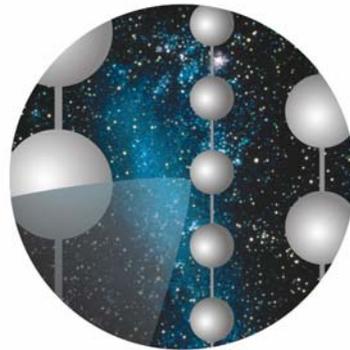


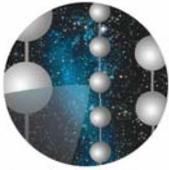
IceCube Offline Software Tools

Erik Blaufuss
University of Maryland



I c e C u b e

Reconstruction/Offline Software Workshop
Uppsala Collaboration Meeting
October 9, 2004

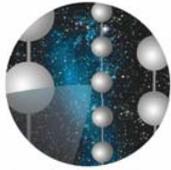


IceCube

Software tools



- Collection of software tools and packages provided to aid in software development
- Includes:
 - External libraries of known version
 - Build system executables
 - Documentation system executables
 - Versioning system/workspace management tools
- I won't be listing each one here, just highlighting a few of the important ones.



IceCube

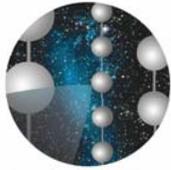
How to get started...



- The ONE thing to take away from this talk:

<http://glacier.lbl.gov/offline>

- Your one stop link collection for all IceCube Offline Software.
 - Introduction and Overview
 - Help getting started, compiling
 - Details on software packages
 - Updated often.

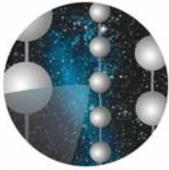


IceCube

Tools you need to supply



- Some tools are too large and complicated to include in a tools distribution.
- These include:
 - ROOT (root.cern.ch)
 - Need to use ROOT v4 (Current: v4.00/08)
 - Java (java.sun.com)
 - Need version 1.4.2
- Set proper environment variables and paths
- See Offline software guide for all details

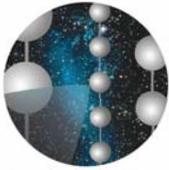


IceCube

Obtaining tools



- You will need an account on glacier.lbl.gov
 - Contact Martin Stoufer (MCStoufer@lbl.gov)
- A single script needed to get tools
 - Copies needed libraries from tools repository on glacier.lbl.gov
 - Installs and properly configures tools in your local tools installation
 - STILL need to be run, even when working on glacier!
- Get and run “updatetools.sh” script.
 - This will create a “tools” directory and fill it.

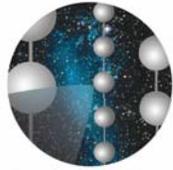


IceCube

BFD



- Baseline File Development tool
- A workspace and release system manager
- Workspace setup:
 - bfd init will setup a workspace for development and software building
 - Install links to appropriate tools
 - setup.[c]sh file to set environment variables correctly
 - Needs to be done **ONCE** per workspace
 - Make sure done in a clean shell (no setup scripts run)
 - Intended to provide a uniform, repeatable build environment

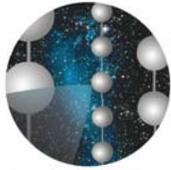


IceCube

BFD Projects/Metaprojects



- BFD is also a power front-end to CVS.
- In addition to workspace mangmt, BFD allows:
 - Checking out (tagged) projects and metaprojects
 - Delivering of tagged project and metaprojects
 - Creating new projects
 - Compile and build tarball packages for deployment
- Automatically handles dependencies across projects in a metaproject



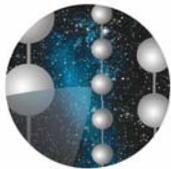
IceCube

Projects & Metaprojects



- A project is the smallest piece of software needed to perform a task
 - An IceTray Module would be a natural example.
 - Each project lives as an independent entry at the top of the CVS repository.
 - Each project can be tagged with a release identifier to track it independent of all other projects.
 - A project will often depend on others



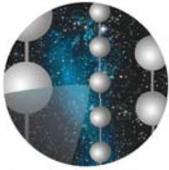


IceCube

Projects & Metaprojects (2)



- A Metaproject is a logical collection of projects.
 - Each project included in the metaproject can have a unique tag
 - No longer need to add tags across the entire CVS archive!
 - Recursive: A metaproject can also be a logical collection of metaprojects.
- Powerful tool for distributing complex collections of code.
- First release of offline software is metaproject: DATACLASS-APP
 - `bfd co -r V01-00-00 DATACLASS-APP`

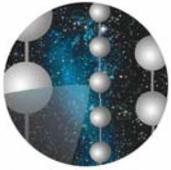


IceCube

Build system



- The build system provides control over:
 - Building projects
 - Making new projects
 - Workspace management (cleaning, etc.)
 - Running tests.
- Currently a great deal of work is going into making the build system more robust.
 - Keep an eye on the offline software website for the latest details:
<http://glacier.lbl.gov/offline>

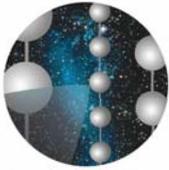


IceCube

Code testing tools



- We want ALL code used in Offline software to be fully tested.
 - Unit testing needs to be included with a project.
 - Fully test project functionality with an independent testing suite.
 - Tests can be repeated with each new release of software to ensure proper functionality.
- Using the TUT testing libraries (part of tools)
 - Provides uniform testing environment
 - Calls attention to failed tests.



IceCube

Documentation



- DOXYGEN used as our documentation system
 - New projects are created with a place to put project specific documentation:
Project/resources/docs/
 - Any documentation files (.dox) put there will be globally built and linked into the documentation system.
 - Powerful system merges:
 - User written docs
 - Doxygen auto-generated class maps and docs.