



January 14, 2009

MEMORANDUM FOR: See Distribution

FROM: Alfred M. Powell, Co-Chair
Satellite Product and Service
Review Board (SPSRB)

Kathy Kelly, Co-Chair
Satellite Product and Service
Review Board (SPSRB)

SUBJECT: SPSRB Policy on NetCDF

This policy memo states the NESDIS Satellite Product and Services Review Board's (SPSRB) requirement to adopt NetCDF4 file format using the Climate and Forecast metadata conventions as the standard for the intermediate file format for all STAR and OSDPD product systems. This will allow the development of an enterprise-wide one-to-many file format converter (e.g., NetCDF to BUFR or GRIB, etc.), allowing reuse between many software development groups, reducing both development and maintenance costs incurred by STAR and OSDPD. A brief overview of NetCDF can be found in the attachment.

It is proposed that all FY10 new start projects implement the requirement that intermediate files from operational STAR product systems conform to the NetCDF4 format and use the CF metadata conventions. In addition, currently funded projects having preliminary design reviews after August 2010 will also implement the NetCDF4 requirement. This allows product teams to include funding for code to write NetCDF4 files and employ CF metadata in their initial project plans.

Existing product teams may be able to implement these changes on a "best effort" basis and are not required to implement the changes if found onerous. Existing product teams should, however, be aware that code-generators are available from the STAR/SMCD Integration Team website¹ to create functions in C or FORTRAN to read and write NetCDF files (NetCDF4 files when linked to the NetCDF4 library during compilation). The use of these existing code generators may be a quick and relatively painless manner in which these changes could be implemented even without funding.

The implementation of the file format and metadata conventions is expected to be included in both preliminary and critical design reviews and will be checked during the code reviews which are proposed to be part of the SPSRB-managed transition to operations process during the 2010 timeframe.

¹ <http://www.star.nesdis.noaa.gov/smcd/spb/sait/sait.php>



Distribution

All personnel in STAR and OSDPD

cc: SPSRB Principals

Brief Overview of the NetCDF4 File Format.

Maurice McHugh

IMSG @ NOAA/NESDIS/STAR

Summary: Over the past decade heritage ASCII and binary file formats have been replaced by modern file formats, including NetCDF and HDF, which offer many advantages and eliminate the main deficiencies of the heritage formats. NetCDF4 is a machine-independent, self-describing file format designed to store and exchange scientific data. This file format is platform-independent and can be easily read on expensive IBM AIX boxes or by individual Linux computers regardless of byte order. Datafiles conforming to the NetCDF4 standard are self-describing and contain sufficient attributes (i.e. metadata elements) to allow the user to understand the nature of the data being accessed.

NetCDF4

NetCDF¹ was developed at the University Corporation for Atmospheric Research (UCAR) as a self-describing and platform-independent specifically for scientific file format. HDF5, the principal competitor to NetCDF has been designed for many user communities and the additional flexibility and capability of HDF5, while useful, results in a format and software that is far too complex and hard to use for the simple task of storing 2- or 3-dimensional data fields as required by the remote sensing or meteorological communities.

Data stored within NetCDF-compliant files are stored as rectangular arrays, not ragged arrays; this makes NetCDF ideal for storing spatial fields or 3-dimensional data cubes as are often produced through analysis of satellite data or output from meteorological or climate models. A second advantage of the regularity of data stored within the NetCDF model is that data fields can be subset efficiently without requiring that the entire file be first read – only the array elements to be subset are accessed. While entire NetCDF files are not read as quickly as binary files may be, very fast subsetting, attached metadata, and platform-independence are massive advantages offered by NetCDF over the binary format. Data can be added to a NetCDF file without affecting other data within the file, or requiring new software to be written to read that data. Third-party groups have produced software that can manipulate, edit, read and display data stored in NetCDF files such as the NetCDF operators, IDL, MATLAB and many more.

For NetCDF files to be accessed the NetCDF library must be installed, the library offers application programming interfaces (APIs) for FORTRAN, C/C++, Java, and other languages. This allows programmers to write code to read, write or modify the contents of NetCDF files using a familiar programming language. The function calls within the NetCDF API are powerful, elegant and easy to use, with NetCDF files created and populated with data by a relatively simple program.

Recent improvements in the NetCDF specification, from NetCDF3 to NetCDF4², allowed for several significant improvements in the NetCDF model. Significantly, data within NetCDF4 files can be stored in a HDF5 data layer, data can be compressed using the SZIP algorithm, and extensions were made to NetCDF's data modeling abstractions, and parallel input/output (I/O).

¹ <http://www.unidata.ucar.edu/software/netcdf/>

² http://www.unidata.ucar.edu/software/netcdf/docs/faq.html#changes_4.0

One of the most important features of the API implementation for the new NetCDF4 model is that function calls are compatible between NetCDF3 and NetCDF4. The importance of this is readily apparent when one considers that code written to manipulate NetCDF3 files can be reused without modification to perform the same actions on NetCDF4 files. Only a recompilation of the existing software while linking to the NetCDF4 libraries is required. Only when implementing the enhanced functionality of the NetCDF4 model will modifications to existing code be required, and even then many of the modifications may occur in the form of optional arguments appended to existing functional calls.

Recent developments in streaming technologies has resulted in software such as OPeNDAP³ and THREDDS⁴ serve NetCDF files to remote computers across the internet or through a network. These remote files can be subset, concatenated and manipulated by the client computers before being downloaded.

The metadata conventions that should be followed when writing data to files are the Climate and Forecast (CF) conventions⁵ which allow for flexible and meaningful metadata elements to be directly attached to variables as attributes stored in NetCDF files. This elegant approach contrasts to the complex metadata conventions used by the NPOESS HDF5 file format in which some metadata is stored in the datafile, and other metadata stored in an external xml file that should accompany the data.

The ease of use and many advantages of using NetCDF4 as an intermediate file format using the CF metadata convention within operationally used STAR product systems is proposed by the STAR data management committee and was briefed to the SPSRB by Ingrid Guch and Maurice McHugh.

One caveat that must be considered in the use of NetCDF4 as a file format standard is the current lack of support from the developers of the Interactive Data Language (IDL), which is widely used within both STAR and OSDPD for visualization and product monitoring. The IDL developers are aware of NOAA's interest in NetCDF4 support but at this point they do not plan on supporting that format in the next release of IDL in early 2009; nor have they made any firm plans to support NetCDF4 in the release after that, presumably in 2010. Several STAR scientists have requested that STAR management ensure that IDL developers provide support for the NetCDF4 format as a matter of urgency.

While IDL works on implementing support for NetCDF4 several options are available to STAR and OSDPD developers to mitigate this. For example, the internal binary files could be read directly by existing IDL code instead of converting to NetCDF4; or it may be possible that NetCDF4 files be read using IDL's existing HDF5 interface; NetCDF4 data could be passed to IDL through IDL's OPeNDAP interface; other workarounds are available also. Regardless, it is only a matter of time before IDL supports NetCDF4 internally.

³ <http://www.opendap.org/>

⁴ <http://www.unidata.ucar.edu/projects/THREDDS/>

⁵ <http://cf-pcmdi.llnl.gov/>