

Recent Changes to NetCDF Development Processes

Ed Hartnett
Unidata/UCAR

NetCDF Workshop
July 25 – 26, 2011

Why Change Our Processes?

- To better cope with multiple engineers working on the code at the same time.
- To provide better visibility to outside stakeholders who want to follow our progress more closely.
- To (hopefully) improve quality of product, and productivity of engineers.

Switch to Subversion

- Historically netCDF-C/Fortran/C++ used cvs, a revision control system from the last Ice Age.
- We switch to subversion due to the fact that it is just like cvs but without the annoying parts.
- Our repository is available (read-only) for all:
<http://svn.unidata.ucar.edu/repos/netcdf/trunk/>
- This is what is made into the daily snapshot each night around midnight (if it passes a distcheck test on our main machine).

Documentation Issues

- We are currently using venerable texinfo system, developed in pre-web days (though capable of web output).
- Texinfo is a non-WYSIWYG editing format that produces multiple outputs (postscript, html, info, etc.)
- The documentation is separate from the code, and there is no checking between documentation and code for mistakes.
- Man pages are separate from the other documentation

More Documentation Issues

- There is no internal documentation (only the public APIs are documented).
- Some repetition between C and Fortran manuals.
- We don't have any diagrams or pictures in the documentation – text only!
- Web output looked old-fashioned back in 2004 when we started using texinfo.
- Complex build required.
- Requires that editor of documentation know where it is, and about texinfo.

Some Documentation Requirement Relaxations

- HTML documentation is the only required output format.
- Fortran users are going to have to read the C manual sometimes (but we can make sure it's understandable to non-native C speakers).

Advantages of Doxygen

- Eliminates separate man page documentation.
- Puts documentation near code.
- More intuitive for programmers to use.
- Checking of function names and parameters when documentation is generated.
- Easily enables internal documentation.
- Good support for images.
- Automatic hyperlinking.
- Access to the code itself in documentation.

More Advantages to Doxygen

- Familiar to most programmers (compatible with javadoc).
- Nice looking output.
- Base tool (doxygen) much more widely installed than old texinfo system.
- Much simpler documentation generation process for netCDF build system.
- Each function gets linking man page.

Some Disadvantages with Doxygen

- Clutters up code. (Can we really complain about comments as clutter?)
- Some code-neutral things, like position of function in file, now become significant for developers.
- Doxygen makes old-fashioned code-reading less nice.

WARNING: ADVANCE PREVIEW!

- The final documentation may have a different appearance, based on stylesheet and other visual changes.
- Organization of sections under construction.
- Doxygen conversion is mostly complete for netCDF Tutorial, Installation Guide, User Guide, and C Reference Manual. Only the documentation of the tools (ncdump/ncgen/nccopy) lags behind.

Things to Note

- Examples are nice-looking and documented functions are hyperlinked.
- Common parameter explanations are centralized and hyperlinked.
- Error codes are hyperlinked.
- Netcdf.h file becomes useful as documentation.

Remaining Documentation Challenges

- Man pages look pretty poor at this point.
- The documentation process tends to highlight “spills” in the code that need to be cleaned up.
- Non-book based reorganization of the material is more necessary than I first thought.

What Will be In 4.2 Release

- The C API and other converted manuals will be ready for the 4.2 release in some reasonable form.
- The tools may maintain their own man page documentation for the 4.2. release until I figure out how to get doxygen to do better man pages for me.

Using Jira

- We are heavily using the Jira tool to track our work.
- Jira (along with associate tools) provides lots of support for agile programming methodologies.
- We are so expert at agile programming that we don't know any of these methodologies. So we are learning the jargon.



Project Summary Page

- Take a look at netCDF-C summary page for information about what we are doing:

<http://www.unidata.ucar.edu/jira/browse/NCF>

Notes About Jira Use

- We are still adding lots of issues from previous planning documents.
-