Data Request Python API

Martin Juckes, August 3rd, 2015

Executive Summary

The Data Request is presented as two XML files whose schema is described in a separate document. A python module is provided to facilitate use of the Data Request. Some users may prefer to work directly with the XML file or with spreadsheets and web page views, but this software provides some support for those who want to use a programming approach.

Objectives

The python API is designed to provide intuitive access to the complete collection of information.

Overview

The basic module provides two objects, the first of which contains the full information content. The 2nd provides some indexing arrays to facilitate navigation through the request.

Installation

The module is currently a simple script to be kept in the working directory. It will be packaged for pypi in the near future.

```
svn co  http://proj.badc.rl.ac.uk/svn/exarch/CMIP6dreq/tags/01.alpha.02
cd src
python simpleCheck.py
```

Usage

The box shows a piece of sample code to print a list of all the variables defined in the "var" section.

```
import dreq
dq = dreq.loadDreq()
print dq.coll.keys()
print dq.coll['var'].attDefn.keys()
for r in  dq.coll['var'].items:
   print '%20s: %s [%s]' % (r.label,r.title,r.units)
```

The content: dq.coll

The content object, dq.coll is a dictionary whose elements correspond to the data request sections represented as a "named tuple" of 3 elements: "items" (a list of records), "header" (a named tuple – see below) and "attDefn" (a dictionary with record attribute definitions).

e.g. dq.coll['var'].items[0] is the first item in the "var" section.

The items are named tuples again, so that dq.coll['var'].items[0].label is the label of the first record.

dq.coll['var'].attDefn['label'] contains the specification of the "label" attribute from the configuration file.

dq.coll['ovar'].attDefn['vid'].class = 'internalLink': this value indicates that the "vid" attribute of

records in the “ovar” section is an internalLink and so must match the “uuid” attribute of another record. To find that record, see the next section.

The index: dq.inx

The index is designed to provide additional information to facilitate use of the information in the data request.

dq.inx.uuid is a simple look-up table: dq.inx.uuid[thisId] returns a tuple of length 2: the first element is the name of the section containing the target record and the 2nd is the record.

dq.inx.iref_by_uuid gives a list of the IDs of objects which link to a given object, these are returned as a tuple of section name and identifier.

dq.inx.iref_by_sect has the same information organised differently: dq.inx.iref_by_sect[thisId].a['ovar'] is a list of the IDs of all the elements in 'ovar' which link to the given element.

There are also dictionaries for each section indexed by label and, if relevant, CF standard name.

dq.inx.var['tas'] will list the IDs of records with label='tas'

Similarly, dq.inx.var.sn['air_temperature'] give a list of records with standard name 'air_temperature'.

Outlook

In the near future a simple search API will be added, making some of the above redundant.

e.g. dq.find(section='var', sn='air_temperature') will provide the same list as dq.inx.var.sn['air_temperature'].

The search will exploit whoosh, a light-weight python module providing faceted search capability which will be adequate for this purpose. It will, however, introduce a library dependency where the present script relies only on modules in the core python distribution.