Practical Data Interoperability for Earth Scientists

Version 1.0

Christopher Lynnes, Ken Keiser, Ruth Duerr, Terry Haran, Lisa Ballagh, Bruce H. Raup, Bruce Wilson

Earth Science Data Systems Working Group

What Is Data Interoperability?

Data interoperability is the ability for a data user to work with (view, analyze, process, etc.) a data provider's science data or model output "transparently," without having to reformat the data, write special tools to read or extract the data, or rely on specific proprietary software.

Why Bother?

Inter-disciplinary science*:* There is an increasing emphasis on Earth System Science, that is, the study of the Earth as a system, rather than individual components. This often requires combining diverse datasets from multiple data providers.

Foster collaborations: Earth System Science increasingly requires more collaborations among scientists from various disciplines, who may use software packages from different sources.

Enable model/data integration and model coupling: Data interoperability helps models to incorporate observational data, and facilitates the use of models to corroborate observational data. Similarly, data interoperability facilitates coupling the output of one model to the input of another.

Make data analysis easier: Many analysis and visualization tools are able to read data files with a single operation if the data are available in an interoperable form.

Producing interoperable data is easier than it looks: ...at least for gridded geographical datasets. You don't need to know the details of standards, formats, etc., because there are a number of tools available to help make data interoperable. The rest of this guide presents data formats, protocols, and tools that provide interoperability by removing dependencies on specific operating systems, software packages or environments. (This overview omits details for brevity's sake.)

Target Audience

This document is designed for both Principal Investigators and their collaborators in Earth science, as well as data centers; that is, anyone who makes Earth science data products available to Earth scientists. We recognize that this represents a wide range of information technology expertise. Accordingly, we have labeled the interoperability technologies below according to the proficiency required to install, configure and use them to provide interoperable data:

Beginner: I (or my system administrator) can follow simple, straightforward install and setup instructions.

Intermediate: I can follow more complicated instructions, so long as they are well-documented and mostly bulletproof.

Advanced: I can follow complex instructions and do some troubleshooting if things go awry.

Expert: I can debug anything that comes my way.

Also, note that sections such as "Limitations" are with respect to its use in Earth science,

specifically. (Some such limitations are viewed as features in the wider community.)

Data Formats

netCDF/CF-1

NetCDF is a relatively simple format that is especially popular for gridded data. The CF-1 (http://cf-pcmdi.llnl.gov/) conventions are a way of describing data within netCDF in a uniform manner that can be interpreted by analysis and display tools. Many tools can read netCDF files that are not in CF-1, but will not be able to use many of the map-related functions in that case.

What Is It Good For?

NetCDF/CF-1 formatted data can be easily read by many client-side tools, among them freeware such as Ferret, GrADS, Integrated Data Viewer (IDV), GDAL, OpenEV, Panoply and commercial tools such as ArcView. Not only can the data values be read, but the geographic coordinates are properly recognized and interpreted and typically displayed in map view.

NetCDF/CF-1 works well for data on a geographic grid. It is also useful for time series data, such as flux tower data. Many of the netCDF tools can handle the time series data.

Limitations

NetCDF/CF-1 is less useful for data that is not on a regular grid. Examples include satellite swath data, which are difficult (albeit not impossible) to describe in CF-1 so that tools can properly read and interpret them.

NetCDF Version 3 does not support internal compression, so it may not be suitable for very large data volumes. Version 4 is built on top of HDF-5, which does support internal compression.

Proficiency Level:

Intermediate

Languages:

netCDF application programming interfaces are available in many languages, including C, C++, Fortran, Java, Perl and Python.

Obtaining and Installing

The basic netCDF library can be obtained from http://www.unidata.ucar.edu/software/netcdf/. It installs with standard GNU autotools.[1]

Hierarchical Data Format (HDF) and HDF-EOS

HDF is a rich data format offering a variety of data structures accompanied by an API for reading and writing those data structures. HDF-EOS is an API on top of HDF that implements additional data structures designed to facilitate access to Earth science data, in particular structures that associate geolocation information with data.

[1] GNU autotools is a toolbox for automating cross-platform installation of code. Typically, they enable installation to be a standard, two-step process: a "configure" script is run (often with optional arguments) to set up makefiles for the local environment, followed by "make install" to build and install the software.

What Is It Good For?

HDF is good for storing complicated data files together with their metadata. In addition, its internal compression capabilities are useful for storing large data volumes. The internal compression is transparently uncompressed on the fly via the HDF API so no change to read software is needed. A useful utility, hrepack, is provided for compressing or uncompressing HDF data. A variety of freeware (e.g., WebWinds, HDF Explorer, OpenDx, GrADS, GDAL, OpenEV) and commercial (e.g., ENVI, ERDAS Imagine, Mathematica, MATLAB) packages can read and manipulate HDF data.

HDF-EOS adds certain standard metadata, both at the data object and file level, that make the data easier to interpret by client programs. In particular, HDF-EOS provides structural metadata that can allow a client program to determine the latitude and longitude of any cell in a gridded data object. The three types of HDF-EOS data objects supported are Grid, Swath (of a satellite) and Point data.

Limitations

Despite the HDF-EOS standards, there are fewer tools that can read and interpret the geographic information in HDF-EOS files than there are for netCDF/CF-1.

HDF also has a non-trivial learning curve.

Proficiency Level

Advanced

Languages:

HDF libraries are available for C, Fortran, Java, and IDL. Add-on modules have been written for Python and Perl, though they do not necessarily implement the full Application Programming Interface (API). IDL includes an implementation of the HDF-EOS API. In addition, it should be noted that the file format underlying the current version of MATLAB is HDF 5.

Obtaining and Installing

The basic HDF library can be obtained from The HDF Group (http://hdf.ncsa.uiuc.edu/). Binaries are available for several operating systems (e.g., Windows, Mac OS X, Linux). Alternatively, source code is available for installation through the standard "configure" and "make install" (autoconf) mechanisms. HDF-EOS can be obtained from the HDF-EOS Tools and Information Center (http://hdfeos.org/).

GeoTIFF

TIFF (Tagged Information File Format) is a popular and versatile format for storage, transfer and display of raster data. The GeoTIFF metadata format is a specialization of the TIFF format that incorporates geographic information embedded as tags within the file. The geographic information allows data in the TIFF formatted file to be displayed in geographically correct locations. GeoTIFF is supported by many current Geographic Information Systems (GIS) and viewers and thus provides a level of interoperability between systems or applications.

Limitations

Many of the original GeoTIFF developers are less active in maintaining and evolving the specification and implementations, though Frank Warmerdam, the developer behind GDAL/OGR and other open source geospatial software, currently curates this format, and it seems to be regaining popularity and support.

Some popular image processing software fail to recognized the “geo” aspect and treat

GeoTIFF files as regular TIFF files. As a result the geographic metadata is lost when working (saving) files.

Proficiency Level

Intermediate - Advanced

Languages:

Implementations of GeoTIFF can be found for most popular programming languages through web searches, but a comprehensive list is not obvious.

Obtaining and Installing

Produced originally by Dr. Niles Ritter, while at the Jet Propulsion Laboratory, and other associates, the GeoTIFF format is completely open and in the public domain. Additional information, such as the format specification, library API and implementations, is available at the geoTIFF main web site, http://www.remotesensing.org/geotiff.

Two standalone programs included with most library API implementations are:

- listgeo – lists the geolocation metadata in a GeoTIFF file in an ASCII format which can be read by geotifcp.
- geotifcp – copies a TIFF (or a GeoTIFF) file and adds (or replaces) the geolocation metadata from an ASCII text file (in the format produced by listgeo), and creates a (new) GeoTIFF file.

GDAL and its utility programs "gdalwarp", "gdal_translate", and others, as well as other open source tools such as OpenEV, QGIS, and GRASS, can all handle GeoTIFF easily.

KML

The Keyhole Markup Language (KML) is an XML-based standard used to produce, modify, exchange, and most importantly display geographic data in an Earth browser such as Google Earth or NASA's World Wind. KML is now an open standard, maintained by the Open Geospatial Consortium (OGC).

What Is It Good For?

The KML specification is open source and can be used with an increasing number of online, downloadable, and mobile web mapping applications. In addition to the suite of Google products that support KML, some of the other applications that currently support KML include ESRI's ArcGIS Explorer, Yahoo!Pipes, and Adobe PhotoShop. KML files can be created from or converted to other formats such as ESRI shapefiles, using "*ogr2ogr*" within the OGR package. KML in combination with applications such as Google Earth and Google Maps allow users with a variety of skill sets to combine and overlay data on a map and share their results with others. In addition, KML can be used to animate time series data, facilitating exploration of changes over time.

Limitations

KML supports data in a limited number of formats and projections. For example, raster images referenced through KML need to be in latitude/longitude on the WGS84 datum to be properly displayed. The OGC will likely make future enhancements to the KML specifications. Therefore, these limitations are subject to change.

Proficiency Level:

Beginner – Intermediate

Languages:

Any programming language that outputs a text file can produce a KML file. The C/C++ library "OGR" can read and write the KML format.

Obtaining and Installing

KML is an XML-based language and therefore does not require downloading and installation. Users can create a KML file with any text editor by saving XML-based text with a .kml extension using UTF-8 encoding. Access the following web pages for more details about: http://www.opengeospatial.org/standards/kml/ and http://code.google.com/apis/kml/documentation/.

Examples of Use

See http://nsidc.org/data/virtual_globes/ for a listing of experimental and operational data products distributed as KML/KMZ files, including a time series of the sea ice extent in March and September from 1979 to present.

ESRI Shapefile format

A vector format originally developed by the ESRI Corporation, the shapefile format is now publicly well documented (http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf, http://en.wikipedia.org/wiki/Shapefile), and is a common format for the transfer and storage of vector information. All the popular open source geospatial software can read and write shapefiles. Such software includes GRASS, QGIS, OGR, uDig, OpenEV, gvSIG, and others. Most commercial software will similarly work with shapefiles.

A shapefile actually consists of at least 3 files, with extensions .shp (to store the geometry), .dbf (to store the attributes), and .shx (an index file). Additional files have been added and have become standard, the most important of which is the .prj file, which stores the projection information. Data are stored in the file in a structured binary format.

What's it good for?

Shapefiles can be used to compactly store and transfer vector data.

Limitations

The fact that a data set is stored in 3 or 4 separate files can be confusing. Care must be taken to keep the associated files together during transfer.

Proficiency Level

Intermediate

Languages

The OGR library (http://www.gdal.org/ogr/) has bindings for C and C++, perl (Geo::OGR module), python, and perhaps others. There is also a library called "shapelib", with bindings in various languages. Perl's relevant module is Geo::Shapelib.

Obtaining and Installing

Shapelib: http://shapelib.maptools.org/
OGR: http://www.gdal.org/ogr/

Examples of Use

Many free data sets, such as political borders, roads, municipal GIS data, as well as Earth

science data, can be downloaded from the Web in shapefile format.

Text Files

For small data sets and certain types of field data, text files may be appropriate. However, there are a wide variety of text file formats which can be used, including comma-delimited, tab-delimited, packed text, and a wide variety of XML schema particular to differing domains (such as Ecological Metadata Language (http://knb.ecoinformatics.org/software/eml/eml-2.0.1/index.html), Water Markup Language (http://his.cuahsi.org/wofws.html), or KML (described above). Some text files are appropriate for GIS applications, such as the ESRI Grid format (http://en.wikipedia.org/wiki/ESRI_grid). Simple tab- or comma-separated files are also often used, particularly by individual investigators. The ORNL DAAC maintains a document describing best practices for working with environmental and ecological data sets for archival, including suggestions for text and image formats (http://daac.ornl.gov/PI/bestprac.html). Some of these best practices, particularly in terms of file naming conventions, are applicable to other types of files. Some older data sets are stored in packed ASCII files, where each data field is in a fixed number of columns.

What's it good for?

Text files are extremely flexible and can be used to transfer a wide variety of information. Text files are human-readable, and most delimited text files can be loaded into a wide variety of tools, including Excel, Matlab, and R. Text files are not subject to binary file limitations (such as byte ordering). The human readability makes verifying the correct reading of the file into a tool relatively straightforward. Forward compatibility is generally not an issue, in that there is an extremely high likelihood that any future tool will be able to read a delimited text file, particularly if the file is in ASCII encoding.

Limitations

A text file often contains very limited metadata, if any, requiring that the metadata be in one or more separate files. Often, the first row of a text file will contain column names (for delimited text files), but not information about units or method of collection, for example. Coordinate information may be present in the file, where each row relates to a particular location, but there is no standard method of representation, nor is there any way to include reference datum or projection. For more complex data sets, the data may need to be spread across multiple text files, complicating the process of loading and working with the entire data set. In most cases, a text file will require more disk space than a binary file, which can be an issue for very large data sets. Dates can be represented in a variety of formats, and without some sort of metadata, it can be difficult to determine whether 10-08-2005 refers to the tenth of August or the eighth of October.

Proficiency Level

Beginner to intermediate.

Languages

Virtually all common programming languages provide access to text files, and most (e.g., Perl, Python, Java,…) have additional modules for parsing such text formats as comma-separated-value (CSV) and XML.

Examples of Use

Many ecological and environmental data sets are stored in delimited text files.

OPeNDAP

The Open-source Project for a Network Data Access Protocol (OPeNDAP) is a set of standards and protocols for transmitting data over the network over HTTP. It has support for server-side subsetting to reduce the amount of data that need be transferred.

What Is It Good For?

OPeNDAP is particularly good for providing data to one of the many clients that have been enabled to read OPeNDAP files as if they were local. These include GrADS, Ferret, Live Access Server (which uses Ferret), Integrated Data Viewer, McIDAS-V, IDL and Matlab.

Limitations

In its default configuration, OPeNDAP does not by itself provide spatial semantics, i.e., the ability to recognize which fields represent spatial coordinates; as such it does not provide server-side spatial subsetting (the client must supply the indices of the arrays desired). Also, all data served by OPeNDAP must be online.

For similar reasons, any CF1-aware clients (e.g. IDV and GrADS) will not be able to directly use OPeNDAP-served datasets directly, unless they are stored in NetCDF or HDF5.

Data Formats Supported

NetCDF, HDF-4, HDF-5, GRIB, flat binary in Hyrax (see below). Additional handlers can be added by experts. Other implementations may support additional formats.

Implementations

Several implementations of the OPeNDAP server are available. The newest arrival is Hyrax, available from opendap.org (http://www.opendap.org/). The GrADS Data Server (GDS) marries the analysis power of GrADS with the serving capabilities of OPeNDAP (http://grads.iges.org/grads/gds/). Dapper is available from PMEL and is particularly adept at handling in situ data (http://www.epic.noaa.gov/epic/software/dapper/). And pydap is an all-Python implementation that includes a Python OPeNDAP client as well (http://pydap.org/).

Proficiency Level:

Intermediate

Obtaining and Installing

Hyrax is available from http://www.opendap.org/download/hyrax.html and comes in two parts. The Back End Server (BES) installs with GNU configure. The OPeNDAP Lightweight Front-end Servlet (OLFS) installs into a Tomcat server as a simple "war" file. See the previous section for the sources of other OPeNDAP servers.

Examples of Use

To “browse” OMI DOAS Ozone data through OPeNDAP:
http://acdisc.gsfc.nasa.gov/opendap/Aura_OMI_Level3/OMDOAO3e.003/

If accessing a specific file from a client, like GrADS, navigate to the desired file and copy the Data URL from the OPeNDAP form, e.g.:

```
macwork:~> gradsdods -l
Grid Analysis and Display System (GrADS) Version 1.9b4
Copyright (c) 1988-2005 by Brian Doty and IGES
Center for Ocean-Land-Atmosphere Studies (COLA)
Institute for Global Environment and Society (IGES)
GrADS comes with ABSOLUTELY NO WARRANTY
See file COPYRIGHT for more information

Config: v1.9b4 32-bit big-endian readline sdf/xdf netcdf dods dods-stn lats
printim

Issue 'q config' command for more information.

GX Package Initialization: Size = 11 8.5
ga-> sdfopen
http://acdisc.gsfc.nasa.gov/opendap/Aura_OMI_Level3/OMDOAO3e.003/2008/OMI-
Aura_L3-OMDOAO3e_2008m0818_v003-2008m0820t020302.he5
Scanning self-describing file:
http://acdisc.gsfc.nasa.gov/opendap/Aura_OMI_Level3/OMDOAO3e.003/2008/OMI-
Aura_L3-OMDOAO3e_2008m0818_v003-2008m0820t020302.he5
SDF file has no discernable time coordinate.
SDF file
http://acdisc.gsfc.nasa.gov/opendap/Aura_OMI_Level3/OMDOAO3e.003/2008/OMI-
Aura_L3-OMDOAO3e_2008m0818_v003-2008m0820t020302.he5 is open as file 1
LON set to -180 179.992
LAT set to -90 90.0023
LEV set to 0 0
Time values set: 1:1:1:0 1:1:1:0
ga->
```

Web Map Server

A Web Map Server provides pictures of data for a specified spatial area to a client over HTTP using the Web Map Service (WMS) protocol from the Open Geospatial Consortium (OGC). In addition to spatial subsetting, other options such as varying styles, time coverages, etc. may be supported by some servers.

What Is It Good For?

WMS is good for serving data to the general public and other non-researchers such as policymakers or applications users. It can also be useful for browsing data. Many clients can make WMS requests, including GoogleEarth and WorldWinds. Servers hosting Web mapping applications can use WMS to combine map layers from many servers across the Web and present them in one interactive map.

Limitations

WMS does not provide numeric data values, so in general it is not suitable for science research. Also, all data served via WMS must be online.

Implementations

MapServer from the University of Minnesota (http://mapserver.gis.umn.edu/) and GeoServer are both open source implementations of WMS. A number of commercial servers are available from companies such as ESRI and Ionix. See the OGC web site http://www.opengeospatial.org for more information.

Proficiency

Intermediate. O'Reilly has published a book, *Web Mapping Illustrated*, detailing how to setup and configure MapServer, among others.

Obtaining and Installing

MapServer can be obtained from the University of Minnesota (http://mapserver.gis.umn.edu/). It can be installed from source using GNU configure, and some precompiled binaries are available, including Linux RPMs and Mac OS X .dmg files. It is included in the popular collection of various open source geospatial tools called "FWTools" (http://fwtools.maptools.org/).

Examples of Use

To access the XML capabilities document at UCL:
http://iceds.ge.ucl.ac.uk/cgi-bin/icedswms?SERVICE=WMS&REQUEST=GetCapabilities

To access a particular layer, e.g. DMSP Nightime Lights:
http://iceds.ge.ucl.ac.uk/cgi-bin/icedswms?
VERSION=1.1.1&SERVICE=WMS&REQUEST=GETMAP&
SRS=EPSG:4326&BBOX=-180,-90,180,90&WIDTH=1000&HEIGHT=500
&LAYERS=lights&STYLES=&FORMAT=image/png&
BGCOLOR=0x000000&TRANSPARENT=FALSE&
EXCEPTIONS=application/vnd.ogc.se_inimage

Web Coverage Server

Like WMS, the Web Coverage Service was developed by the Open Geospatial Consortium. WCS allows the numeric data to be transmitted over HTTP in a variety of formats (or "profiles"), including HDF-EOS, NetCDF/CF-1 and GeoTIFF.

What Is It Good For?

WCS is especially useful for transmitting spatial subsets of horizontally gridded data. As such, it excels for machine-to-machine interaction. Non-gridded data, such as satellite swaths, can also be served but must be reprojected first to a regular grid, either beforehand or on-the-fly.

Limitations

WCS is not well suited for vertical profile data. Use of WCS for long time series can be problematic depending on the implementation of the server.

Clients that can make WCS requests are limited, though more are coming online every day. Also, note that the WCS response is only as usable as the client's ability to read the particular format (or profile). Thus, an HDF-EOS profile, for example, is useful only to clients that can interpret HDF-EOS format data properly.

Implementations

MapServer has some limited support for WCS, as GeoTIFF profile.

THREDDS Data Server supports WCS access to netCDF/CF-1 data.

Some commercial tools also provide support, as do a number of custom servers in the community.

Proficiency:

Advanced

Obtaining and Installing

MapServer can be obtained from the University of Minnesota (http://mapserver.gis.umn.edu/). It can be installed from source using GNU configure, and some precompiled binaries are available, including Linux RPMs and Mac OS X .dmg files. It is

included in the popular collection of various open source geospatial tools called “FWTools” (http://fwtools.maptools.org/).

THREDDS Data Server can be obtained from NCAR

Examples of Use

Access GeoTIFF from GMU’s GeoBrain WCS Server:
http://geobrain.laits.gmu.edu:81/cgi-bin/wcs110?service=WCS&version=1.0.0&request=getCoverage&coverage=GEOTIFF:"/Volumes/RAIDL1/LANDSAT/WRS2/ETM/p179/p179r051_7x20021106.ETM-EarthSat-Orthorectified/p179r051_7k20021106_z34_nn62.tif":Band&crs=urn:ogc:def:crs:epsg:6.3:4326&bbox=23.0331,12.5291,24.0093,13.0999&format=image/geotiff&width=451&height=399

Web Feature Server

Web Feature Server (WFS) is an Open Geospatial Consortium (OGC) protocol in the same family with WMS and WCS designed to facilitate the transmission of vector data.

Data Interchange Tools

There are existing tools available that help hide the complexity of working with disparate data formats and structures. Some of these tools are available as software libraries that can be utilized or incorporated by other applications and scripts. A couple examples of these types of tools are: (1) the Earth Science Markup Language (ESML) and (2) the Geospatial Data Abstraction Library (GDAL).

Earth Science Markup Language (ESML)

The Earth Science Markup Language (ESML) is not another data format, but rather provides a metadata-based solution for decoding science data formats. The three primary components of the ESML solution include the schema (grammar for the ESML description file), the ESML description file(s) providing the structure and semantics of a science data format, and the ESML software library enabling applications to parse the description and ultimately retrieve data from files of varying formats.

What Is It Good For?

EMSL provides the capability for applications to interoperate with multiple exiting data formats and structures without the overhead and complication of translating data to a new format. ESML is particularly well-suited for handling legacy data sets available in many scientific data archives. The ESML tools handle supports many known scientific data formats, such as HDF, netCDF, Grib, etc., but can also readily supports user-defined ASCII and binary structures as well.

Limitations

Only data reading is supported with the current version of the ESML libraries. This is the primary purpose and focus of the technology, but there have been requests for write capabilities as well. Only C++ libararies are currently available.

Implementations

The technology has been incorporated to a number of other research efforts and applications dealing with heterogeneous data issues. Related usage information is and community interactions are available at the projects web site (http://esml.itsc.uah.edu).

Proficiency:

Advanced. These tools are primarily targeted for application developers or science researchers who write their own software programs or scripts.

Obtaining and Installing

ESML was developed at the Information Technology and Systems Center of the University of Alabama in Huntsville. ESML is available as an Open Source project on SourceForge, available at https://sourceforge.net/project/showfiles.php?group_id=72129 .

Geospatial Data Abstraction Library (GDAL/OGR)

(From Wikipedia) GDAL is a library for reading and writing raster geospatial data formats, and is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation. As a library, it presents a single abstract data model to the calling application for all supported formats. It may also be built with a variety of useful command-line utilities for data translation and processing. Its companion library, OGR, is a library for reading and writing vector geospatial data formats.

What Is It Good For?

GDAL and OGR are cross platform C++ translator libraries for raster and vector (respectively) geospatial data formats. They present a single abstract data model to the calling application for all supported formats. They also come with a variety of useful command line utilities for data translation and processing.

Limitations

The library is specific for geographic data.

Implementations

Several geographic software packages, such as Map Server, ArcGIS, GRASS, OpenEV, and QGIS, currently use GDAL and OGR to support read/write of common formats. FWTools combines many of these open source tools into a single package.

Proficiency:

Command line tools: Intermediate
Building software to use the libraries: Advanced

Obtaining and Installing

(From Wikipedia) GDAL was primarily developed by Frank Warmerdam until the release of version 1.3.2, when maintenance was officially transferred to the GDAL/OGR Project Management Committee under the Open Source Geospatial Foundation.

Examples of Use

MapServer uses GDAL to transform data files into WMS pictures on the fly.
The utilities “ogrinfo” and “gdalinfo” print metadata, including geolocation information, about specified files. The utilities “ogr2ogr” and “gdal_translate” can be used to convert from one vector or raster format to another. The utilities “ogr2ogr” and “gdalwarp” can reproject data sets.