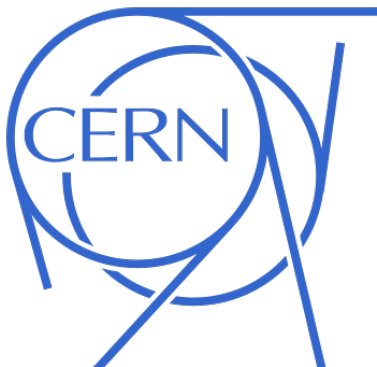


Modernising ATLAS Software Build Infrastructure

ACAT 2017
21-25 August 2017, Seattle

Elmar Ritsch (CERN)
for the ATLAS Collaboration



Overview

- **The Past**
- **Review of the Past Infrastructure**
- **The Solutions**
- **The Migration Process**

The Past: Version Control

- > 2000 packages in **Subversion**
- **Package-level** access control
- **Custom-made tools**
 - *ATLAS Tag Collector* to define list of package tags for each nightly and release
 - *ATLAS SVN Admin* to manage access control lists

Subversion Repository

atlasoff Search

logged in as ritsch | [Logout](#) | [Preferences](#) | [Help/Guide](#) | [About Trac](#)

[Wiki](#) | [Timeline](#) | [Roadmap](#) | **[Browse Source](#)** | [View Tickets](#) | [New Ticket](#) | [Search](#)

[Last Change](#) | [Revision Log](#)

source: View revision: View diff against:

Name ▲	Size	Rev	Age	Author	Last Change
AnalysisBase		805318	5 weeks	atagcol	Tag Collector commit
AnalysisSUSY		731690	15 months	atagcol	Tag Collector commit
AnalysisTop		806778	2 weeks	atagcol	Tag Collector commit
AsgExternal		806792	2 weeks	krumnack	updating the data and mc
AtlasAnalysis		274940	7 years	atagcol	Tag Collector commit
AtlasAnalysisRelease		768813	10 months	atagcol	Tag Collector commit
AtlasAnalysisRunTime		192526	8 years	dquarrie	AtlasAnalysisRunTime?-00

ATLAS Tag Collector

Dependency package version tree of release 20.20.0 for AtlasOffline

(collapse all)

20.20.0 (switch to simple tree)

- AtlasCommonPolicy-00-00-55
- AtlasCxxPolicy-00-00-77
- AtlasFortranPolicy-00-00-81
- AtlasGeometryCommon
 - BeamPipeGeoModel-00-00-19

The Past: Building

- Use **CMT** to
 - Define dependencies between packages
 - Generate Makefiles
- **Custom build system** to schedule, execute and distribute daily nightly builds
 - Use distcc and ccache to speed up builds

Configuration Management Tool (CMT)

```

package AthenaPoolServices

author Marcin Nowak <Marcin.Nowak@cern.ch>

use AtlasPolicy           AtlasPolicy-*           External
use GaudiInterface        GaudiInterface-*           External

private
use AtlasPOOL             AtlasPOOL-*           External
use DataModelRoot        DataModelRoot-*       Control
use AthenaBaseComps      AthenaBaseComps-*     Control
use AtlasROOT             AtlasROOT-*           External

apply_pattern declare_joboptions files="*.py"

apply_pattern named_component_library library=AthenaRootStreamerSvc
    
```

Nightly Control System (NICOS)

UPGRADE_INTEGRATION GROUP:											
20.20.X	rel_4	1/11	20.20.10	🟢	29-JUN 10:35	29-JUN 11:15	🟢	🟢	🟢	0 (205)	70 (67)
20.20.X-VAL	rel_4	1/11	20.20.10	🟢	29-JUN 10:25	29-JUN 10:49	🟢	🟢	🟢	9.0 (211)	56 (55)
21.9.X-VAL	rel_4	1/12	21.9.1	🟢	29-JUN 05:26	29-JUN 05:37	🟢	🟢	🟢	0 (246)	70 (63)
TIERO GROUP:											
20.7.X-VAL	rel_4	1/11	20.7.10	🟢	29-JUN 12:00	29-JUN 12:44	🟢	🟢	🟢	0 (188)	83 (73)
PATCH GROUP:											
19.2.X.Y-VAL-Prod	rel_4	1/1	19.2.5.23	🟢	29-JUN 03:38	29-JUN 03:50	🟢	🟢	🟢	0 (0)	100 (100)
20.20.8.Y-VAL-Prod	rel_4	1/1	20.20.8.9	🟢	29-JUN 01:09	29-JUN 01:17	🟢	🟢	🟢	0 (4.0)	50 (48)
20.20.X.Y-VAL-Prod	rel_4	1/1	20.20.9.2	🟢	29-JUN 00:53	29-JUN 00:58	🟢	🟢	🟢	0 (1.0)	65 (52)
20.7.5.Y-VAL-Prod	rel_4	1/1	20.7.5.15	🟢	29-JUN 08:02	29-JUN 08:38	🟢	🟢	🟢	0 (18)	61 (60)
20.7.X.Y-VAL-AtlasDerivation	rel_4	1/1	20.7.8.20	🟢	29-JUN 02:21	29-JUN 02:45	🟢	🟢	🟢	0 (21)	100 (100)
21.0.X.Y-VAL-AtlasDerivation	rel_4	1/1	21.0.19.6	🟢	29-JUN 01:34	29-JUN 01:41	🟢	🟢	🟢	0 (5.0)	67 (67)

The Review

- **Internal review** of ATLAS offline software and build infrastructure at the end of 2015
 - With external experts
- **Custom tools and workflows**
 - Developers have to learn them specifically for ATLAS
 - Have to be maintained by ATLAS
- **Have to wait 1+ day** to see build and test result
- A broken nightly build
 - Is **annoying** for developers if they wanted to test / build on top of their latest changes in this nightly build
 - Can **delay a release** by at least a day since we'll need to wait for the next successful nightly build
- Rely on developers that changes are sufficiently tested by them to **not break the nightly for everyone**

Example

- Adding a change that involves modifications to multiple packages (e.g. interface change) is tedious
 - Access rights to *all* packages that need modification required
 - Resulting set of Subversion tags needs to be put into the nightly in one go to not break the nightly
 - Difficult to review

The Solutions



Jenkins

- Move build configuration from CMT to **CMake**
- Move version control from Subversion to **Git**
- Establish **ATLAS Flow** as the main development workflow
 - Heavily based on GitLab Flow
- **Code Review** for all changes
 - Using GitLab
- **Continuous Integration Tests** for all changes
 - Using Jenkins
- Create **public Offline Software Documentation webpage**
 - Searchable through Google

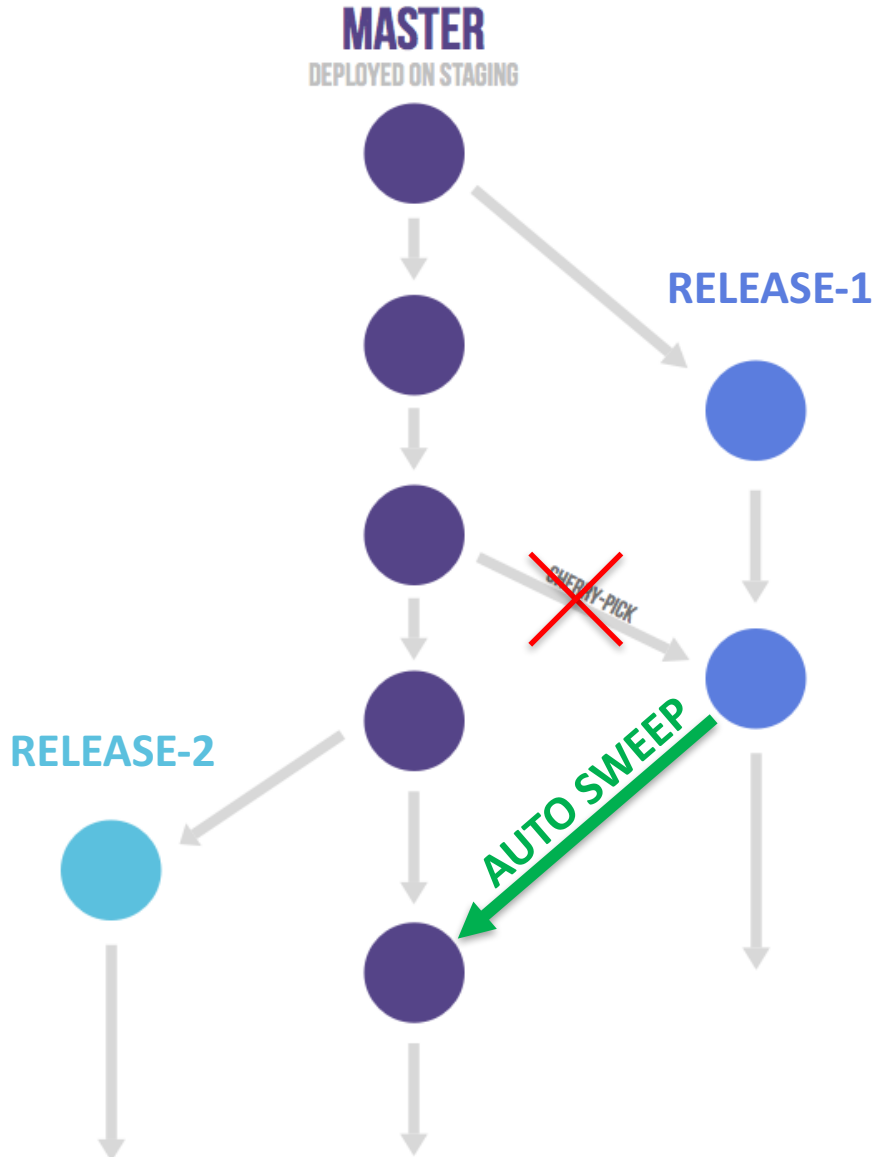
Migration from CMT to CMake

- **Custom migration** script to convert CMT requirements files to CMakeLists.txt for every package
 - Very few exceptions needed to be migrated/adjusted manually
- Features migrated from CMT to CMake:
 - **Build RPMs**
 - Building of **lightweight releases (“flavors”)** containing only a subset of all packages
 - Analysis, Detector simulation

Migration from Subversion to Git

- Available migration scripts **not suited** for structure of ATLAS offline software repository
- **Custom migration scripts** developed to move code to Git
 - Only migrated historical package versions **that were in a release** in the last two years
 - Current development branches migrated
 - **Dropped** no longer used packages
 - **Dropped unused files** >100 KiB
 - Put Copyright statements in source files
- Repository size decreased by more than a **factor of 200x**
 - From **47 GiB** in Subversion to **~220 MiB** in Git
 - Working tree ~640 MiB at the moment
- Created **custom git-atlas command** to allow developers to do sparse checkout of specified packages only

ATLAS Development Workflow



- Almost same as [GitLab Flow](#)
 - Exception is cherry-picking
- **Multi-stage code review** for every merge request
 - Distributes knowledge of codebase within collaboration
- Difficult to apply “upstream first” policy
 - Divergence of code
 - Bug reports often only reproducible in release branches
- Automatically “sweep” (=cherry pick) changes **from release branches to master branch**

Continuous Integration

- **Dedicated build machines**
- **Incremental builds**
 - Reusing build directories to avoid re-building the entire project every time
- Build + test time **typically 3h-5h** for the entire CI pipeline
- Build and test **results posted on GitLab Merge Request page**

[Phase] - preparation

 [CI-stamp-mr](#) [build #4638](#) (0.15 sec) [Console Output](#)

[Phase] - labels

 [CI-assign-labels](#) [build #4379](#) (10 sec) [Console Output](#)

[Phase] - externals

 [CI-build-externals](#) [build #4344](#) (11 min) [Console Output](#)

[Phase] - cmake

 [CI-cmake-configuration](#) [build #4388](#) (24 min) [Console Output](#)

[Phase] - build

 [CI-gcc-build](#) [build #4323](#) (1 hr 30 min) [Console Output](#)

[Phase] - test

 [CI-q431-test](#) [build #1534](#) (22 min) [Console Output](#)

 [CI-q221-test](#) [build #778](#) (34 min) [Console Output](#)

 [CI-unit-tests](#) [build #4291](#) (2 min 34 sec) [Console Output](#)

 [CI-test-driver](#) [build #1687](#) (12 min) [Console Output](#)

[Phase] - logs_handler

 [CI-logs-handler](#) [build #86](#) (3 min 10 sec) [Console Output](#)

Nightly Builds

- **Full build** every night
- All **release branches + master branch** are built
 - Building 7 different branches currently
- Builds for **multiple platforms** and **7 different “flavors”** (Simulation, Analysis, ...)
- Version is **tagged in Git repository**
- Generate **RPM packages**
 - Allows us to deploy every build as an official release
- **Binaries installed** on network filesystem
 - Accessible to all developers

Build and Test Results

Dynamic Display

- ATLAS custom tool
- Performs **error analysis for build and test results** and stores results in Oracle database
- **Web application** dynamically displays information from the database
- Summary displays of **different levels** (system, branch, specific release)
- Used for **continuous integration builds and nightly builds**
- **Improves usability** as developers don't need to search through raw build and test log files (15+ MiB)

RES	Project	Package	Mail	Container
	Athena	DerivationFrameworkSUSY	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkInDet	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkMuons	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkCalo	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkEGamma	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkTools	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkTop	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkHiggs	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkExotics	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkMCTruth	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkTau	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkL1Calo	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkCore	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkInterfaces	nomail@cern.ch	PhysicsAnalysis/DerivationFramework
	Athena	DerivationFrameworkSM	nomail@cern.ch	PhysicsAnalysis/DerivationFramework

CI Pipeline Interacts with GitLab Merge Request

Discussion 3 Commits 2 Pipelines 1 Changes 4



ATLAS Robot @atlasbot commented about 21 hours ago

Developer



This merge request affects 2 packages:

- Simulation/ISF/ISF_Config
- Simulation/ISF/ISF_FastCaloSim/ISF_FastCaloSimServices

Adding @jchapman ,@ritsch as watchers

ATLAS Robot @atlasbot added **review-pending-level-1** **21.0** labels about 21 hours ago



ATLAS Robot @atlasbot commented about 18 hours ago

Developer



✓ CI Result **SUCCESS**

✓ externals

✓ cmake

✓ make

✓ test

Full details available at [NICOS MR-3282-2017-07-11-20-23](#)

For experts only: Jenkins output [[CI-MERGE-REQUEST 5115](#)] (for remote access see instructions in Jenkins section [here](#))

Add
watchers

Add labels

Build and
test result
summary

Code Review & Automated Sweep to master

Discussion 5 Commits 2 Pipelines 1 Changes 4

Review comments and labelling



Carl Suster @csuster commented a day ago

Developer



This looks clear and well-tested, and the build looks ok so I'll approve.



Carl Suster @csuster added **review-approved** and removed **review-pending-level-1** labels a day ago



Hass AbouZeid @oabouzei merged about 21 hours ago



Hass AbouZeid @oabouzei mentioned in commit [99f619e5](#) about 21 hours ago



Atlas Nightlybuild @atnight added **sweep:done** label about 11 hours ago



Atlas Nightlybuild @atnight commented about 11 hours ago

Developer



Sweep summary

success:

- master

Merge accepted

Automated sweep labelling and status message

Tutorial Sessions

- **Prepared the collaboration** for the new workflow and tools before the change to the infrastructure was made
- Organized multiple **Git tutorials** before moving the codebase to Git
- Organized multiple **code reviewer tutorials** before adopting the new workflow

Public Documentation Webpage

The screenshot shows a web browser window with the URL <https://cern.ch/atlassoftwaredocs/gittutorial/>. The page title is "ATLAS Software Git Workflow". The left sidebar contains a navigation menu with the following items: Tutorial Home, Basics (Help With Git), Detailed Tutorial (Set Up, Fork the Repository, Clone Repository Locally), Develop Code (Make a Merge Request, Resolving Conflicts), Code Review (Continuous Integration, Review a request), Reference (Workflow Quick Reference, Git-ATLAS), and Misc (Migration from SVN). The main content area has the heading "ATLAS Software Git Workflow" and "Introduction". The introduction text states: "These web pages describe how to work with ATLAS offline code hosted in the git version control system. If you have no experience with git at all, it's worth checking out our collection of links in the [Help with git](#) section before anything else. There you'll find a getting started guide for git itself and a leg up from SVN. As well as using git as the source control management program, ATLAS uses the [GitLab service at CERN](#) to provide web".

- **Collection of tutorials and guides**
 - E.g. Development workflow, Code review guide, Release coordinator guide
- **Public** and Google-indexed
 - More likely that developers will find right page through search as opposed to internal TWiki
- **Source in Git** repository, changes are reviewed and automatically deployed through continuous integration & deployment pipeline

Summary

- Moved much of our software infrastructure **from custom tools to well established third-party tools**
 - CMake, GitLab, Git, Jenkins
- Established **code reviews** and **continuous integration tests** as part of the standard workflow
- **Public documentation** webpage
- Held **tutorials to prepare the collaboration** for the new workflow and tools

BACKUP

The Human Factor

- Migration mainly driven by **a few very engaged people**
 - High “**bus factor**”
- Somewhat **less resistance** from the rest of the collaboration than initially expected
 - Most developers are eager to learn de-facto standard tools that are new to them (e.g. Git)