

Repository and Releases

Status and Plans

ATLAS Software Workshop

17.03.99

M. Stavrianakou

Overview

- Packages
- Supported Platforms
- Releases
- Tools and Quality assurance
- Miscellaneous

PACKAGES

- **In the repository (under \$CVSROOT/offline)**
 - **Production (TDR) software:** 25 top level packages mainly in F77 and Age
 - **Domain software:** 16 top level packages, mainly in C++
 - **Applications:** top level package for various applications across domains (for the moment used for dice and atrecon executables, job scripts and datacards)
 - **Doc:** top level package for documentation
 - **Contrib:** top level package for contributed software: 1 package in F77 and C++
 - **External:** top level package for external software: 2 packages in C++

Lines of code: F77 480 K Age 93 K
 C++ 269 K (compared to 197 K in Dec. 98)

- **In release (under \$SRT_DIST/current)**
 - 40 top level packages (106 packages total)

PLATFORMS

- Supported
 - HP, DEC, IBM, LINUX and SUN
 - hppa1.1-hp-hpux10.20 with HP aCC 1.18
 - alpha-dec-osf4.0b with Digital C++ 6.0
 - powerpc-ibm-aix4.1.5.0 with IBM xlC 3.1.x (moving to aix 4.3)
 - i586-pc-linux-gnu with egcs 1.1.1
 - sparc-sun-solaris2.5.1 with SUN CC 4.2
- Not (fully) supported
 - SGI: partly supported by Boston (mainly for production), low priority
 - WNT: to be decided...

RELEASES (1)

- Fortnightly (developers') releases
 - so far averaged 20 releases (HP, DEC, IBM, LINUX) in ~10 months
 - aim at weekly releases?
- Nightly builds
 - cron jobs update to the head, build from scratch on 5 platforms (HP, DEC, IBM, SUN, LINUX) and produce “reduced” build logs using the SRT make-chopper utility
 - TO DO:
 - Improve readability of build logs
 - Advertise build results to developers concerned

RELEASES (2)

- Release frequency and release procedure
 - maybe fortnightly releases suffice
 - even so, the procedure could be simplified and sped up by
 - encouraging package coordinators to make pragmatic decisions on supported platforms (a la ATRIG)
 - improving package structures and dependencies to allow or facilitate “reverting” to earlier working versions
 - using nightly builds for early debugging
 - we’ll soon see, so far the head is mostly broken)
 - easing the developers’ job
 - WWW based GUI for building on supported platforms automatically and remotely - in preparation by L. Tuura?

RELEASES (3)

- releasing some packages independently or using binaries from previous releases
 - debatable and not (yet?) supported by the tools
- building on local disks and/or dedicated machines for initial steps
 - some speed-up (not yet worth the effort...)
- building on fastest platforms first
 - some speed-up but in the end you do have to wait for the slower ones...
- working incrementally at the preparation stage
 - in practice it saves very little time at a considerable risk hiding previous failures and incorrect dependencies; a “clean” start has so far been found indispensable...
- sharing partial or full support for releases on some platforms with other institutes
 - to be investigated

Tools and Quality Assurance

- **Browsers**
 - cvsweb (latest version installed by C. Onions)
 - Bonsai (installed by R. Iles and L. Tuura)
- **Dependency Grapher (J. Hrivnac)**
 - new version displaying using as well as used packages to be installed
- **CodeCheck, Insure++, Purify and Logiscope**
 - no clear policy yet on who runs what and when...

Miscellaneous

- **SPIDER SRT project**
 - common glossary and work model for a Software Release Tool to be used for requirement consolidation, to be followed by evaluation of existing implementations, adoption of baseline solution and build plan
- *see S. Fisher's presentation on Friday*
- **Manpower and sharing of responsibilities**
 - work in progress at LBL (*C. Tull et al.*)
- **Collaboratory tools**
(*e.g. electronic notebook, see presentation at AWWC on Tuesday*)
 - may facilitate collaboration of librarians and developers and coordination of releases (*to be investigated with LBL*)