references to named dimensions within the netCDF input file.</p> <p>For example:</p> <pre data-bbox="437 651 1315 786" style="border: 1px solid #ccc; padding: 10px;"> &lt;variableURI&gt; rdf:type bald:Array ;                 bald shape '(1,1,16,9)' . </pre> <p>If the bald:Array instance has a single dimension, then the first and last values from the data payload, as ordered within the netCDF file, shall be encoded as Literals within the metadata graph, using bald:firstValue and bald:lastValue.</p> <p>For example:</p> <pre data-bbox="437 1093 1315 1308" style="border: 1px solid #ccc; padding: 10px;"> &lt;variableURI&gt; rdf:type bald:Array ;                 bald shape '(72,)' ;                 bald:firstValue -90 ;                 bald:lastValue 90 . </pre>
-------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 7.2. Requirement Class B: Prefixing

Requirements for Prefix definition and interpretation.

Requirements Class	
<a href="http://www.opengis.net/spec/netcdf-ld/1.0/req/req-class-b">http://www.opengis.net/spec/netcdf-ld/1.0/req/req-class-b</a>	
Target type	Prefixing

### 7.2.1. Requirement B-1

**Requirement B-1**

/req/req-class-b/requirement-1

A list of prefixes, defined within a netCDF file, within a single variable or group that is referenced by the global root group attribute:

`bald__isPrefixedBy`

shall be interpreted as Prefixes and used to interpret prefixed contents within the file.

If a prefix variable or group is defined, that prefix variable shall contain the prefix:

`bald__ = https://www.opengis.net/def/binary-array-ld/`

Prefixes defined within a prefix variable shall be used as prefixes for the parsing of that file.

The definition of multiple prefix resources within a single file is invalid in this standard. Implementations may choose to combine prefix collections in invalid cases, but no precedence is implied, and prefix conflict is unresolved.

**7.2.2. Requirement B-2****Requirement B-2**

/req/req-class-b/requirement-2

A prefix attribute name, defined in a prefix variable, shall use the limited character set:

`[a-zA-z0-9_]`

A prefix attribute name, defined in a prefix variable, shall end with the double-underscore sub-string:

--

**7.2.3. Requirement B-3**

**Requirement B-3**

/req/req-class-b/requirement-3

A prefix value, defined in a prefix variable, shall be a HTTP URI or a HTTPS URI.

A prefix value, defined in a prefix variable, shall end in either a / or a # (a valid URI separator).

#### 7.2.4. Requirement B-4

**Requirement B-4**

/req/req-class-b/requirement-4

Prefixes shall be interpreted from provided JSON-LD context files at run time.

#### 7.2.5. Requirement B-5

**Requirement B-5**

/req/req-class-b/requirement-5

A prefix attribute name, defined in a JSON-LD context file, shall use the limited character set:

[a-zA-z0-9\_]

A prefix attribute name, defined in a JSON-LD context file, shall have the double-underscore sub-string:

--

appended to the provided string.

#### 7.2.6. Requirement B-6

**Requirement B-6**

/req/req-class-b/requirement-6

A prefix value, defined in a JSON-LD context file, shall be a HTTP URI or a HTTPS URI.

A prefix value, defined in a JSON-LD context file, shall end in either a / or a # (a valid URI separator).

### 7.2.7. Requirement B-7

**Requirement B-7**

/req/req-class-b/requirement-7

If a prefix string is defined multiple times in provided JSON-LD context files with different URI interpretations, then implementations shall ignore that prefix and treat the prefix as locally unresolved.

### 7.2.8. Requirement B-8

**Requirement B-8**

/req/req-class-b/requirement-8

Prefix definitions provided explicitly within a file shall not be overwritten by context files. Prefixes defined within a file have precedence.

If a prefix defined within a file is also defined within provided JSON-LD context files with different URI interpretations, then implementations shall ignore that JSON-LD context definition and treat the prefix as locally resolved.

This standard does not consider this as a parsing error.

Implementations may treat this condition as a warning condition and as a validation error within the provided context.

## 7.3. Requirement Class C: Aliasing

Requirements for Alias definition and interpretation.

**Requirements Class**



Target type	Aliasing
-------------	----------

### 7.3.1. Requirement C-1

<b>Requirement C-1</b>	<p>/req/req-class-c/requirement-1</p> <p>Aliases shall be defined as one or more RDF graphs.</p> <p>Implementations may choose to provide prefixes for provided alias graphs, to aid readability of output content metadata graphs.</p>
------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 7.3.2. Requirement C-2

<b>Requirement C-2</b>	<p>/req/req-class-c/requirement-2</p> <p>An alias, defined in an alias graph, shall declare the statement</p> <pre>&lt;\$entity&gt; &lt;http://purl.org/dc/terms/identifier&gt; "\$Literal" .</pre> <p>The defined "<b>\$Literal</b>" shall be used as the exact string to match on when interpreting aliases.</p> <p>If multiple exact string matches exist across the provided alias graphs, then this is an error condition, no preference is implied.</p>
------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 7.3.3. Requirement C-3

<b>Requirement C-3</b>	<p>/req/req-class-c/requirement-3</p> <p>An attribute alias, defined in an alias graph, shall declare the statement</p> <pre>&lt;\$entity&gt; &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#type&gt; &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#Property&gt; .</pre> <p>or</p> <pre>&lt;\$entity&gt; &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#type&gt; &lt;http://www.w3.org/2002/07/owl#ObjectProperty&gt; .</pre>
------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 7.4. Requirement Class D: Attribute names

Requirements for Attribute name definition and interpretation.

Requirements Class	
<a href="http://www.opengis.net/spec/netcdf-ld/1.0/req/req-class-c">http://www.opengis.net/spec/netcdf-ld/1.0/req/req-class-c</a>	
Target type	Attribute names

### 7.4.1. Requirement D-1

<b>Requirement D-1</b>	/req/req-class-d/requirement-1
	All attribute names in a netCDF file shall be interpreted as URIs.

### 7.4.2. Requirement D-2

<b>Requirement D-2</b>	/req/req-class-d/requirement-2
	If an attribute name in a netCDF file matches a prefix definition, an attribute name in the prefix variable of that file, then that attribute name shall be interpreted using the defined prefix.

### 7.4.3. Requirement D-3

<b>Requirement D-3</b>	/req/req-class-d/requirement-3
	If an attribute name in a netCDF file does not match a prefix definition, but does match an attribute name alias, then that attribute name shall be interpreted using the defined alias.

### 7.4.4. Requirement D-4

<b>Requirement D-4</b>	/req/req-class-d/requirement-4
	If an attribute name in a netCDF file does not match a prefix definition, or attribute alias, then that attribute name shall be interpreted locally, as a URI relative to the file identity.

## 7.5. Requirement Class E: Variable-to-variable referencing

Requirements for variable-to-variable definition and interpretation.

Requirements Class	
<a href="http://www.opengis.net/spec/netcdf-ld/1.0/req/req-class-e">http://www.opengis.net/spec/netcdf-ld/1.0/req/req-class-e</a>	
Target type	Variable-to-variable referencing

### 7.5.1. Requirement E-1

<b>Requirement E-1</b>	<p>/req/req-class-e/requirement-1</p> <p>An attribute may be declared by a metadata standard as a variable reference attribute.</p> <p>In order to do this, the attribute shall declare itself in a resolveable RDF graph and include the statement:</p> <pre>&lt;{attribute}&gt; rdfs:range &lt;https://www.opengis.net/def/binary-array-ld/Resource&gt; .</pre> <p>It is also permissible to provide an rdfs:range statement targeting another resource and for that resource to declare that it is an rdfs:subClassOf bald:Resource, for example:</p> <pre>&lt;{attribute}&gt; rdfs:range &lt;{AClass}&gt; .</pre> <pre>&lt;{AClass}&gt; rdfs:subClassOf &lt;https://www.opengis.net/def/binary-array-ld/Resource&gt; .</pre>
------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 7.5.2. Requirement E-2

**Requirement E-2**

/req/req-class-e/requirement-2

A 'variable to variable reference' is established when an attribute, declared as a variable reference attribute (Requirement E-1) meets one of the following conditions.

An attribute declared by a metadata standard as a variable reference attribute where: - the attribute value matches the name of a variable within the file; shall be interpreted as a reference to that other variable.

An attribute declared by a metadata standard as a variable reference attribute where: - the attribute value is a set of space separated names, all of which match a variable name within the file; shall be interpreted as an unordered set of references to the other variables.

An attribute declared by a metadata standard as a variable reference attribute where: - the attribute is a parenthesis wrapped, space-separated, list of names, all of which match a variable name within the file; shall be interpreted as an ordered list of references to the other variables.

**7.5.3. Requirement E-3**

**Requirement E-3**

/req/req-class-e/requirement-3

Each variable within a netCDF file that is defined with respect to one or more dimensions shall be described by a statement:

```
<$entity> <https://www.opengis.net/def/binary-array-ld/shape>  
($d0 $d1 $d2) .
```

where  $d_n$  is an integer, taken from the defined dimension size in the netCDF file and the count of the number of values is the dimensionality of the variable.

Where a variable is defined with respect to a single dimension, the statements

```
<$entity> <https://www.opengis.net/def/binary-array-ld/shape>  
($d0) .
```

and

```
<$entity> <https://www.opengis.net/def/binary-array-ld/shape>  
$d0 .
```

shall be interpreted as semantically equivalent.

**7.5.4. Requirement E-4**

## Requirement E-4

/req/req-class-e/requirement-4

A variable to variable reference is defined in Requirement E-2

Each variable to variable reference between two variables, both of which are defined with respect to one or more dimensions, shall be described by the statement.

```
<${sourceEntity}> <https://www.opengis.net/def/binary-array-ld/references> <${referenceEntity}> .
```

and the definition of a <\${referenceEntity}>, an instance of <<https://www.opengis.net/def/binary-array-ld/Reference>>. This entity carries the broadcast relationship details and the reference to the target variable.

```
<${referenceEntity}> a <https://www.opengis.net/def/binary-array-ld/Reference> ;
```

```
OPTIONAL(<https://www.opengis.net/def/binary-array-ld/sourceRefShape> ($d1 $d2 $d3) ;)
```

```
<https://www.opengis.net/def/binary-array-ld/targetRefShape> ($d4 $d5 $d6) ;
```

```
<https://www.opengis.net/def/binary-array-ld/target>  
<${targetEntity}> .
```

$d_n$  are all defined to be integers.

The object of the <https://www.opengis.net/def/binary-array-ld/sourceRefShape> and <https://www.opengis.net/def/binary-array-ld/targetRefShape> statements are each an RDF list of integer elements.

Each integer element shall be taken directly from the named dimension size defined within the input netCDF file.

If the sourceRefShape is not provided, this is inferred to be the same as the defined `bald:shape`.

The lengths of these two RefShape lists shall be the same.

The product of the sourceRefShape (if provided) elements shall match the product of the `bald:shape`, the number of elements shall be the same.

### 7.5.5. Requirement E-5

#### Requirement E-5

/req/req-class-e/requirement-5

Each variable that is not itself a netCDF Coordinate Variable, that is defined with respect to a netCDF Coordinate Variable, shall be interpreted as a variable to variable reference from that variable to the netCDF Coordinate variable.

A netCDF Coordinate Variable is defined within the netCDF user guide as a one-dimensional variable with the same variable name as the only netCDF dimension with respect to which it is defined.

### 7.5.6. Requirement E-6

#### Requirement E-6

/req/req-class-e/requirement-6

Where any ambiguity in ordering exists in the netCDF file for explicitly defined sourceRefShape and targetRefShape instances, due to partial overlap of named dimensions for both netCDF variables, then source reference dimensions shall be ordered before target reference dimensions.

### 7.5.7. Requirement E-7

#### Requirement E-7

/req/req-class-e/requirement-7

A specific reference shall be made in the output graph to capture the containment relationship between netCDF groups and variables.

Each netCDF group in the metadata graph shall declare a statement for each variable that it contains.

----

```
{groupURI} bald:contains {variableURI}
```

----

## 7.5.8. Requirement E-8

<b>Requirement E-8</b>	<p>/req/req-class-e/requirement-8</p> <p>The BALD vocabulary elements:</p> <p>---- bald:isPrefixedBy bald:isAliasedBy ----</p> <p>are exceptions within the standard.</p> <p>They are variable to variable reference definitions, but they provide capability within this standard, they are not metadata content for the file metadata summary.</p> <p>---- bald:isPrefixedBy bald:isAliasedBy ----</p> <p>statements shall not be represented within the standard output metadata graph. Target variables and groups from these variable to variable references shall be omitted from the output metadata graph.</p>
------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 7.6. Requirement Class F: NetCDF Coordinate Variables

Requirements for netCDF Coordinate Variables.

### 7.6.1. Requirement F-1

<b>Requirement F-1</b>	<p>/req/req-class-f/requirement-1</p> <p>NetCDF Coordinate Variables, which are limited to one-dimensional variables of data, shall be interpreted as cases of variable to variable referencing and produce</p> <p><b>bald:references</b></p> <p>statements within created graphs.</p>
------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 7.6.2. Requirement F-2



**Requirement F-2**

/req/req-class-f/requirement-2

Information on the NetCDF Coordinate Variables' data payload, which are limited to one-dimensional variables of data, shall be included in created graphs.

The first value of the coordinate variable data payload shall be included in the created graph, using the predicate:

`bald:arrayFirstValue`

Optionally the last value, if there is more than one value, of the coordinate variable data payload, shall be included in the created graph, using the predicate:

`bald:arrayLastValue`

The first and last positions are taken from the order of the coordinate variable array values, as presented within the file.

If the value is encoded as a missing value, then no statement is included within the metadata graph.

# Chapter 8. Media Types for any data encoding(s)

The media type for output encodings for metadata graphs interpreted by this standard shall conform to one of the W3 media type choices:

- `text/turtle` <https://www.iana.org/assignments/media-types/text/turtle>
- `application/rdf+xml` <https://www.iana.org/assignments/media-types/application/rdf+xml>
- `application/ld+json` <https://www.iana.org/assignments/media-types/application/ld+json>

The media type definition for defining the netCDF encoding is currently detailed in the specification as:

- `application/netcdf`

A new media Type is being targeted for registration with IANA:

- `application/netcdf`

# Annex A: Conformance Class Abstract Test Suite (Normative)

## A.1. Conformance Class A: URIs

Conformance class for Universal Resource Identifiers definition and interpretation.

### A.1.1. Abstract Test for Requirements A-1, A-2, A-3, A-4, A-5

<b>Test id:</b>	/conf/conf-class-a/test-1
<b>Requirement:</b>	/req/req-class-a/requirement-1, /req/req-class-a/requirement-2, /req/req-class-a/requirement-3, /req/req-class-a/requirement-4, /req/req-class-a/requirement-5
<b>Test purpose:</b>	Verify that the identities of the file entities and their descriptors are present within the RDF graph parsed from that file.
<b>Test method:</b>	Parse the input netCDF file, created from the provided CDL, into an RDF Graph, given the provided identity. Verify that the output RDF graph conforms to the expected result.

Identity:

```
http://secret.binary-array-ld.net/identity.nc
```

Input NetCDF File - Defined in CDL: [https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract\\_tests/CDL/ogcClassA.cdl](https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract_tests/CDL/ogcClassA.cdl)

```
netcdf ogcClassA {
  dimensions:
    d0 = 1 ;
    d1 = 1 ;
  variables:
    int var0 ;
    int var1 ;

  data:
}
```

Validated output RDF graph: [https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract\\_tests/TTL/ogcClassA.ttl](https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract_tests/TTL/ogcClassA.ttl)

```

@prefix bald: <https://www.opengis.net/def/binary-array-ld/> .
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix this: <http://secret.binary-array-ld.net/identity.nc/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

this: a bald:Container ;
    dct:format [ a dct:MediaType ;
        dct:identifier <http://vocab.nerc.ac.uk/collection/M01/current/NC/> ] ;
    dcat:distribution [ a dcat:Distribution ;
        dcat:mediaType [ a dct:MediaType ;
            dct:identifier "application/netcdf" ] ] ;
    bald:contains this:var0,
        this:var1 .

this:var0 a bald:Resource .

this:var1 a bald:Resource .

```

## A.2. Conformance Class B: Prefixing

### A.2.1. Abstract Test for Requirement B-1, B-2, B-3, B-4, B-5, B-6, B-7, B-8

<b>Test id:</b>	/conf/conf-class-b/test-1
<b>Requirement:</b>	/req/req-class-b/requirement-1, /req/req-class-b/requirement-2, /req/req-class-b/requirement-3, /req/req-class-b/requirement-4, /req/req-class-b/requirement-5, /req/req-class-b/requirement-6, /req/req-class-b/requirement-7, /req/req-class-b/requirement-8
<b>Test purpose:</b>	Verify that defined prefixes within a file are recognised and used to expand prefix notation attribute names and values.
<b>Test method:</b>	Parse the input netCDF file, created from the provided CDL, into an RDF Graph, given the provided identity. Verify that the output RDF graph conforms to the expected result.

Identity:

```
http://secret.binary-array-ld.net/prefix.nc
```

Input NetCDF File - Defined in CDL: [https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract\\_tests/CDL/ogcClassB.cdl](https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract_tests/CDL/ogcClassB.cdl)

```

netcdf ogcClassB {
dimensions:
    d0 = 1 ;
    d1 = 1 ;
variables:
    int var0 ;
    var0:rdfs__label = "Variable Zero" ;
    int var1 ;
    var1:rdfs__label = "Variable One" ;
    int prefix_list ;
    prefix_list:bald__ = "https://www.opengis.net/def/binary-array-ld/" ;
    prefix_list:rdf__ = "http://www.w3.org/1999/02/22-rdf-syntax-ns#" ;
    prefix_list:rdfs__ = "http://www.w3.org/2000/01/rdf-schema#" ;
    :bald__isPrefixedBy = "prefix_list" ;
data:
}

```

Validated output RDF graph: [https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract\\_tests/TTL/ogcClassB.ttl](https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract_tests/TTL/ogcClassB.ttl)

```

@prefix bald: <https://www.opengis.net/def/binary-array-ld/> .
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix this: <http://secret.binary-array-ld.net/prefix.nc/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

this: a bald:Container ;
    dct:format [ a dct:MediaType ;
        dct:identifier <http://vocab.nerc.ac.uk/collection/M01/current/NC/> ] ;
    dcat:distribution [ a dcat:Distribution ;
        dcat:mediaType [ a dct:MediaType ;
            dct:identifier "application/netcdf" ] ] ;
    bald:contains this:var0,
        this:var1 .

this:var0 a bald:Resource ;
    rdfs:label "Variable Zero" .

this:var1 a bald:Resource ;
    rdfs:label "Variable One" .

```

### A.3. Conformance Class C: Aliasing

### A.3.1. Abstract Test for Requirement C-1, C-2, C-3

<b>Test id:</b>	/conf/conf-class-c/test-1
<b>Requirement:</b>	/req/req-class-c/requirement-1, /req/req-class-c/requirement-2, /req/req-class-c/requirement-3
<b>Test purpose:</b>	Verify that defined aliases within a file are recognised and used to expand attribute names and values.
<b>Test method:</b>	Parse the input netCDF file, created from the provided CDL, into an RDF Graph, given the provided identity and provided alias graph. Verify that the output RDF graph conforms to the expected result.

Identity:

```
http://secret.binary-array-ld.net/alias.nc
```

Alias Graph

```
https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract_tests/aliases/NetCDF.ttl
```

Input NetCDF File - Defined in CDL: [https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract\\_tests/CDL/ogcClassC.cdl](https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract_tests/CDL/ogcClassC.cdl)

```
netcdf ogcClassC {
  dimensions:
    d0 = 1 ;
    d1 = 1 ;
  variables:
    int var0 ;
    int var1 ;
  :title = "Sample netCDF file definition with alias terms from the netCDF user guide" ;
  data:
}
```

Validated output RDF graph: [https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract\\_tests/TTL/ogcClassC.ttl](https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract_tests/TTL/ogcClassC.ttl)

```

@prefix NetCDF: <http://def.scitools.org.uk/NetCDF/> .
@prefix bald: <https://www.opengis.net/def/binary-array-ld/> .
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix this: <http://secret.binary-array-ld.net/alias.nc/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

this: a bald:Container ;
  NetCDF:title "Sample netCDF file definition with alias terms from the netCDF user
guide" ;
  dct:format [ a dct:MediaType ;
    dct:identifier <http://vocab.nerc.ac.uk/collection/M01/current/NC/> ] ;
  dcat:distribution [ a dcat:Distribution ;
    dcat:mediaType [ a dct:MediaType ;
      dct:identifier "application/netcdf" ] ] ;
  bald:contains this:var0,
    this:var1 .

this:var0 a bald:Resource .

this:var1 a bald:Resource .

x

```

## A.4. Conformance Class D: Attribute Names

### A.4.1. Abstract Test for Requirement D-1, D-2, D-3, D-4

<b>Test id:</b>	/conf/conf-class-d/test-1
<b>Requirement:</b>	/req/req-class-d/requirement-1, /req/req-class-d/requirement-2, /req/req-class-d/requirement-3, /req/req-class-d/requirement-4
<b>Test purpose:</b>	Verify that attribute names within a file are represented as URIs in the output RDF graph and that attribute naming rules are conformed to.
<b>Test method:</b>	Parse the input netCDF file, created from the provided CDL, into an RDF Graph, given the provided identity and provided alias graph. Verify that the output RDF graph conforms to the expected result.

Identity:

```
http://secret.binary-array-ld.net/attributes.nc
```

Alias Graph

[https://raw.githubusercontent.com/opegeospatial/netcdf-ld/v0.2/standard/abstract\\_tests/aliases/NetCDF.ttl](https://raw.githubusercontent.com/opegeospatial/netcdf-ld/v0.2/standard/abstract_tests/aliases/NetCDF.ttl)

Input NetCDF File - Defined in CDL: [https://raw.githubusercontent.com/opegeospatial/netcdf-ld/v0.2/standard/abstract\\_tests/CDL/ogcClassD.cdl](https://raw.githubusercontent.com/opegeospatial/netcdf-ld/v0.2/standard/abstract_tests/CDL/ogcClassD.cdl)

```
netcdf ogcClassD {
dimensions:
    d0 = 1 ;
    d1 = 1 ;
variables:
    int var0 ;
        var0:rdfs__label = "Variable Zero" ;
    int var1 ;
        var1:rdfs__label = "Variable One" ;
    int prefix_list ;
        prefix_list:bald__ = "https://www.opengis.net/def/binary-array-ld/" ;
        prefix_list:rdf__ = "http://www.w3.org/1999/02/22-rdf-syntax-ns#" ;
        prefix_list:rdfs__ = "http://www.w3.org/2000/01/rdf-schema#" ;
        :bald__isPrefixedBy = "prefix_list" ;
        :title = "Sample netCDF file definition with alias terms from the netCDF user
guide" ;
        :localtitle = "OGC netCDF-LD sample file" ;
data:
}
```

Validated output RDF graph: [https://raw.githubusercontent.com/opegeospatial/netcdf-ld/v0.2/standard/abstract\\_tests/TTL/ogcClassD.ttl](https://raw.githubusercontent.com/opegeospatial/netcdf-ld/v0.2/standard/abstract_tests/TTL/ogcClassD.ttl)



```

@prefix NetCDF: <http://def.scitools.org.uk/NetCDF/> .
@prefix bald: <https://www.opengis.net/def/binary-array-ld/> .
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix this: <http://secret.binary-array-ld.net/attributes.nc/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

this: a bald:Container ;
  NetCDF:title "Sample netCDF file definition with alias terms from the netCDF user
guide" ;
  this:localtitle "OGC netCDF-LD sample file" ;
  dct:format [ a dct:MediaType ;
    dct:identifier <http://vocab.nerc.ac.uk/collection/M01/current/NC/> ] ;
  dcat:distribution [ a dcat:Distribution ;
    dcat:mediaType [ a dct:MediaType ;
      dct:identifier "application/netcdf" ] ] ;
  bald:contains this:var0,
    this:var1 ;
  bald:isPrefixedBy "prefix_list" .

this:var0 a bald:Resource ;
  rdfs:label "Variable Zero" .

this:var1 a bald:Resource ;
  rdfs:label "Variable One" .

```

## A.5. Conformance Class E: Variable-to-variable Referencing; Class F: Coordinate Variables

### A.5.1. Abstract Test for Requirement E-1, E-2, E-3, E-4, E-5, E-6, E-7, E-8, F-1, F-2

<b>Test id:</b>	/conf/conf-class-ef/test-1
<b>Requirement:</b>	/req/req-class-e/requirement-1, /req/req-class-e/requirement-2, /req/req-class-e/requirement-3, /req/req-class-e/requirement-4, /req/req-class-e/requirement-5, /req/req-class-e/requirement-6, /req/req-class-e/requirement-7, /req/req-class-e/requirement-8, /req/req-class-f/requirement-1, /req/req-class-f/requirement-2
<b>Test purpose:</b>	Verify that variable to variable References and coordinate variables within a file are recognised and represented as references in the output RDF graph.

<b>Test method:</b>	Parse the input netCDF file, created from the provided CDL, into an RDF Graph, given the provided identity. Verify that the output RDF graph conforms to the expected result.
---------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Identity:

`http://secret.binary-array-ld.net/reference.nc`

Input NetCDF File - Defined in CDL: [https://raw.githubusercontent.com/opegeospatial/netcdf-ld/v0.2/standard/abstract\\_tests/CDL/ogcClassEF.cdl](https://raw.githubusercontent.com/opegeospatial/netcdf-ld/v0.2/standard/abstract_tests/CDL/ogcClassEF.cdl)

```

netcdf tmpMwXy8U {
dimensions:
    pdim0 = 11 ;
    pdim1 = 17 ;
    xy = 2 ;
variables:
    int data_variable1(pdim0, pdim1) ;
    data_variable1:bald__references = "location_variable" ;
    data_variable1:long_name = "data variable one";

    int data_variable2(pdim0, pdim1) ;
    data_variable2:bald__references = "location_variable" ;
    data_variable2:long_name = "data variable two";

    int pdim0(pdim0) ;

    int pdim1(pdim1) ;

    int location_variable(pdim0, pdim1, xy) ;
    location_variable:bald__references = "location_reference_system" ;

    int location_reference_system;
    location_reference_system:pcode = "4897";

    int set_collection ;
    set_collection:bald__references = "data_variable1 data_variable2" ;

    int list_collection ;
    list_collection:bald__references = "( data_variable1 data_variable2 )" ;

// prefix group
group: prefix_list {
    :bald__ = "https://www.opengis.net/def/binary-array-ld/" ;
    :rdf__ = "http://www.w3.org/1999/02/22-rdf-syntax-ns#" ;
}
// global attributes:
    :bald__isPrefixedBy = "prefix_list" ;
}

```

Validated output RDF graph: [https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract\\_tests/TTL/ogcClassEF.ttl](https://raw.githubusercontent.com/opengeospatial/netcdf-ld/v0.2/standard/abstract_tests/TTL/ogcClassEF.ttl)

```

@prefix bald: <https://www.opengis.net/def/binary-array-ld/> .
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix this: <http://secret.binary-array-ld.net/reference.nc/> .

```

```

@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

this: a bald:Container ;
  dct:format [ a dct:MediaType ;
    dct:identifier <http://vocab.nerc.ac.uk/collection/M01/current/NC/> ] ;
  dcat:distribution [ a dcat:Distribution ;
    dcat:mediaType [ a dct:MediaType ;
      dct:identifier "application/netcdf" ] ] ;
  bald:contains this:data_variable1,
    this:data_variable2,
    this:list_collection,
    this:location_reference_system,
    this:location_variable,
    this:pdim0,
    this:pdim1,
    this:set_collection ;
  bald:isPrefixedBy "prefix_list" .

this:list_collection a bald:Resource ;
  bald:references ( this:data_variable1 this:data_variable2 ) .

this:set_collection a bald:Resource ;
  bald:references this:data_variable1,
    this:data_variable2 .

this:location_reference_system a bald:Resource ;
  this:pcode "4897" .

this:data_variable1 a bald:Array ;
  this:long_name "data variable one" ;
  bald:references [ a bald:Reference ;
    bald:target this:pdim1 ;
    bald:targetRefShape ( 1 17 ) ],
  [ a bald:Reference ;
    bald:target this:pdim0 ;
    bald:targetRefShape ( 11 1 ) ],
  [ a bald:Reference ;
    bald:sourceRefShape ( 11 17 1 ) ;
    bald:target this:location_variable ;
    bald:targetRefShape ( 11 17 2 ) ],
  this:location_variable ;
  bald:shape ( 11 17 ) .

this:data_variable2 a bald:Array ;
  this:long_name "data variable two" ;
  bald:references [ a bald:Reference ;
    bald:sourceRefShape ( 11 17 1 ) ;
    bald:target this:location_variable ;
    bald:targetRefShape ( 11 17 2 ) ],
  [ a bald:Reference ;

```

```

    bald:target this:pdim0 ;
    bald:targetRefShape ( 11 1 ) ],
  [ a bald:Reference ;
    bald:target this:pdim1 ;
    bald:targetRefShape ( 1 17 ) ],
  this:location_variable ;
bald:shape ( 11 17 ) .

this:pdim0 a bald:Array ;
  bald:shape ( 11 ) .

this:pdim1 a bald:Array ;
  bald:shape ( 17 ) .

this:location_variable a bald:Array ;
  bald:references [ a bald:Reference ;
    bald:target this:pdim1 ;
    bald:targetRefShape ( 1 17 1 ) ],
  [ a bald:Reference ;
    bald:target this:pdim0 ;
    bald:targetRefShape ( 11 1 1 ) ],
  this:location_reference_system ;
bald:shape ( 11 17 2 ) .

```

# Annex B: Relationship to other OGC Standards (Informative)

This standard is related to a number of other standards in the OGC community.

First and foremost, this standard provides a set of conventions to encode knowledge graphs built upon the [OGC Network Common Data Form \(NetCDF\) Core Encoding Standard version 1.0](#). Further, this standard makes explicit use of the rules for variables across groups defined by the Climate and Forecast (CF) Conventions for netCDF files, with CF adopted by OGC in the CF-netCDF Core and Extensions Primer. The CF Conventions for netCDF provide standardised geospatial and temporal metadata for use in netCDF files.

It should be noted that practical usage of this standard should allow for representation of the content of netCDF files using other OGC standards. For example, the European Commission's INSPIRE Spatial Data Infrastructure makes use of the ISO/OGC Abstract Specification [Observations and Measurements](#) as an encoding mechanism for, amongst other themes, meteorological and oceanographic data. INSPIRE also allows for the use of netCDF files as a data transport format from the Observation's result attribute. It can then be seen that graph patterns to identify the result within a broader netCDF file could be encoded in Observations and Measurements as Linked Data by following the standard laid out here. Further examples of this increased connectivity from this standard to others in the OGC suite of standards

- Geospatial features in the netCDF file could be mapped to the ontology of the [GeoSPARQL standard](#) to express the spatial extent of the netCDF file in Linked Data
- Explicit linkages could be made from variables described in netCDF files to, for example, descriptions of the sensor making the measurements expressed in the [Sensor Model Language](#)

Finally, whilst this standard is targeted at netCDF files, the Binary Array Linked Data vocabulary could be used to describe files adhering to the [Hierarchical Data Format Version 5](#) standard. (This is beyond the scope of the netCDF-LD standard).

# Annex C: Revision History

<b>Date</b>	<b>Release</b>	<b>Editor</b>	<b>Primary clauses modified</b>	<b>Description</b>
2021-06-29	0.1	M Hedley, J Yu	all	draft version
2021-08-10	0.2	M Hedley, J Yu	all	draft version
2021-11-24	0.3	M Hedley, J Yu	all	SWG version

# Annex D: Bibliography

Eaton, B., Gregory, J., Drach, B., Taylor, K., Hankin, S.: NetCDF Climate and Forecast (CF) Metadata Conventions, (2011).

Gregory, J.: The CF metadata standard. CLIVAR Exch. 8, 4 (2003).

Miles, A., Bechhofer, S.: SKOS Simple Knowledge Organization System Reference. (2008).

OGC: CF-netCDF Core and Extensions Primer, [http://portal.opengeospatial.org/files/?artifact\\_id=43733](http://portal.opengeospatial.org/files/?artifact_id=43733), (2011).

Rew, R., Davis, G.: NetCDF: An interface for scientific data access. IEEE Comput. Graph. Appl. 10, 76–82 (1990).

UML Profile, ISO 19103:2015 Geographic information — Conceptual schema language, <https://www.iso.org/standard/56734.html>