User Support Goals For DC 2

James Peachey

GSFC/L3
Goals In Major Support Areas

• **Documentation**
  – Help synopsis for Science Tools ("fhelp" output)
  – Detailed tutorials/manuals for individual Science Tools
  – Analysis threads/cookbooks which show how tools interact

• **Installation and Runtime Environment**
  – Simple, robust installer(s) for Linux and Windows
  – Revive the “Install Area”
  – Simple, robust init script

• **Windows Support**
  – Increase activity on Windows
  – Stabilize RM

• **OSX Support?**
  – Interest in this is growing
  – Clearly at least a “B” priority
  – Proof of concept underway
- Off to a good start; efforts in the last two months have been particularly fruitful.
- Chuck Patterson at SLAC is coordinating with Seth, David, and developers (a.k.a. “Subject Matter Experts”).
- Anyone can browse current version of docs under development at [http://glast-ground.slac.stanford.edu/documentation](http://glast-ground.slac.stanford.edu/documentation) (password needed for modification access).
- Small variations in format and style are being reconciled.
- Still under consideration:
  - Whether to, and if so, how to automate generating hard copies, and/or other changes to form of docs from a master file.
  - How to automate extracting summaries of the detailed manuals to produce “fhelp” synopses for Science Tools.
Linux Installation and Runtime Environment

- RM and Navid’s installer are stable and work well. Tony Johnson has also implemented an installer in Java, which interacts with RM in a similar way.

- **How Navid’s installer works:**
  - Last step of RM is to add the current configuration/tags to a database
  - Installer reads database, determines what packages and versions are needed, downloads and installs them

- Currently runtime environment is set by wrapper scripts which are installed in a central bin/ directory; the wrapper scripts set up the environment, then run the actual executables.

- This runtime scheme works, but there are path length issues due to CMT’s decentralized structure.

- Decentralization also leads to a proliferation of environment variables.
Status Of Windows Support

- Manual builds on Windows work.
- The installer (same script as on Linux) works, but there is no equivalent to the Linux wrapper scripts, so it is not convenient to run the tools. CMT is therefore still needed at runtime.
- Windows builds run by RM have problems detecting failures (false negatives). No pattern has been understood yet, but Navid believes it may be caused by an interaction with LSF.
- There have been path length problems on Windows too.
- A non-technical point: we need more Windows users, particularly for the Science Tools checkup 3!
Why Revive The Install Area?

- For both Linux and Windows, using central Install Area will help achieve a simple and more robust installation process and a more rational runtime environment.
- CMT leaves build products and ancillary data files distributed throughout the source tree; the aforementioned environment variable length problems result from adding many individual directories to PATH, LD_LIBRARY_PATH, PFILES, etc.
- Setting up the environment becomes odious, and the most straightforward way to do it is to use CMT, but this means CMT is needed in order to run the tools, even if binaries are installed.
- Centralized installation is what most users expect and prefer.
- Number of environment variables will be greatly reduced.
- Initialization and maintenance of correct runtime environment will be more easily accomplished.
Install Area Issues

- Difficulty with GlastRelease: centralized organization of some ancillary files will be tricky, particularly for jobOptions files.
- Many packages read environment variables directly, making it difficult to eliminate them without inconveniencing developers.
- Proposed plan is to transition *gradually* to using an Install Area.
- Start with Science Tools, which will minimize impact on Glast Release.
- Converting Science Tools will be easier because there is less code and fewer developers are involved.
- Support for Science Tools is also more critical because it will eventually have a wider user base; in particular, for checkup 3, it is hoped that many users will try the software.
How Do We Get There?

- Already have ability to install libraries in central lib/ directory, thereby reducing length of LD_LIBRARY_PATH.
- Next step is to install parameter files (for Science Tools) in a central pfiles/ directory, thereby reducing length of PFILES.
- Adding an installed data/ directory will then reduce the number of environment variables.
- Final step would be to install the “real” executables in a central location, allowing complete separation of source tree from installed tree.