

Data Request XML format

Martin Juckes, January 28th, 2016.

1 Executive Summary

The Data Request is presented as two XML files: a configuration file and the content. Each file has an associated XSD schema. The XSD schema for the content file is generated automatically from the configuration file. For many users it will be more convenient to deal with the python interface or web and spreadsheet versions of the request, which will be described in a separate document. The transformation to an XML format from the traditional spreadsheet format is designed to deal with a number of issues associated with growing complexity and a need to support automation driven by the scale of the request. In order to preserve continuity, many of the records in the XML files will have a direct relation to spreadsheet rows in the traditional format.

A separate document describes a simple python API for the data request.

The variables are now also listed in a spreadsheet of MIP tables:

proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/latest/dreqPy/docs/CMIP6_MIP_tables.xlsx

2 Objectives

The broad objectives of the data request are:

- (1) Define variables, together with technical information required for generation of output files;
- (2) Define collections of variables, from specified experiments, which are needed for or relevant to specific scientific objectives;

3 Files

The framework schema:

http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/latest/dreqPy/docs/vocabFrameworkSchema_01beta.xsd

Configuration file:

<http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/latest/dreqPy/docs/dreq2Defn.xml>

Data request schema:

<http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/latest/dreqPy/docs/dreq2Schema.xsd>

Data request XML:

<http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/latest/dreqPy/docs/dreq.xml>

4 Overview

Configuration file

The XML Data Request is presented as a configuration file and a content file.

The configuration file contains three types of information:

- (1) Layout information which is used to generate the content schema;

- (2) Comments on the purpose and intent of attributes;
- (3) Technical labels to facilitate automated navigation of the contents.

If users wish to exploit the XML files directly it is recommended that they make use of the configuration file, as the information types (2) and (3) are not embedded in the content file.

Each section of the document is defined by a “table” element with the following attributes:

- label (e.g. 'var'): a name for a section of the content – will be used as the XML element name;
- title (e.g. 'MIP variable’): a longer, human readable string;
- id: an opaque name;
- itemLabelMode: specifies whether the “label” attribute of records in this section should permit use of '-';
- level: an integer, designed to assisted automated processing by giving an indication of the structure of the request;
- maxOccurs: maximum number of times the section is allowed;
- labUnique [Yes|No]: set to yes if label values for records are unique within each section.

Within each section there are definitions for attributes of items. Each item attribute is defined using the following configuration attributes:

- label: this will be the attribute name;
- title: a longer string explaining usage;
- class: the class supports automation. e.g. attributes which refer to another record in the document will have the class set to “internalLink”;
- type: the xsd content type (e.g. “XS:STRING”);
- techNote: to support automation. e.g. if class is “internalLink”, this attribute should be set to the name of the intended section:
- required: indicates whether the attribute is required;¹
- usage: notes on the usage of the attribute.¹

In addition to the standard XSD content types “string”, “boolean”, “integer”, “duration” and “float”, the following types are defined:

- st__integerList: a list of integers;¹
- st__integerListMonInc: a monotonic increasing list of integers (monotonicity is not checked by the XSD schema, but is verified by the python API);¹
- st__floatList: a list of integers;¹
- st__attLabel_def: a string composed of characters “a” to “z”, “A” to “Z”, “0” to “9” and “-”;
-

The following table summarises the specifications of the core attributes:

¹ New in 01.beta.17

label	title	description	usage
label	Record Label	A single word, with restricted character set	A short mnemonic word which is potentially meaningful but also concise and suitable for use in a programming environment
uid	Record Identifier	Unique identifier	Must be unique in the data request. For well known concepts this may be related to the label, but for items such as simple links between concepts an a random string will be used.
title	Record Title	A few words describing the object	A short phrase, suitable for use as a section heading
description	Record Description	An extended description of the object/concept.	
useClass	Record Class	The class: value should be from a defined vocabulary. All records in the schema definition section must have class set to "__core__".	The useClass declared for an attribute can affect its interpretation in the Python package. For example, attributes labelled as "useClass=internalLink" should refer to another data request record.
type	Record Type	The type specifies the XSD value type constraint, e.g. xs:string.	
techNote	Technical Note	Additional technical information which can be used to specify additional properties.	
superclass	Superclass	States what class the property is derived from	
id	Alternative identifier	Alternative identifier	For sections, the id provides a short alias for the section label.
itemLabelMode	Item Label Mode	Item Label Mode	
level	Level	Level	Redundant
maxOccurs	Maximum	Maximum number of	Used in defining sections. In the

	number of permissible occurrences of this section	permissible occurrences of this section	CMIP6 Data Request each section only occurs once.
labUnique	Set true if label of each record is unique within section	Set true if label of each record is unique within section	Used in defining sections.
usage	Usage notes	Notes on the usage of the predicate/concept defined by this node	

Content file (dreq.xml)

The content file contains three elements at the top level: “prologue”, “main” and “annex”². The “prologue” contains Dublin Core metadata describing the document. The “main” element has the sections specified in the configuration file, and within each section a list of records (“item” elements). Each item element has attributes as specified in the configuration file, a different set of attributes for each section. There are no child elements or text content, all the information is in the defined attributes. Every item, across all sections, will have at least these 3 attributes:

- uid: an identifier which is unique within the document;
- label: a short name, using only the characters a-z, A-Z, 0-9 and '-' (in some sections the '-' is disallowed);
- title: a longer name.

The “annex” element also contains a list of sections with the same structure as in the “main” element. The “annex” has been introduced to allow some flexibility in the version management.

Sections

There are 26 sections in the current preliminary document, 6 of which contain information about variables, output format and their priorities. An index to the request sections is available here: <http://clipc-services.ceda.ac.uk/dreq/index.html>

[1.1 Model Intercomparison Project \[mip\]](#)

[1.2 MIP Variable \[var\]](#)

Each MIP variable record defines a MIP variable name, associated with a CF Standard Name.

[1.3 CMOR Variable \[CMORvar\]](#)

Each Output variable record corresponds to a MIP table variable specification. In a change from the August draft, this record does not contain the “priority” attribute: the priority is now set in the “Request Variable” record. The other change is that a collection of attributes specifying dimensions etc have been moved into the “structure” record, and each CMOR Variable record links to one structure record. This will facilitate provision of clear and consistent definitions of output formats.

² New in 01.beta.16

1.4 Request variable (carrying priority and link to group) [requestVar]

The request variable is now a short record which combines a CMOR variable with a priority and assigns it to a request group.

1.5 Experiments [experiment]

The experiment record contains the key information from the “Experiment” sheet of the request template, including the tier of the experiment, the duration and start and end dates.

1.6 Scientific objectives [objective]

The objectives defined by each MIP can be used to select data requirements.

1.7 Specification of dimensions [grids]

A section for the CMOR dimensions has been added

1.8 CF Standard Names [standardname]

1.9 Experiment Group [exptgroup]

The experiment group defines a collection of experiments within a MIP which might be part of a collective data request.

2.1 Spatial dimensions [spatialShape]

The spatial shape record contains the spatial dimensions of the field, and also, for convenience, an integer specifying the number of levels if that number is specified. A boolean level flag is set to “true” if the number of vertical levels is specified.

2.2 Temporal dimension [temporalShape]

The temporal shape record contains the temporal dimensions.

2.3 Dimensions and related information [structure]

The structure record combines specification of dimensions, cell_measures and cell_methods attributes. Spatial and temporal dimensions are specified through links to “spatialshape” and “temporalshape” records.

3.1 Request variable group: a collection of request variables [requestVarGroup]

The request variable groups collect variables.

3.2 Request Item: specifying the number of years for an experiment [requestItem]

The request item links a collection of variables with a specific experiment or group of experiments, and a temporal range for output. The “esid” attribute links to an experiment, and experiment group or a MIP. In the latter case, the request applies to all experiments defined by that MIP. The Request Item includes a “Tier Reset” attribute (“treset”)³ which can override the Tier assigned to the experiments identified by “esid”. Has an optional link to a time slice³.

3.3 Request link: linking a set of variables and a set of experiments [requestLink]

The request link records specify some additional information about variable groups, concerning shared output requirements and objectives.

3.4 CMOR Table Sections [tableSection]

3.5 Model configuration options [modelConfig]

3.6 Links a variable to a choice element [varChoiceLinkC]

3.7 Link between scientific objectives and requests [objectiveLink]

Each objective link record joins one objective to one request link. Some requests are linked to multiple objectives and most objectives are linked to multiple requests.

3 New in 01.beta.17

3.08 Remarks about other items [remarks]

The remarks section contains additional comments about other records. It can be used to add detail without adding to the complexity of the other sections.

3.09 Links a variable to a choice element [varChoiceLinkR]

3.10 Indicates variables for which there is a range of potential CMOR Variables [varChoice]

There are several instances where variables defined in the tables are mutually exclusive options of which only one should be requested. The varChoice section is designed to hold this information, but is not yet complete. Examples are between ocean cell volume on a fixed grid for some models and monthly means for others, or between 6 hourly pressure level data on 8 levels vs. 4 levels for different objectives in the HighResMIP request.

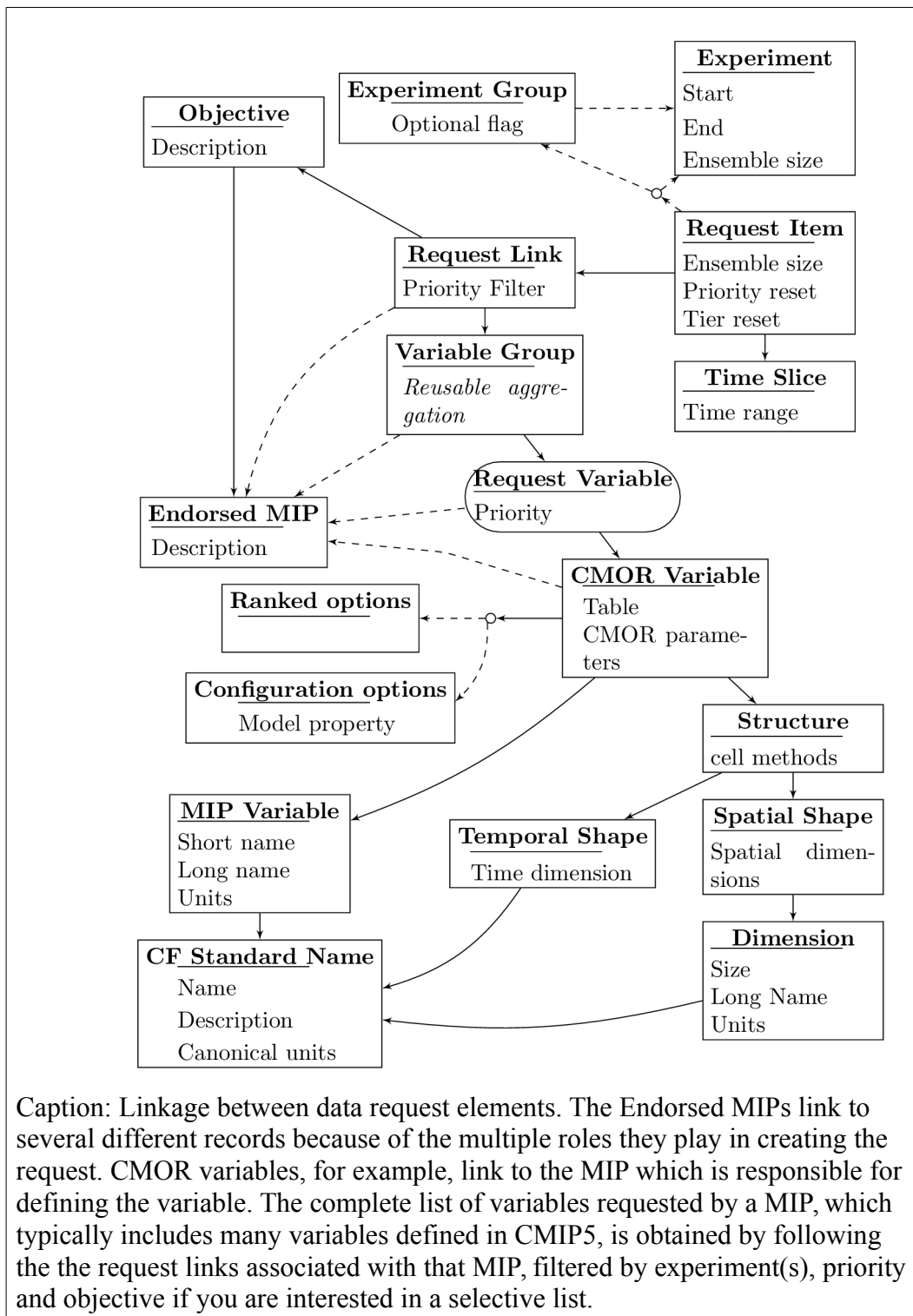
3.11 Time Slices for Output Requests [timeSlice]

Specifies time slices (i.e. subsets of an experiment when data for the full duration of the experiment is not required).

[X.1 Core Attributes \[__core__ \]](#)

[X.2 Data Request Attributes \[__main__ \]](#)

[X.3 Section Attributes \[__sect__ \]](#)



5 Discussion

The layout of the variable definitions has been rationalised into 5 sections: the “MIP variables”

defining the physical parameters, “structure”, “spatialShape” and “temporalShape” defining output configuration and a “CMOR Variable” bringing all these together. The Request Variable table then links CMOR variables together in Request Groups. The request groups give the MIP coordinators the ability to pick and choose precisely the variables needed for each analysis, avoiding requests for unnecessary data. This will result in request groups which contain overlapping data requirements. The use of links back to CMOR variables make it possible to unambiguously determine the union of any set of request groups (provided that there is no duplication of variable in the CMOR variables section – removal of duplicates will be a priority in the coming weeks).

The sections on structure and shape separate out different aspects of the CMOR variable specification and make it possible to ensure that terms are used consistently. The contents of these sections in this draft have been created by scanning the CMOR tables, and there is some duplication (e.g. the cell_measures variable attribute is set for some variables and omitted for others, creating two sets of structure records which are identical except for this distinction. In CMIP6 the cell_measures attribute will always be set).

The link between the request items and the experiment definitions is not fully implemented in this version, but the links through to the variables are. This means it is possible to gain an estimate of the data volumes for each MIP and for combinations of MIPs, but not yet to select specific tiers in a clean way (see dreqPy.pdf for more details). The data volumes given by the current version should be treated with caution. The contents may not fully reflect the intentions of the MIP coordinators, and there may be adjustments to variable priorities.

6 Annex 1: Known issues

The document still contains some irregularities. The major ones, many of which will be addressed in the next few weeks, are listed below. This list will become more comprehensive in future releases.

1. CMOR variables with no matching MIP variables

There are several CMOR variable records which do not link correctly to MIP variables

2. Block copies of tables

There are multiple cases where, for instance, PMIP has specified that all monthly variables in a group should also be requested as climatologies. This is not implemented yet.

3. Request variables with no matching CMOR variables

There are several request variables which do not link correctly to CMOR variables.

4. Dimension specifications not yet complete

5. Not all request items linked cleanly to objectives

6. cf3hr ending up in em3hr..