

# Data Request XML format

Martin Juckes, August 3<sup>rd</sup>, 2015.

## 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.

## 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/01.alpha.02/docs/vocabFrameworkSchema\\_01.alpha.xsd](http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/01.alpha.02/docs/vocabFrameworkSchema_01.alpha.xsd)

**Configuration file:**

<http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/01.alpha.02/docs/dreqDefn.xml>

**Data request schema:**

<http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/01.alpha.02/docs/dreqSchema.xsd>

**Data request XML:**

<http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/01.alpha.02/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.

## **Content file**

The content file 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 of text content, all the information is in the defined attributes. Every item, across all sections, will have at least these 3 attributes:

- uuid: 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.

There are eight sections in the current preliminary document, 4 of which contain information about variables, output format and their priorities. 3 are associated with the specification of the request: which experiments, objectives etc are associated with each variable. There is a final section which contains remarks about other records.

## **Sections**

### **1. MIP variables**

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

### **2. Output Variables**

Each Output variable record corresponds to a MIP table variable specification.

### **3. Group items**

The “group items” table combines two functions: it includes the specifications of output syntax for new variables and also allows variables to be grouped together.

### **4. Revised output variables**

The revised output variable section allows for output variables to be reused by different MIPs with a different priority.

### **5. Request variable groups**

The request variable groups collect variables.

### **6. Request link**

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

### **7. Request item**

The request item links a collection of variables with a specific experiment or group of experiments, and a temporal range for output.

### **8. 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.

## **5 Discussion**

The layout of sections reflects, to some degree, the way things have evolved. It may be simplified by combining sections 2-4 into 2 sections: one specifying the output syntax and 2<sup>nd</sup> associating a priority in a specified context (i.e. saying that an output variable should be in a specified group at a certain priority).