

Example Data Request Use Cases.

Table of Contents

1 Get Version of the Data Request.....	1
2 All variable requested by a MIP.....	2

1 Introduction

This document describes various approaches to some sample problems, describing how to find the information via a variety of routes.

2 Get Version of the Data Request

Web Page

Visit the home page at <http://clipc-services.ceda.ac.uk/dreq/index.html> , the version is in the title.

Command line

```
drq -v
```

From dreq.py module

This works after version 01.beta.26

```
from dreqPy import dreq
print dreq.version
```

From the XML document with minidom

The version information is in the “document/prologue/pav:version” element, as simple content.

```
import sys, os, xml
import xml.dom.minidom

# load file
xmlf = sys.argv[1]
doc = xml.dom.minidom.parse( xmlf )

# find the “ Provenance, Authoring and Versioning” version element
(http://purl.org/pav/2.3)
ver = doc.getElementsByTagName('pav:version')[0]
print ver.firstChild.data
```

3 All variable requested by a MIP.

MIPs may request variables they have defined and variables defined by others. To get a complete list of all variables associated

Web

Visit http://clipc-services.ceda.ac.uk/dreq/tab01_1_1.html for Tier 1, priority 1 variables of http://clipc-services.ceda.ac.uk/dreq/tab01_3_3.html for all variables, and follow the link from the right hand column of the “DCPP” row.

Command line

This command will place a spreadsheet file named “DCPP-all_1_1.xlsx” in the directory “xls”. The file will list all the priority 1 variables requested by DCPP. To include all variables, add “-p 3” to the command line.

```
drq -m DCPP -xls
```

The command can be generalised to obtain a list of variables requested by multiple MIPs by replacing “DCPP” with a comma-separated (no spaces) list of MIPs.

Using the scope.py module

The scope module provides some high level routines for common requests.

As above, this will generate a spreadsheet and write it to the “xls” directory. Set “p=3” to get all variables.

```
from dreqPy import scope
sc = scope.dreqQuery()
p=1
sc.xlsByMipExpt('DCPP',None,p,odir='xls')
```

To generalise to cover multiple MIPs, replace the string 'DCPP' with a set of strings, e.g. “set(['DCPP','C4MIP','AerChemMIP'])”.

Using dreq.py

The dreq module reads the data request into a structure which closely resembles the underlying XML file, but it has some methods to facilitate the linking between records.

```

from dreqPy import dreq
dq = dreq.loadDreq()

# select the “requestLink” items defined by DCPP
dcppLinks = [x for x in dq.coll['requestLink'].items if x.mip == 'DCPP']
print 'Request links found for DCPP: %s' % len( dcppLinks )

# find the associated variable groups (some might be re-used groups defined by
other MIPs)
dcppRequestedGroups = set( [dq.inx.uid[x.refid] for x in dcppLinks] )
print 'Variable groups found for DCPP: %s' % len( dcppRequestedGroups )

# find the requestVar elements that link to the variable groups, using the iref
index set up in the dreq package.
dcppRequestedCmorVars = set()
for vg in dcppRequestedGroups:
    for x in dq.inx.iref_by_sect[vg.uid].a['requestVar']:
        dcppRequestedCmorVars.add( dq.inx.uid[x].vid )
print 'Requested CMOR Variables found for DCPP: %s' %
len( dcppRequestedCmorVars )

```

This code fragment will generate a list of CMOR variables (all priorities). The code can be generalised by modifying the test used to filter the requestLink records in the 5th line.

Using the xml.dom.minidom module.

The data request may also be accessed directly using any XML library. This example use the minidom module of the python core xml package.

As above, the script generates a list of CMOR variable identifiers.

```
import sys, os, xml

xmlf = sys.argv[1]
doc = xml.dom.minidom.parse( xmlf )

dcppLinks = []
requestedGroups = set()

# find request links
sect = doc.getElementsByTagName( 'requestLink' )[0]
s = set()
for item in sect.getElementsByTagName( 'item' ):
    if item.getAttribute( 'mip' ) == 'DCPP':
        dcppLinks.append( item )
        requestedGroups.add( item.getAttribute( 'refid' ) )
print 'Request links found for DCP: %s' % len( dcppLinks )

print 'Request groups found for DCP: %s' % len( requestedGroups )

# find request variables and CMOR variables
sect = doc.getElementsByTagName( 'requestVar' )[0]
cmorVars = set()
for item in sect.getElementsByTagName( 'item' ):
    if item.getAttribute( 'vgid' ) in requestedGroups:
        cmorVars.add( item.getAttribute( 'vid' ) )

print 'CMOR variables requested by DCP: %s' % len(cmorVars)
```