ATLAS Offline Release Process

David R. Quarrie

LBNL

DRQuarrie@lbl.gov

ATLAS CVS Structure

- Single CVS repository for
  - ATLAS Offline Release
    - Offline software and trigger algorithms
  - ATLAS Production system
    - Python job submission system
  - User repositories
    - For individual users
  - Group repositories
    - For physics working groups
  - High Level Trigger infrastructure
    - Glue for using offline software in online trigger environment
- Other CVS repositories for
  - Gaudi
    - Common framework shared with LHCb and others
  - ATLAS Online and Trigger/Data Acquisition
- Focus of this talk is the ATLAS Offline Release

ATLAS Offline

- Approximately 1124 CVS modules (packages)
  - 152 containers
  - 971 leaf
    - Contain source code or act as glue to external software
- Container hierarchy for commit & tag management
  - Hierarchy by detector
    - E.g. Inner Detector, Muon Spectrometer
    - Each detector has containers for e.g. Simulation, Reconstruction, Reco EDM, etc.
  - Hierarchy by subsystem
    - E.g. Control (framework), Database
  - Hierarchy by processing stage
    - E.g. Simulation, Reconstruction
- Single container hierarchy cannot adequately reflect overall structure

CMT & Tags

- ATLAS uses CMT for configuration management

- Each package has a *requirements* file that specifies
  - Entities (libraries, plug-ins, applications) built from sources within this package
    - Can include e.g. Reflex dictionary generation
  - Other packages that this package depends upon (e.g. for header files, libraries)
  - Information exported to clients (header files, libraries, configuration variables)
  - Run-time environment variables
  - Special make targets (e.g. for building and running component tests

- Some packages act as interface/glue to external software

- Each package tagged:
  - Pkg-ii-jj-kk (e.g. StoreGate-01-02-03)
    - Major, minor and patch ids
    - Non backwards compatible change to API implies change to major id
    - Internal restructuring implies change to minor id
  - Can also have bug-fix ids for bug-fix release branches (see later slides)
    - E.g. StoreGate-01-02-03-04

David R. Quarrie: ATLAS Offline Release Process

ATLAS Offline Release Types

- Nightly builds
    - Complete release built from most recent tags for each supported platform
    - Kept in cycle of typically 7 so that developers have time to diagnose and fix problems
    - Managed by NICOS (Nightly Control System)
- Developer builds (e.g. 11.x.0)
    - Stability or convergence builds every 3-4 weeks
    - Limited testing, no patches created
- Production builds (e.g. 11.0.0)
    - Every 3-6 months, generally associated with a major milestone
        - E.g. data challenge, test beam, physics workshop
    - Extensive testing and validation
- Bug-fix builds (e.g. 11.0.1)
    - Bug fixes to production builds resulting from problems uncovered during more extensive production running

David R. Quarrie: ATLAS Offline Release Process

Release Testing & Distribution

- Two test scaffolds
  - ATNight
    - Tests run in context of nightly builds as part of NICOS build procedure
    - Restricted statistics; no Grid access
    - ~70 unit and integration tests run
  - Run Time Tester (RTT)
    - More extensive tests on cluster at UCL (other instances being created)
    - ~200 tests on 24 packages
    - Grid access underway

- Distribution kits created for development, production and bug-fix builds
  - Using pacman caches
  - Currently only binary kits available - source kits available this month
  - Goal is to use distribution kit for CERN installations as well as remote sites

- Extensive "KitValidation" test suite
  - Includes all processing stages (generators, simulation, reconstruction, etc.)
  - Components from this run both on ATNight & RTT test scaffolds

David R. Quarrie: ATLAS Offline Release Process

http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/dist/nightlies/global/ · Google

ATLAS ▾ Experiments ▾ Laboratories ▾ Mac ▾ Misc ▾ News ▾ Online Accounts ▾ Science ▾ Technology ▾ Travel ▾

NICOS common page · https://atlastagcollector... · E*TRADE FINANCIAL

NICOS version 1_0_0

NICOS GLOBAL LIST OF PROJECTS

Last modified Mon Jan 09 2006 15:30:00

This web page shows the list of ATLAS nightly builds and provides the links to the most recent releases (updated when the copy of the newly built release to AFS is completed). There is also the list of experimental ATLAS nightly project builds available here.

Project Title	NICOS Suffix	Latest Release	Status	Date	Failed Builds	Test Success(%)
10.5.1,OPT	1051	rel_0	completed	11/15 18:49	6	38
11.0.4,DBG	Slc110XDbg	atlrel_1	make completed, copy(if needed) and tests started	01/09 15:22	1	N/A
11.0.4,OPT	Slc110X	atlrel_1	make completed, copy(if needed) and tests started	01/09 15:21	0	N/A
11.1.0_root5,OPT	1110root5	rel_0	completed	01/09 03:50	5	42
11.2.0,DBG	SlcDbg	atlrel_1	completed	01/09 07:45	0	63
11.2.0,OPT	SlcOpt	atlrel_1	completed	01/09 07:22	0	62

- This page shows the list of nightly projects managed by the NICOS system
- Click on the project title to get the project web page with the list of all releases available
- Click on the latest release name to get the summary of build and test results

---

http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/dist/nightlies/nicos_web_are · Google

ATLAS ▾ Experiments ▾ Laboratories ▾ Mac (2) ▾ Misc ▾ News ▾ Online Accounts ▾ Science ▾ Technology (14) ▾ Travel ▾

COS project page · https://atlastagcollector...

NICOS version 1_0_0

NICOS (Nightly COntrol System) project page

Project: 11.2.0,DBG
tags for 11.2.0 ; compiler: gcc 3.2.3 ; dbgion: dbg ; CMT v1r18p20051108
Location: /build/atnight/localbuilds/nightlies/rel
Releases are copied to: /afs/cern.ch/atlas/software/dist/nightlies/rel

Last modified Mon Jan 09 2006 07:45:54

NEWS: Nov 02, 2005. The start times of 11.X.0 DBG and OPT builds are swapped: the DBG build starts first and then the OPT build (which is because optimization takes time) follows. Previously the slower OPT build started first and an interlocking mechanism (the package can not ltaneously compiled in dbg and opt mode) prevented the faster pace of the DBG build.

2005. "Global" NICOS web pages with the lists of nightlies are now available:

tandard ATLAS nightlies
xperimental project nightlies

mportant nightlies:

ptimized 11.X.0 builds
ptimized 11.0.X builds

release name	status	date	copy status	list of tags
atlrel_1	completed	01/09 07:45	completed	list of tags
atlrel_0	completed	01/08 16:00	completed	list of tags
atlrel_6	completed	01/07 07:25	completed	list of tags
atlrel_5	completed	01/06 07:12	completed	list of tags
atlrel_4	unit tests completed	01/05 01:28	completed with errors; additional copy of test results expected	list of tags
atlrel_3	completed	01/04 06:35	completed	list of tags
atlrel_2	completed	01/03 06:31	completed with errors	list of tags

mary of all available projects.

ows the status of nightly builds of 11.2.0,DBG project managed by the NICOS system
ovides a link to the Project Configuration Page

e above

NICOS (NIghtly COntrol System) build results

Project: 11.2.0,DBG
Release: atlrel_1 -- Build incrementally on: lxbuild012.cern.ch
tags for 11.2.0 ; compiler: gcc 3.2.3 ; dbgion: dbg ; CMT v1r18p20051108
Work area for incrementals: /build/atnight/localbuilds/nightlies/rel/atlrel_0
Releases are copied to: /afs/cern.ch/atlas/software/dist/nightlies/rel
Highlighted packages have problems, click on names to see logfiles

Last modified Mon Jan 09 2006 07:44:25

Integration+Unit tests results		Click for details
passed:	33 + 13	failed:	25 + 2

Useful links: UCL Run Time Testing (RTT) results
ATNight (ATLAS Testing Nightly) wiki page

Build results for individual packages. Sorted by:

packages names, failures first   build order   containers names

Package Name	Container	Build	Checkreq Test	Unit Test	Manager(s)
CTB_G4Sim	Simulation/G4Sim	✓	N/A	FAIL	mgallas@mail.cern.ch, andrea.di.simone@cer
InDetRecExample	InnerDetector/InDetExample	✓	N/A	FAIL	markus.elsing@cern.ch, edward.moyse@cern.
AGDD_DTD	DetectorDescription/AGDD	✓	N/A	N/A	arnault@lal.in2p3.fr
AGDD_Model	DetectorDescription/AGDD	✓	N/A	N/A	arnault@lal.in2p3.fr
AGDD_Parser	DetectorDescription/AGDD	✓	N/A	N/A	arnault@lal.in2p3.fr
AGDD_Svc	DetectorDescription/AGDD	✓	N/A	N/A	arnault@lal.in2p3.fr
AGDD_XML	DetectorDescription/AGDD	✓	N/A	N/A	arnault@lal.in2p3.fr
AMI	Database/Bookkeeping	✓	N/A	N/A	solveig.albrand@lpsc.in2p3.fr, jerome.fulachie
ATNight	AtlasTest/NightlyTestTools	✓	N/A	N/A	undrus@bnl.gov
AcerMC_i	Generators	✓	N/A	N/A	borut.kersevan@cern.ch
Age	Tools	✓	N/A	N/A	arnault@lal.in2p3.fr
AgeCommon	Tools	✓	N/A	N/A	arnault@lal.in2p3.fr
AgeStructures	Tools	✓	N/A	N/A	arnault@lal.in2p3.fr
AgeToCxx	Tools	✓	N/A	N/A	arnault@lal.in2p3.fr

David R. Quarrie: ATLAS Offline Release Process

Tag Collector

- Web-based tool to manage
    - Creation of new releases
    - Specification of which packages and versions belong in a release
    - API to extract set of packages and versions for a release
        - Used by NICOS for nightly builds
    - Management tools to control update access
        - Role based
            - Release coordinator, package manager(s)
- Based on Grenoble AMI DB framework

David R. Quarrie: ATLAS Offline Release Process

David R. Quarrie: ATLAS Offline Release Process

Issues with existing system

- Full nightly build takes ~19 hours
  - AFS instabilities caused us to make builds on local disk & copy to AFS
  - Having to rsync CVS repository & copy to AFS contribute to total time
    - But failures dropped from ~20% to <5%
- Use incremental builds to reduce build time
  - Typically ~6-8 hours (but up to 19 hours if core packages changed)
  - Do a full build every Saturday night
- Disk space becoming a problem
  - Present disk layout doesn't scale well with new platforms
  - Reaching 20GB AFS volume size limit
- Problem with instability
  - Changes to core packages immediately visible (and disruptive)
  - Manual integration builds necessary
- Solution: Project-based builds

David R. Quarrie: ATLAS Offline Release Process

Project Dependencies & Sizes

AtlasOffline [5]

AtlasAnalysis [73]

AtlasTrigger [88]

AtlasReconstruction [170]

AtlasSimulation [142]

AtlasEvent [182]

AtlasConditions [150]

AtlasCore [108]

Gaudi [20]

LCGCMT [39]

<project> [<n>]

<n> No.of leaf packages
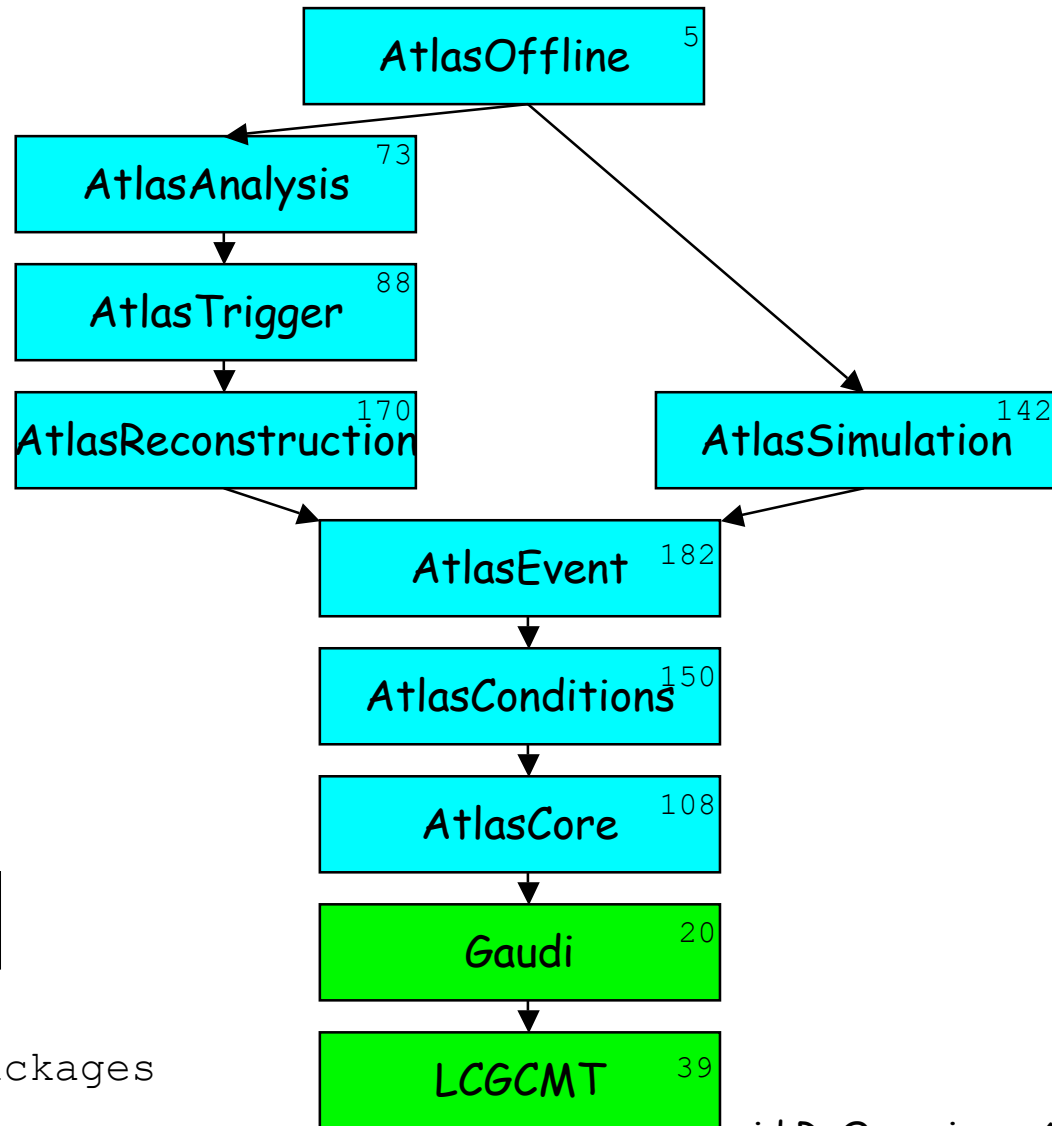
David R. Quarrie: ATLAS Offline Release Process

Projects

- ATLAS Offline split into projects

- Packages associated with a single project

- Each can have it's own independent development timeline

- Higher level projects can develop against stable versions of lower level ones

- Concerns about manual integration cascades - e.g.

    - New version of lower level project to be integrated
    - Initial testing restricted to project itself
    - Discover bugs in lower level project during integration with next project up
    - Requires new version of lower level project etc. etc.

- CMT provides support for projects

    - Project package (same name as project itself)
        - Specifies dependencies against other projects and *project container* package and version
    - Project container package (<project>Release)
        - Specifies top level container & leaf package versions

David R. Quarrie: ATLAS Offline Release Process

Tag Collector Support for Projects

- Multiple top level projects supported
  - Further extensions to specify different CVS repository or CVS offset underway

- Releases associated with each project

- Each leaf package associated with a project
  - Causes packages within e.g. the InnerDetector hierarchy to be associated with different projects (AtlasSimulation, AtlasEvent, AtlasReconstruction, etc.)

- Each container package "belongs" to every project that has one or more of it's children leaf packages associated
  - Where TC generates automatically the CMT requirements file specifying the set of child packages belonging to each project

- Release coordinators per project & release

- Doesn't yet provide support for *project* package
  - Not yet possible to see or specify project version dependencies

- Does provide support for *project container* package
  - Hierarchical autotagging

David R. Quarrie: ATLAS Offline Release Process

TC & Projects

David R. Quarrie: ATLAS Offline Release Process

Project Releases & Types

- Existing release types retained, but now per project
  - *Development*. Snapshot with limited testing every 3-4 weeks
    - Release X.Y.z (where bug-fixes can now be associated with development releases)
  - *Production*. Intended for integration with other projects and possible production
    - Release X.0.0
  - *Bug-fix*. Bugs discovered by the production use or during project integration are fixed in these
    - Release X.y.z (can be associated with both development & production releases)
  - *Nightly*. Performed every night for each project. Releases kept in a cycle for several days but then overwritten
    - Corresponds to open (i.e. not yet existing) releases within the tag collector
- New types of nightly builds
  - *Standalone*. Project specifies fixed versions of the project or projects it depends upon
    - Supports development against a stable base of other projects
  - *Integration*. Each project built against the corresponding nightly of the next project down in the project dependency tree
    - Supports bug-fixing across multiple projects simultaneously
    - Within each such integration nightly, higher level projects don't begin their build until lower level ones have completed
    - The nightly release cycle for all projects must be the same number of days
    - Already supported by NICOS

David R. Quarrie: ATLAS Offline Release Process

Sets of Project Nightlies

- A maximum of 3 per project
  - One integration nightly corresponding to an open bug-fix branch
    - C.f. 11.0.3
  - One integration nightly corresponding to an open development or production branch
    - C.f. 11.1.0 or 12.0.0
  - One standalone nightly corresponding to the project development head
    - New for project builds
    - Supports development against stable versions of other projects
    - Allows a project to embark on a potentially disruptive development cycle without disrupting other projects that depend upon it
      - Once this project developers think it is stable enough for others to try out, a project integration phase can be started, which could be staged to minimize overall disruption
- The first two are essentially identical to monolithic release situation
- The 3rd one adds "stable development"
- Note that this is a worst case, and not necessary at all times
  - But it might be simplest to keep these running even if they're not active

David R. Quarrie: ATLAS Offline Release Process

What does this look like for Developers?

- On Developers Home view for each open release the following options for applying new tags to a package (MyPackage) in the AtlasCore package:

  AtlasCore/3.0.3/MyPackage-01-02-03   [AtlasOffline/12.0.1]

  AtlasCore/3.1.1/MyPackage-01-02-03   [AtlasOffline/12.1.0]

  AtlasCore/3.2.0/MyPackage-01-02-03   [standalone]

  - AtlasCore/3.0.3 is the project version that's being integrated into release 12.0.1

  - AtlasCore/3.1.0 is being integrated into release 12.1.0

  - AtlasCore/3.2.0 is the standalone development version

ATLAS Offline Release Process

Status

- Project builds being built in parallel to monolithic builds
  - Package tags swept every night from open AtlasRelease monolithic releases to corresponding projects
- Plan is to terminate monolithic releases after 11.2.0
  - 19 Jan 2006
- Additional Tag Collector support due by end of March
  - View & set project dependencies

David R. Quarrie: ATLAS Offline Release Process

Summary

- ATLAS relies heavily on tools and automation for release building and testing
    - Tag Collector, NICOS, CMT
- Hierarchy of release builds
    - Nightly, development, production, bug-fix
- Monolithic release has problems with scaling to multiple platforms and instability
- Decomposition into projects designed to address these
- Standalone and integration nightlies to address integration cascade concerns
- Tool project support almost all in place
- Full conversion to projects scheduled for 19 Jan