SCIDIP-ES
SCIENCE DATA INFRASTRUCTURE FOR PRESERVATION - EARTH SCIENCE

D21.3 Data Virtualization Toolkit Installation, Deployment and User Manual

Work package	WP21	Services/Toolkits Development and Adaptation	
Task	T21.5 Data Virtualizaiton Toolkit Development		
Author (s)	Shirley Crompton		STFC
Author (s)	Jinsongdi Yu		Jacobs University
Author (s)			
Author (s)			
Author (s)			
Author (s)			
Author (s)			
Authorized by			
Reviewer	Name Surname		Company
Doc Id			
Dissemination Level		~~CONFIDENTIAL/~~PUBLIC	
Issue	1.0		
Date	17/02/2014		

Abstract:

This is the Deployment and User Manual for the Data Virtualisation Toolkit, a software developed for the SCIDIP-ES project. This document contains relevant information on the design, installation and usage of the Data Virtualisation Toolkit.

Document Log

Date	Author	Changes	Version	Status
17/02/2014	Shirley Crompton, Jinsongdi Yu	Created base document from D21.4 and updated for M30 release	1.0	Draft

TABLE OF CONTENTS

1 Introduction

1.1 Purpose and Scope

This document provides an overview of the M30 release of the Data Virtualisation Toolkit (DVT) focusing in particular to its design, installation and usage.

1.2 Who should read this document

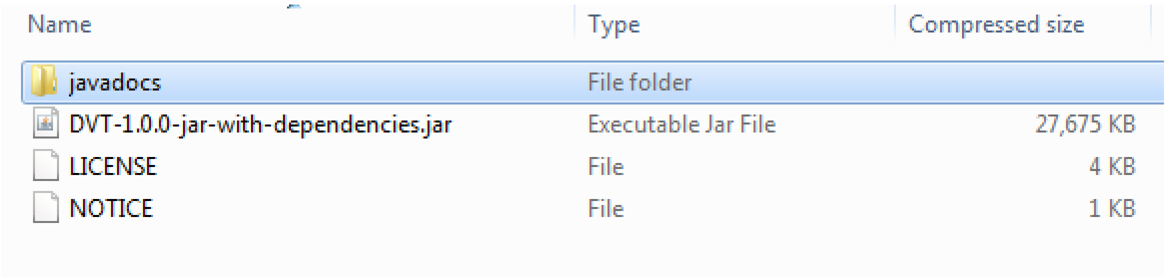
Users who wish to install and use the DVT.

1.3 System Context

DVT is developed as part of the SCIDIP-ES e-infrastructure. Its main application is to enable format independent access of preserved digital data objects. DVT can be used to create/edit Structural Representation Information (see Section 3) which maps bits sequences into data types and then to higher level concepts needed to understand a digital data object.

1.4 Release Notes

The release is packaged as archive files (zip and tar.gz) which have the following structure:



Name	Type	Compressed size
javadocs	File folder	
DVT-1.0.0-jar-with-dependencies.jar	Executable Jar File	27,675 KB
LICENSE	File	4 KB
NOTICE	File	1 KB

Figure 1. Data Virtualization Toolkit Archive File Structure

1.5 License and Conditions of Use

DVT is licensed under the Apache License, Version 2.0 (the "License"). You may not use this software except in compliance with the License. A copy of the License could be obtained at: http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

DVT uses:

- the SAXON XSLT Processor from Michael Kay (http://saxon.sourceforge.net/) which is licensed under Mozilla Public License Version 1.0 (http://www.mozilla.org/MPL/)
- the Data Request Broker (http://www.gael.fr/drb/) from GAEL Consultancy as the bit stream interpreter. DRB is licensed under the GNU Lesser Public License Version 3.0 (http://www.gael.fr/drb/license.html)

- TOPCAT (http://www.star.bris.ac.uk/~mbt/topcat/), an interactive graphical viewer and editor for tabular data developed for the Astronomy domain, to demonstrate the tool's visualization functionality. TOPCAT is licensed under the General Public License (http://www.gnu.org/copyleft/gpl.html).

2 Installation Guide

2.1 Overview

DVT is provided as a self-contained jar file.

2.2 Download Information

The recent stable source code could be accessed from the *Sourceforge* SVN. The URL to the svn trunk is: svn://svn.code.sf.net/p/digitalpreserve/code/SCIDIP-ES/software/toolkits/DataVirtualizationToolkit/trunk

Milestone releases of the software may also be downloaded via the SCIDIP-ES maven nexus repository at: http://nexus.scidip-es.eu/content/repositories/releases/eu/scidipes/toolkits/datavirtualizationtoolkit/DVT/

2.3 Prerequisites

2.3.1 Software prerequisites

Software prerequisites respect SCIDIP-ES guidelines. The only specific prerequisite is the availability of a Java 6 JVM or upwards on the target system.

2.3.2 Hardware prerequisites

None.

2.4 OSS/COTS Installation

None.

2.4 Data Virtualization Toolkit Installation

After downloading the distribution archive (see Sections 1.4 and 2.2), user should unpack it. The tool can be started by either:

1) Double clicking the jar file directly
2) In the command line, enter the following command:

```
java –jar <path/to/the/jar/>DVT-1.0.0-jar-with-dependencies.jar
```

2.5 2.4.1 Uninstallation

DVT does not need to be uninstalled. If no longer required, user can simply delete the DVT jar file and the associated files (see Figure 1).

3 Software Design

DVT uses the concept of Open Archival Information System[1] (OAIS) Representation Information (RepInfo) to provide the support to explore the structure and contents of a digital data object in a format independent manner. RepInfo is defined in OAIS as the additional information that maps a Data Object into more meaningful concepts. OAIS further categories RepInfo into three main types:

- Structure
- Semantic
- Other.

Structure RepInfo describes the format or data structure concepts expressed as mapping rules for mapping the digital bits sequences into data types, and then to higher level concepts needed to understand a digital Data Object. DVT uses Structure RepInfo in specific to facilitate format independent access to preserved data.

The tool's design is based on the bit stream interpretation API, e.g. DRB[2] API is used in this implementation, to provide a software abstraction layer that helps developers to abstract data format specific concerns from their analysis or data access applications. Abstraction is achieved using the physical data object model and standardized ES models. See [Yu2013] for a case study of using DVT in facilitating Earth Science (ES) data interoperation.

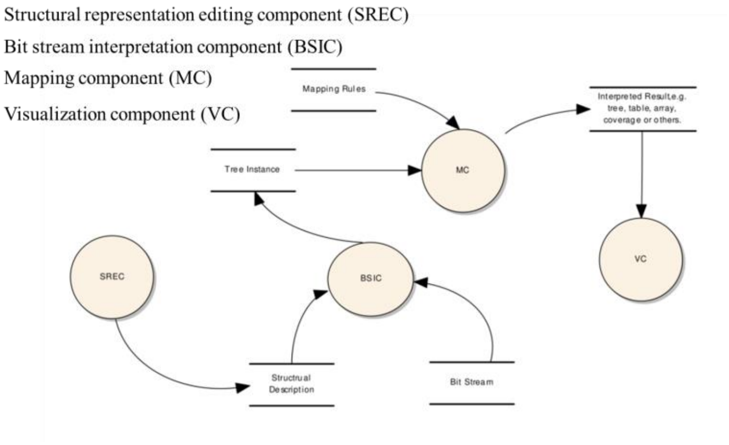

Structural representation editing component (SREC)
Bit stream interpretation component (BSIC)
Mapping component (MC)
Visualization component (VC)
Mapping Rules
Interpreted Resulte.g. tree, table, array, coverage or others.
Tree Instance
MC
VC
SREC
BSIC
Structrual Description
Bit Stream

Figure 2. DVT data flow diagram.

DVT is written in pure Java as a SWING[3] GUI tool that offers four main components (Figure 2):

- The structural representation editing component (SREC) - performs create, delete, update and insert operations on an ordered tree structure, eg. XML[4]. The output is a structural description which describes the physical data structure.
- The bit stream interpretation component (BSIC) - takes preserved digital data from archives as input and interprets the bit stream according to the given structure representation. The result is an ordered tree instance of the given representation model.
- The mapping component (MC) - takes the ordered tree instance as input and constructs legacy

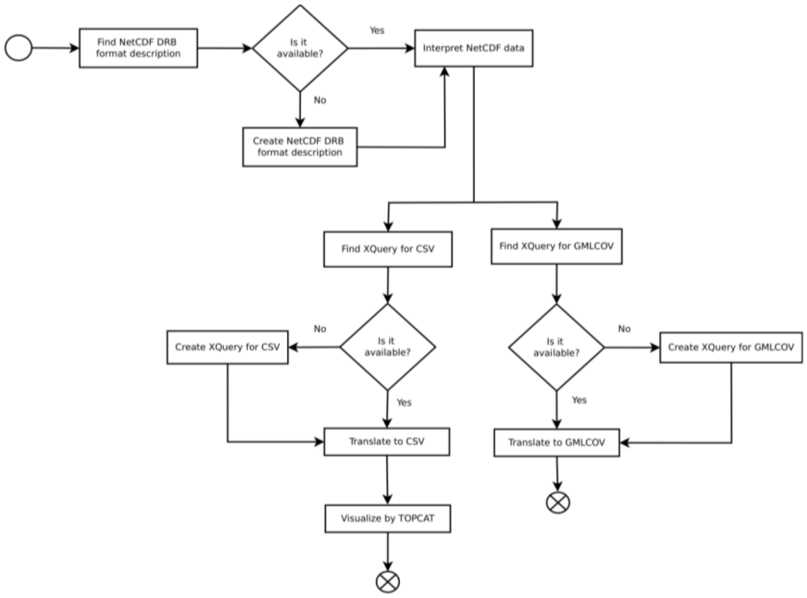
[1] OAIS - http://public.ccsds.org/publications/archive/650x0m2.pdf
[2] Data Request Broker - http://www.gael.fr/drb/
[3] SWING - http://docs.oracle.com/javase/7/docs/technotes/guides/swing/
[4] Extensible Markup Language - http://www.w3.org/XML/

information from data values according to mapping rules of the given standardized information model. The result is an instance of the given model.

- The visualization component (VC) - uses the configured tools to visualize the result.

The tool supports generic interaction as illustrated in the example workflow for NetCDF[5] data (Figure 3). These involve the user:

- creating/supplying a DRB description for interpreting the target NetCDF data. DVT uses the DRB bit stream interpreter to convert the NetCDF data into generic XML format
- using custom XQuery aligned to a standard information model (eg. CSV[6], GMLCOV[7]) to extract and format specific data from the XML
- visualizing the extracted data in two different ways:
  - Extensively as a table
  - Intensively as a GML grid coverage (GMLCOV).



4 Using SCIDIP-ES Data Virtualization Toolkit

4.1 Basic Concepts

Current DVT demonstrates three basic functions:

- Data extraction: users provide a Structure RepInfo for extracting information from the target digital data. The output depends on the provided RepInfo and the processor used: DRB description and the DRB bit stream interpreter are used in this implementation.

[5] Network Common Data Format - http://www.unidata.ucar.edu/software/netcdf/
[6] Comma Separated Values - http://www.digitalpreservation.gov/formats/fdd/fdd000323.shtml
[7] GML Application Schema: Coverages - http://www.opengeospatial.org/standards/gml

- Data transformation: users can use it to translate the extracted information to the required format. The behaviour of this operation depends on the provided translation script and engine used: XQueries and SAXON are used in this implementation.
- Data visualization: user can visualize the derived data using common rendering tools. By mapping the original data to standardised formats, eg. OGC coverage, it is possible to access and use legacy data without the need for custom or legacy tools.

3.2 The Basic Functions

We illustrate the use of DVT using an example of exploring and querying BADC[8] netCDF data using Data Request Broker (DRB) description and to visualize the results using TOPCAT[9].

When the tool started it presents a window similar to that in the screenshot below:

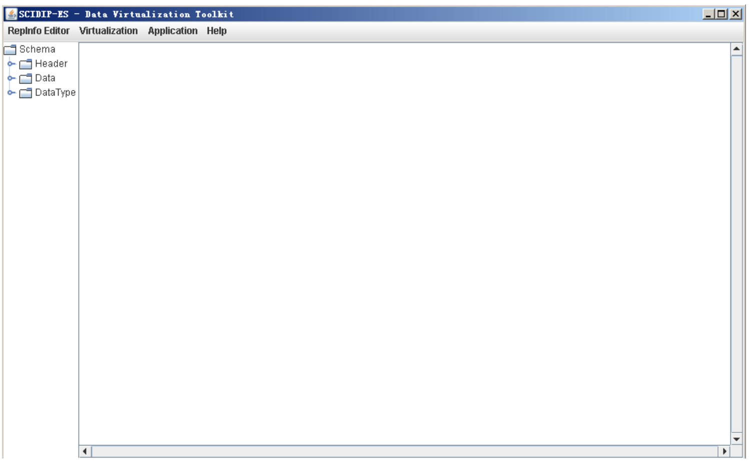


Figure 3. DVT Main GUI Dialog.

Get DRB description:

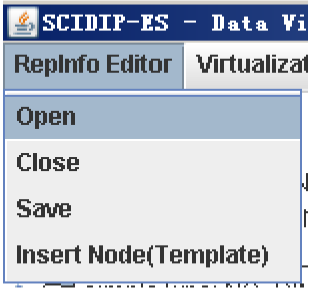
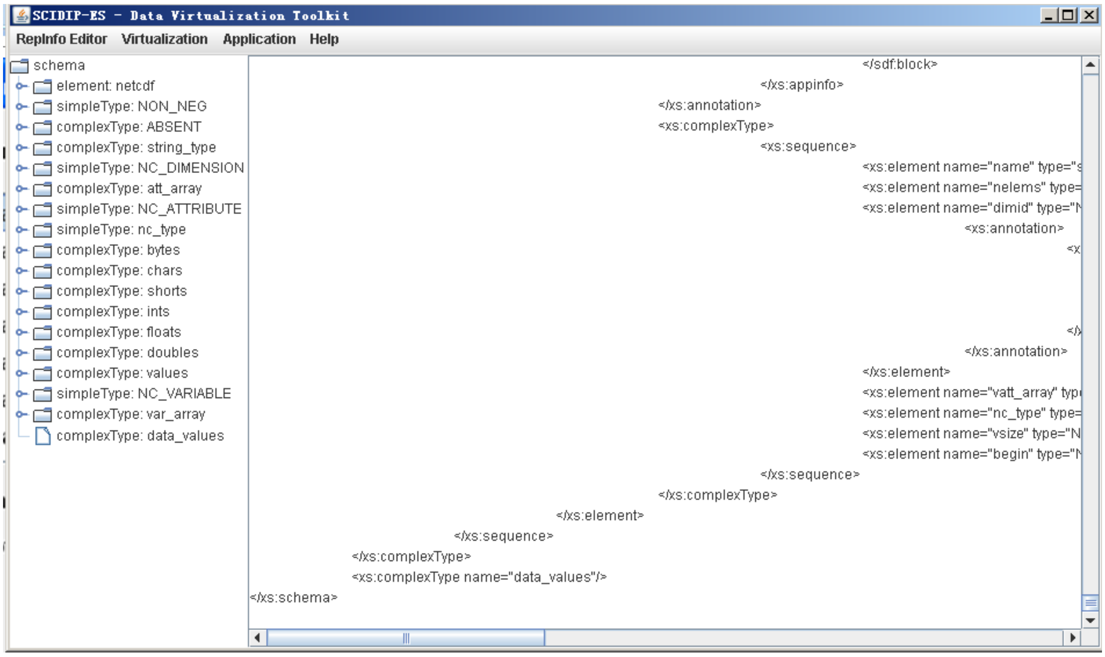


Figure 4. Menu item for opening a DRB description**.**

DRB description displayed in editor dialog:

[8] http://badc.nerc.ac.uk/home/index.html

[9] http://www.star.bristol.ac.uk/~mbt/topcat/



```
schema
  element: netcdf
  simpleType: NON_NEG
  complexType: ABSENT
  complexType: string_type
  simpleType: NC_DIMENSION
  complexType: att_array
  simpleType: NC_ATTRIBUTE
  simpleType: nc_type
  complexType: bytes
  complexType: chars
  complexType: shorts
  complexType: ints
  complexType: floats
  complexType: doubles
  complexType: values
  simpleType: NC_VARIABLE
  complexType: var_array
  complexType: data_values
```

```
                        </sdf:block>
                    </xs:appinfo>
                </xs:annotation>
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="name" type="s
                        <xs:element name="nelems" type=
                        <xs:element name="dimid" type="N
                            <xs:annotation>
                                <x



                                </x
                            </xs:annotation>
                        </xs:element>
                        <xs:element name="vatt_array" typ
                        <xs:element name="nc_type" type=
                        <xs:element name="vsize" type="N
                        <xs:element name="begin" type="N
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="data_values"/>
</xs:schema>
```

Figure 5. Viewing a DRB description.

Insert a new DRB node into description:

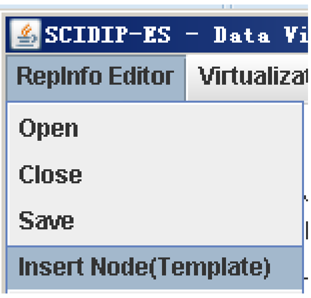
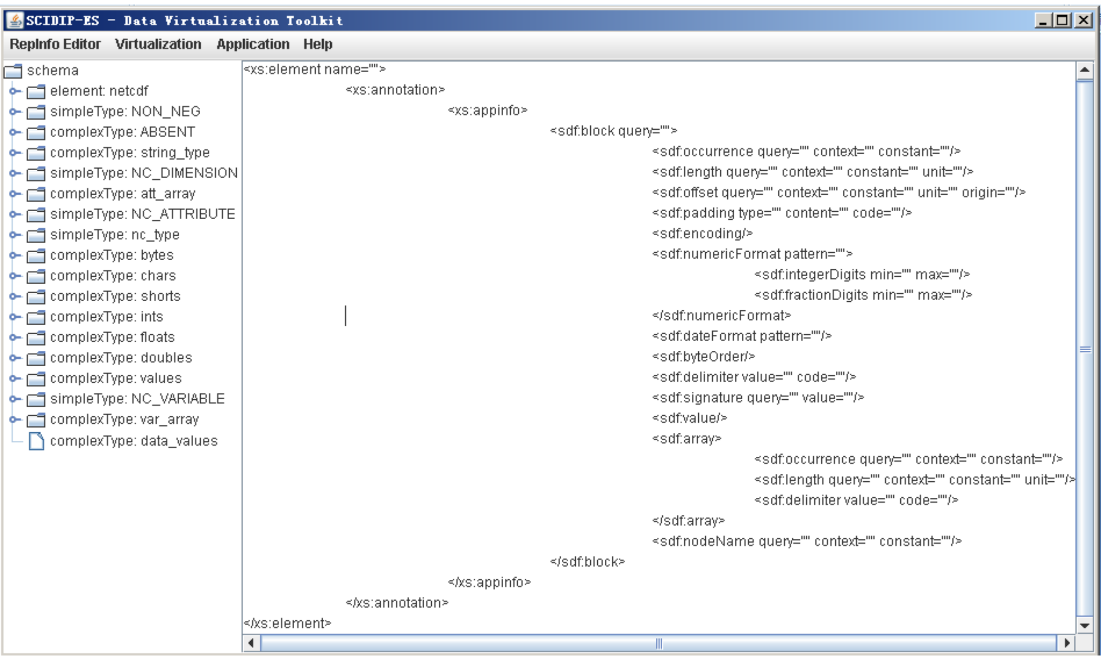


Figure 6. Menu item for inserting a new node.

A blank node is inserted:



- schema
  - element: netcdf
  - simpleType: NON_NEG
  - complexType: ABSENT
  - complexType: string_type
  - simpleType: NC_DIMENSION
  - complexType: att_array
  - simpleType: NC_ATTRIBUTE
  - simpleType: nc_type
  - complexType: bytes
  - complexType: chars
  - complexType: shorts
  - complexType: ints
  - complexType: floats
  - complexType: doubles
  - complexType: values
  - simpleType: NC_VARIABLE
  - complexType: var_array
  - complexType: data_values

```
<xs:element name="">
        <xs:annotation>
                <xs:appinfo>
                        <sdf:block query="">
                                <sdf:occurrence query="" context="" constant=""/>
                                <sdf:length query="" context="" constant="" unit=""/>
                                <sdf:offset query="" context="" constant="" unit="" origin=""/>
                                <sdf:padding type="" content="" code=""/>
                                <sdf:encoding/>
                                <sdf:numericFormat pattern="">
                                        <sdf:integerDigits min="" max=""/>
                                        <sdf:fractionDigits min="" max=""/>
                                </sdf:numericFormat>
                                <sdf:dateFormat pattern=""/>
                                <sdf:byteOrder/>
                                <sdf:delimiter value="" code=""/>
                                <sdf:signature query="" value=""/>
                                <sdf:value/>
                                <sdf:array>
                                        <sdf:occurrence query="" context="" constant=""/>
                                        <sdf:length query="" context="" constant="" unit=""/>
                                        <sdf:delimiter value="" code=""/>
                                </sdf:array>
                                <sdf:nodeName query="" context="" constant=""/>
                        </sdf:block>
                </xs:appinfo>
        </xs:annotation>
</xs:element>
```

Figure 7. A default DRB node is inserted into the description.

Perform extraction of data:

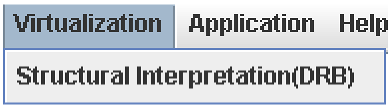
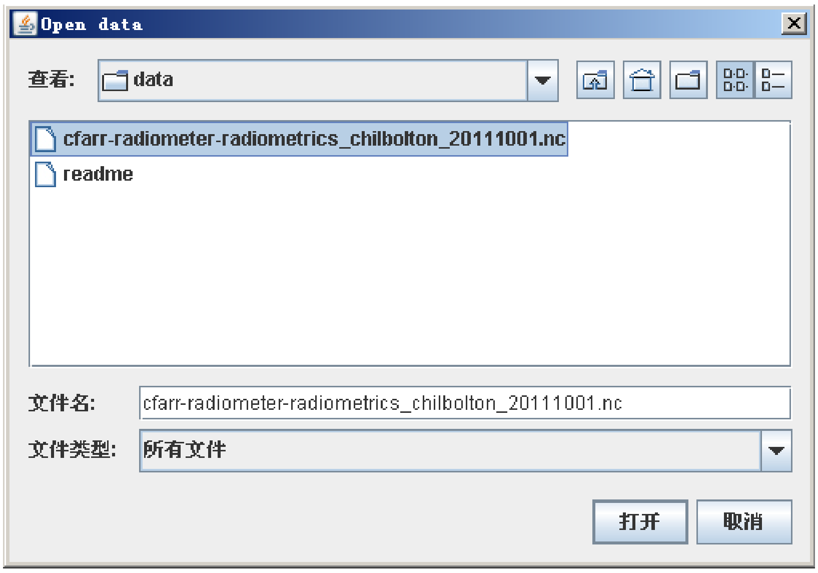


Figure 8. Menu item for performing data extraction using a Structural RepInfo (i.e. the DRB description).



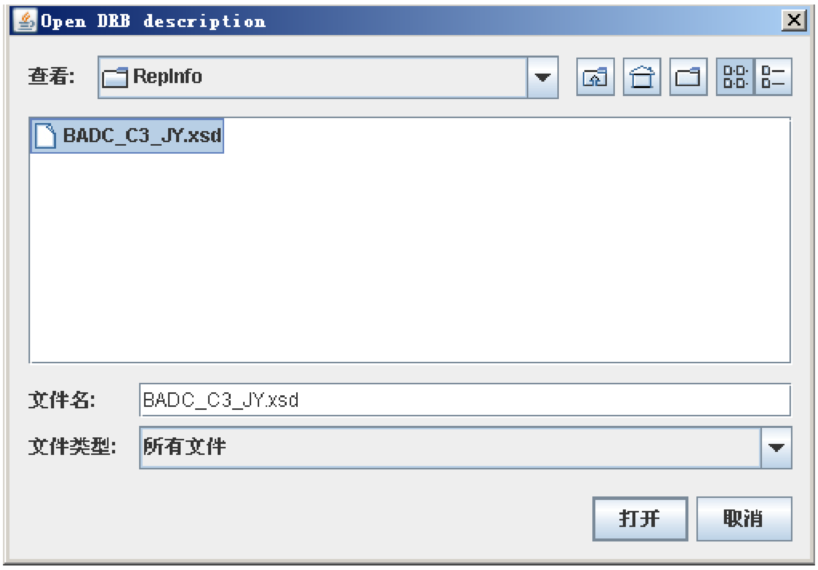
查看: data

- cfarr-radiometer-radiometrics_chilbolton_20111001.nc
- readme

文件名: cfarr-radiometer-radiometrics_chilbolton_20111001.nc

文件类型: 所有文件

Figure 9. Select the source data file.



Open DRB description

查看: RepInfo

BADC_C3_JY.xsd

文件名: BADC_C3_JY.xsd

文件类型: 所有文件

Figure 10. Select the DRB description to apply to the data.

Transform netCDF data to a generic CSV format:

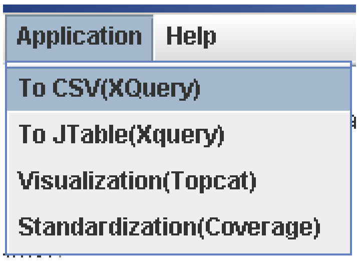


Figure 11. Menu item for extracting custom data into generic csv formt.

Visualise CSV data as a JTable:

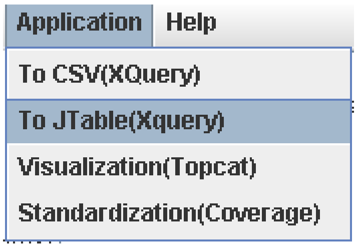
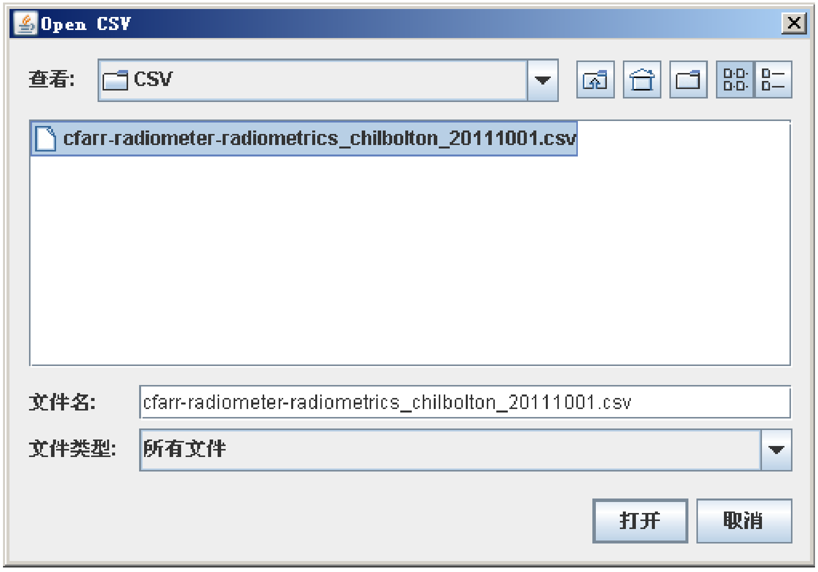


Figure 12. Menu item for visualising CSV data as a JTable.



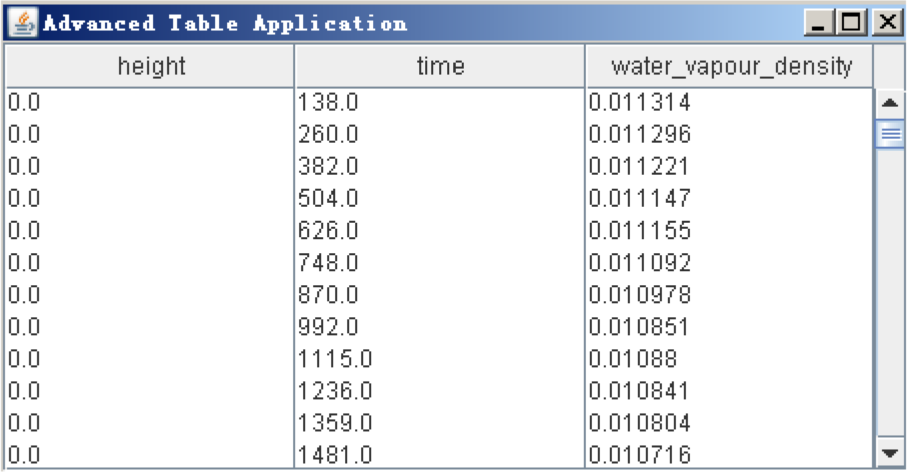
查看: CSV

cfarr-radiometer-radiometrics_chilbolton_20111001.csv

文件名: cfarr-radiometer-radiometrics_chilbolton_20111001.csv

文件类型: 所有文件

Figure 13. Select the CSV data to view.



height	time	water_vapour_density
0.0	138.0	0.011314
0.0	260.0	0.011296
0.0	382.0	0.011221
0.0	504.0	0.011147
0.0	626.0	0.011155
0.0	748.0	0.011092
0.0	870.0	0.010978
0.0	992.0	0.010851
0.0	1115.0	0.01088
0.0	1236.0	0.010841
0.0	1359.0	0.010804
0.0	1481.0	0.010716

Figure 14. Data displayed as a JTable.

Visualise CSV data using TopCat:

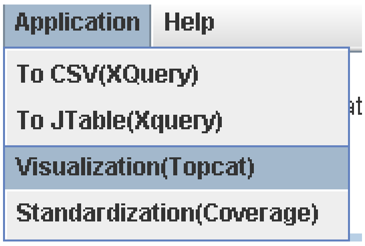
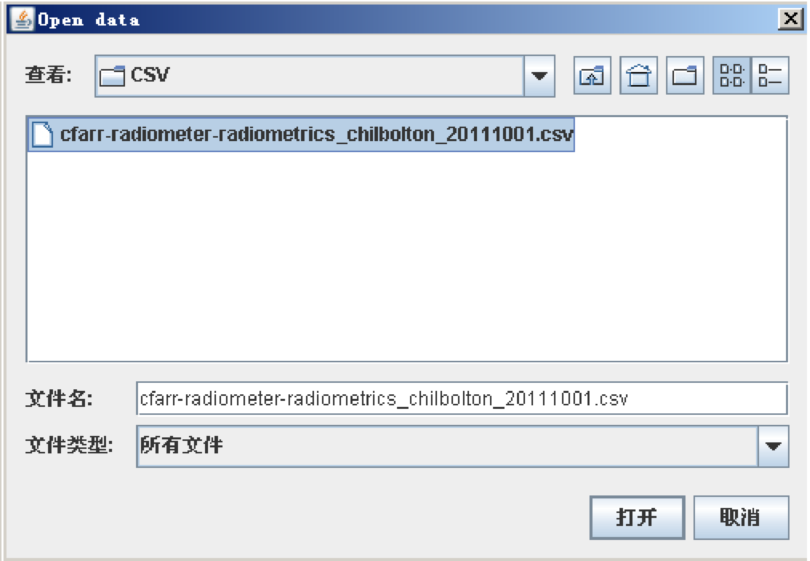


Figure 15. Menu item for visualising data using TopCat.



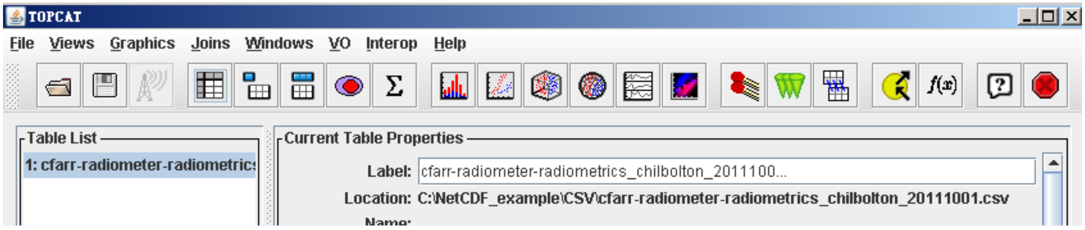
查看: CSV

cfarr-radiometer-radiometrics_chilbolton_20111001.csv

文件名: cfarr-radiometer-radiometrics_chilbolton_20111001.csv

文件类型: 所有文件

Figure 16. Select CSV data to view.



1: cfarr-radiometer-radiometric:

Label: cfarr-radiometer-radiometrics_chilbolton_2011100...

Location: C:\NetCDF_example\CSV\cfarr-radiometer-radiometrics_chilbolton_20111001.csv

Name:

Figure 17. Partial snapshot of the TopCat GUI with the CSV file loaded.

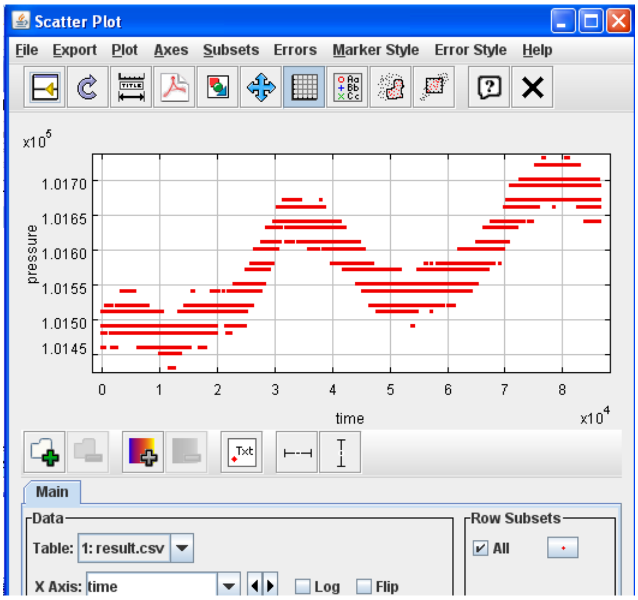


Table: 1: result.csv

X Axis: time ☐ Log ☐ Flip

☑ All

Figure 18. Bivariat scatter plot of the CSV data.

Standardization (OGC Coverage 2D) demon:

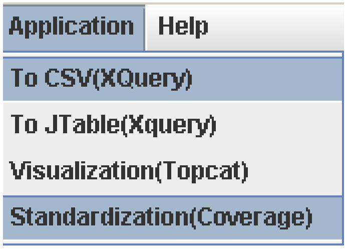
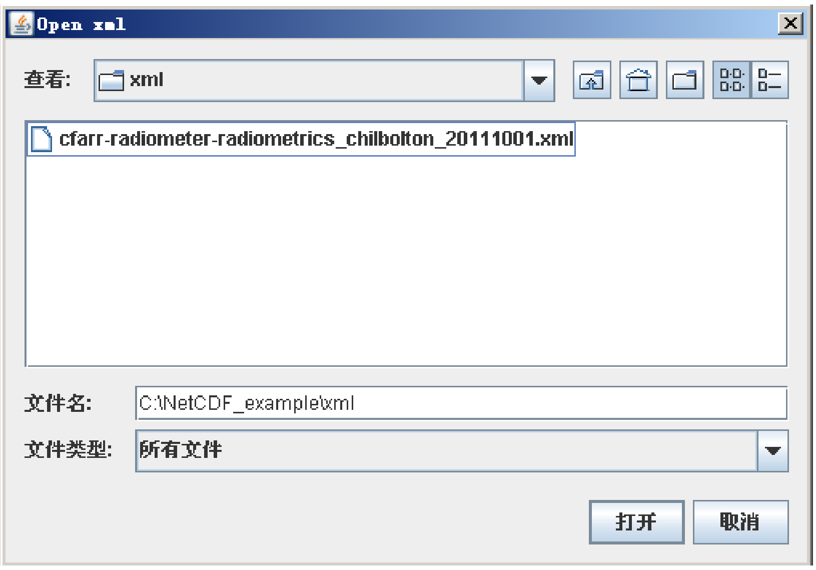


Figure 19. Menu Item for Standardization (Coverage) operation.



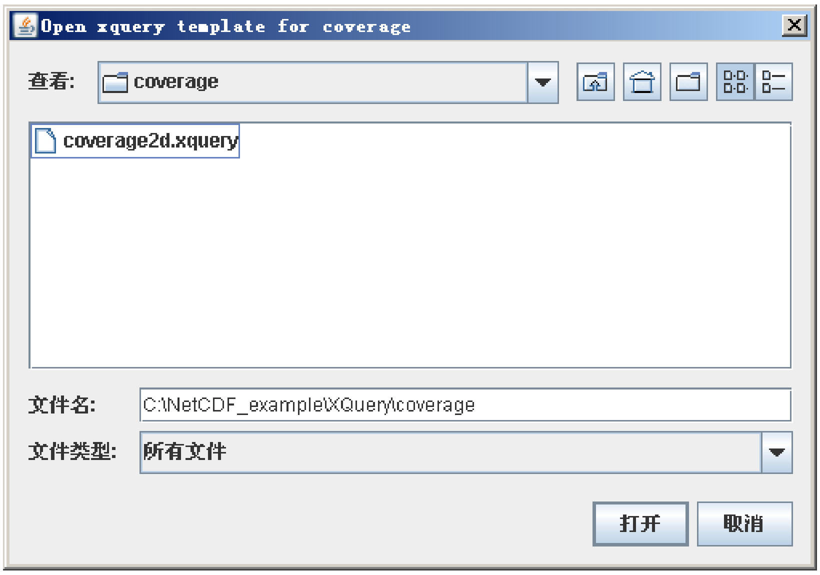
查看: xml

cfarr-radiometer-radiometrics_chilbolton_20111001.xml

文件名: C:\NetCDF_example\xml

文件类型: 所有文件

Figure 20. Select the XML representation of the netCDF data obtained from the previous demo.



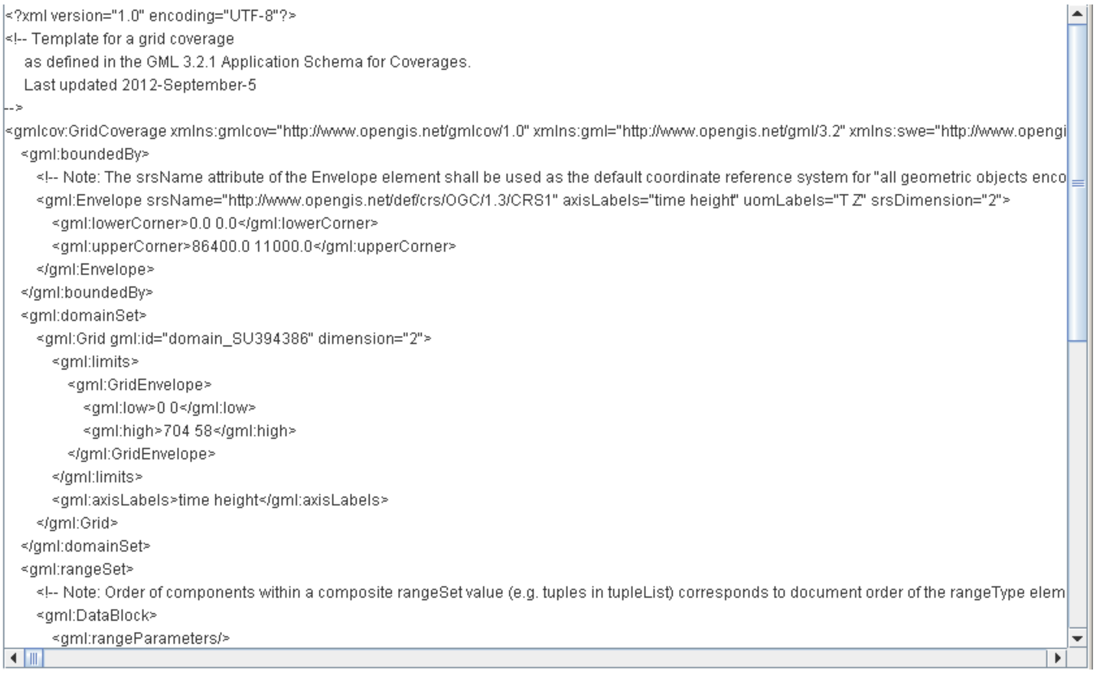
查看: coverage

coverage2d.xquery

文件名: C:\NetCDF_example\XQuery\coverage

文件类型: 所有文件

Figure 21. Select the XQuery to perform the transformation.



```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Template for a grid coverage
   as defined in the GML 3.2.1 Application Schema for Coverages.
   Last updated 2012-September-5
-->
<gmlcov:GridCoverage xmlns:gmlcov="http://www.opengis.net/gmlcov/1.0" xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:swe="http://www.opengi
  <gml:boundedBy>
    <!-- Note: The srsName attribute of the Envelope element shall be used as the default coordinate reference system for "all geometric objects enco
    <gml:Envelope srsName="http://www.opengis.net/def/crs/OGC/1.3/CRS1" axisLabels="time height" uomLabels="T Z" srsDimension="2">
      <gml:lowerCorner>0.0 0.0</gml:lowerCorner>
      <gml:upperCorner>86400.0 11000.0</gml:upperCorner>
    </gml:Envelope>
  </gml:boundedBy>
  <gml:domainSet>
    <gml:Grid gml:id="domain_SU394386" dimension="2">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>0 0</gml:low>
          <gml:high>704 58</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisLabels>time height</gml:axisLabels>
    </gml:Grid>
  </gml:domainSet>
  <gml:rangeSet>
    <!-- Note: Order of components within a composite rangeSet value (e.g. tuples in tupleList) corresponds to document order of the rangeType elem
    <gml:DataBlock>
      <gml:rangeParameters/>
```

Figure 22. The OGS Coverage XML output from the operation.

5 Future Work

Further enhancement will be provided in the M30 release to offer more intuitive navigation within an operation and to provide an interface to support programmatic access.

6 Reference Manual

None

6.1 Keyboard shortcuts

None

6.2 Command-line commands

None

6.3 Public APIs

None

7 Troubleshooting Common Issues

NA

Annex A. References

[Yu2013] Yu, Jinsongdi, Baumann, Peter and Crompton, Shirley: Facilitates ES Data Interoperability using the SCIDIP-ES Data Virtualisation Toolkit, *Pro. Ensuring Long-term Preservation and Adding Value to Scientific and Technical Data* (PV2013), Frascati, Italy (2013). URL: http://www.congrexprojects.com/docs/default-source/13c17_docs/pv2013-paperbook.pdf?sfvrsn=2

Annex B. Figures and Tables

B.1. List of Figures

Annex C. Terminology

ACRONYM	DESCRIPTION
ES	Earth Science
OAIS	Open Archive Information System
RepInfo	Representation Information
VM	Virtual Machine
WP	Work Package
XML	eXtensible Mark-up Language