Table of Contents

InstallingRelease14050102

Introduction: AtlasProduction as ATLAS offline software patching mechanism

The AtlasProduction patching release mechanism ("cache") can be used to patch (to a certain level) the base ATLAS release deployed for GRID production. Actually we have two independent patching mechanisms (AtlasPoint1 and AtlasProduction) that operate always on top of the AtlasOffline project of the base release.

AtlasProduction cache: release procedures and validation

The AtlasProduction cache is devoted to the GRID production meanwhile the AtlasPoint1 cache is used in Point1/T0 operation. The patching works by incorporating into the cache tags that will "override" the tags in the base release.

The cache releases are named with the four digit numbering schema. The three initial digits correspond to the base release and the last digit corresponds to the cache number. For example 14.1.0.1 is the first cache created on top of the release 14.1.0. Apart from the number you will always find the name of the project "AtlasProduction" or "AtlasPoint1" in order to identify in a better way the scope of the patch. Once a given cache is made available the number and name used can not be re-used.

This Twiki page is intended for the AtlasProduction release coordinators and all the points here described are under the responsibility of the AtlasProduction release coordinator. In brief the AtlasRelease coordinator:

takes care of the release coordination of the ATLAS production patches (cache):

- tag manipulation and follow up in the TagCollector
- the cache creation and documentation
- the software validation and quality control of the cache assuming also the AtlasProduction project software validation coordinator role in all the open production and development releases

AtlasProduction Release Coordinators

Coordinator	dates-releases
Andreu Pacheco	May 2008, 14.1.0.2
Manuel Gallas	Feb. 2007,12.5.3 - May 2008, 14.1.0.1

Activities with the Tag Collector

The patching caches can collect tags from any of the ATLAS offline software projects (from AtlasCore to AtlasOffline). The tags should not contain changes in the header files because this can imply the need of re-compilation of all the clients of a given package.

The tags in a patching cache are managed within the Tag Collector (TC). Once the cache is available in the TC it will be identified as AtlasProduction or AtlasPoint1 and the corresponding number (4 digits).

The previous cache release (14.1.0.1) has to be "terminated" before a new one (14.1.0.2) can be created and "closed". For the ATLAS patching caches we work in the tag-approval mode, this means that the cache in the TC is always closed (red) and only the release coordinator/s of the cache can accept or reject tags.

The flag "terminated" can be added in TC only if the pacman kit is already out and available for production and after the "AutoTag" of the cache (Termination and AutoTag are TC commands available in the TC for the AtlasProduction release coordinator).

The AtlasProduction release coordinator can create a new patch release based on the previous one which acts as father. So, the new one inherits all the patches from the father. In the TC power use mode click in the "AtlasProduction" and do "CreateRelease" as "leaf-base". Mark the patched release box in the TC and fill all the information concerning the compiler and CMT version.

Tags for the cache can be submited by the developers, release coordinators or AtlasProduction release coordinator using the procedure described in the AtlasReleaseValidation Twiki.

The tags submitted should be validated by the developers using the cache specific nightlies (see below) and they will remain in "pending" mode until they will be accepted by the AtlasProduction release coordinator.

Nightlies associated to the AtlasProduction caches

- In order to guarantee the quality of the patching caches the AtlasProduction and AtlasPoint1 projectes have associated nightlies named as the base release and an extra dummy character "Y" to allocate all the possible caches of a given base release (ex: 14.1.0.Y, can be used for the creation of 14.1.0.1, 14.1.0.2, 14.1.0.3, etc). This nightlies can be found in the NICOS page and the detailed description as well as deadline for accepting tags and release coordinators can be found here. The AtlasProduction release coordinatior should keep updated the previous page.

As soon as a new cache is available in TC, it is needed to sent an email to AtlasRelase in order to enable the associated nigthlies and kit creation.

Tests associated to these nightlies (ATN/RTT/FCT) are available to facilitate the validation procedure (see below for more details)

Nightly validation of the production cache and weekly report

The AtlasProduction cache is devoted to the GRID production using the Python tool infrastructure known as "production transforms". Details about this ATLAS GRID production tool can be found in the TWiki-AtlasPythonTrf. Tests within the NICOS-ATN, RTT and FCT have been implemented as it is explained in the previous page.

The AtlasProduction release coordinator is responsible for the quality control of the production cache assuming the role of the AtlasProduction software validation coordinator and reporting in a weekly basis in the Atlas software validation.

As part of the validation of the cache the Kit Validation plays also an important role as it guarantees that the cache can be installed and simple jobs can be run in an external center or laptop without afs access.

Building the cache

Since release 13.0.30 and in a nightly basis, the cache candidate is built after the cvs check out and compilation of the needed tags (referenced in the TC). No manual intervention is needed. The corresponding pacman files (used in GRID production) and rpms files (used at P1) are generated automatically and they can be made available and deployed. This means that over a week there are 7 (rel_0-rel_6) AtlasProduction candidates that could be made public available.

The decision of making a cache public available or not out of one of the nightly cache candidates is based on the performance of the tests (ATN, RTT,FCT) and on the physics coordination requirements in terms of expected functionality and time constrains.

As it was said the cache is created every night and the log information can be found as well in NICOS (see for example Kit_Prod_14.1.0.Y for the AtlasProduction cache 14.1.0.1 or Kit_Pnt_14.0.1.Y for the AtlasPoint1 14.0.1.Y). The above mentioned pages will contain the information concerning the pacman and rpm creation process as well as the directory in afs used to copy the resulting files, as for example:

```
projcache: /afs/cern.ch/atlas/software/builds/kitrel/nightlies/14.1.0.Y/rel_2
rpmdir: /afs/cern.ch/atlas/software/builds/kitrel/nightlies/14.1.0.Y/rel_2/rpms
```

Error filtering

The production transforms perform a check of the log file looking for ERROR lines and reporting them. If an unexpected line is found the task will be marked as a failure.

If it is strictly needed and the impact of the reported ERROR line is understood, it is possible to filter the corresponding ERROR line in the automatic check the production transforms are doing at the very end of the job. With this error filtering the transform will ignore the error line found in the log file and the job will be marked as a success.

The instructions for filtering error lines can be found in the AtlasPythonTrf Wiki page.

Being ready to deploy a new cache

Once a good nightly cache candidate is identified the deploying procedure can start following the next steps in a lxplus CERN machine (access to the directories should be granted by AtlasRelease to the AtlasRelease coordinator):

- Step 0. Check release consistency between Tag Collector and nightly used to make the cache

```
~alibrari/scripts/tags_diffs_tc_afs.sh --tc 15.1.0.8 --afs rel_6 -p AtlasProduction --night 15.3.
```

- Step 1. Copy the pacman files to the unvalidated path:

```
cd /afs/cern.ch/atlas/software/kits/Production/unvalidated/
cp -rv /afs/cern.ch/atlas/software/builds/kitrel/nightlies/15.3.X.Y-VAL-Prod/rel_6/cache/AtlasPro
cp -rv /afs/cern.ch/atlas/software/builds/kitrel/nightlies/15.3.X.Y-VAL-Prod/rel_6/kits/AtlasProd
cd cache
rm -v *.save
```

- Step 2. Install the unvalidated cache in the unvalidated path for the FCT final test

```
cd /afs/cern.ch/atlas/software/unvalidated/caches/
fs listquota
mkdir 15.1.0.8
cd 15.1.0.8
export CMTCONFIG=i686-slc4-gcc34-opt
source /afs/cern.ch/atlas/software/releases/15.1.0/cmtsite/setup.sh -tag=15.1.0,forceConfig
source /afs/cern.ch/atlas/software/pacman/pacman-latest/setup.sh
pacman -allow lock-override -get http://cern.ch/atlas-computing/links/kitsDirectory/Production/un
```

- Step 3. Test the installation in the unvalidated path in a tmp directory:

```
export CMTCONFIG=i686-slc4-gcc34-opt
source /afs/cern.ch/atlas/software/releases/15.1.0/cmtsite/setup.sh -tag=15.1.0,forceConfig
export CMTPATH=/afs/cern.ch/atlas/software/unvalidated/caches/15.1.0.8/AtlasProduction/15.1.0.8
source /afs/cern.ch/atlas/software/unvalidated/caches/15.1.0.8/AtlasProduction/15.1.0.8/AtlasProd
export AtlasVersion=15.1.0.8
export AtlasPatchVersion=15.1.0.8
```

- Step 4. Run at least one of the transforms from the FCT:

```
cd /tmp/$USER
nohup csc_evgen_trf.py --test runNumber=5180 firstEvent=1 maxEvents=100 randomSeed=26740107 jobCo
tail -f nohup.out
```

The transform command line can be found in the FCT pages (run directories, last_* file)

- Step 5. Prepare the base directory for running panda jobs using the unvalidated release for the Prodsys system (http://panda.cern.ch:25880/server/pandamon/query?mode=taskquery)

```
cd /afs/cern.ch/atlas/software/unvalidated/caches
mkdir -p 15.1.0
cd 15.1.0
ln -s /afs/cern.ch/atlas/software/releases/15.1.0/cmtsite
ln -s /afs/cern.ch/atlas/software/releases/15.1.0/AtlasOffline
ln -s /afs/cern.ch/atlas/software/releases/15.1.0/AtlasReconstruction
mkdir -p AtlasProduction
cd AtlasProduction
ln -s ../../15.1.0.8/AtlasProduction/15.1.0.8
ls -l
```

- Step 6. Recheck the installation against the TC release

```
 ~alibrari/scripts/tags_diffs_tc_afs.sh --tc 15.1.0.8 --afs 15.1.0.8 -p AtlasProduction -b /afs/c
```

- Step 7. Copy the nightly to the builds area

```
ssh alibrari@lxplus
cd /afs/cern.ch/atlas/software/builds/logs/AtlasProduction
~alibrari/scripts/copyNightly.py -p AtlasProduction -r 15.1.0.8 -n 15.3.X.Y-VAL/rel_6
```

Now should be ready to be used from /afs/cern.ch/atlas/software/builds/AtlasProduction/.

- Step 8. Send and email to AtlasRelease to activate the final test of the cache with the FCT and KV (include in CC relevant people like Alessandro de Salvo). This process will take normally 12 hours.

Making the cache public

If the previous steps and tests (FCT and KV) are OK the cache is ready to be deployed for GRID production and installed at CERN. The steps are the following and in the order printed below:

- Step 0. Check consistency with the nightly and the tag collector

```
~alibrari/scripts/tags_diffs_tc_afs.sh --tc 15.1.0.8 --afs rel_6 -p AtlasProduction --night 15
```

- Step 1. Installing in afs releases area:

```
cd /afs/cern.ch/atlas/software/releases/15.1.0
```

here we do:

```
export CMTCONFIG=i686-slc4-gcc34-opt
source /afs/cern.ch/atlas/software/releases/15.1.0/cmtsite/setup.sh -tag=15.1.0,forceConfig
fs listquota
source /afs/cern.ch/atlas/software/pacman/pacman-latest/setup.sh
pacman -allow lock-override -get http://cern.ch/atlas-computing/links/kitsDirectory/Production/un
```

If there is no enough quota (>92%) then login with the `alibrari` account and add quota with the command:

```
afs_admin sq . +1500000
```

- Step 2. Installing a new DBRelease if necessary

```
cd /afs/cern.ch/atlas/software/releases/15.1.0
fs listquota
pacman -allow tar-overwrite -get http://atlas.web.cern.ch/Atlas/GROUPS/DATABASE/pacman4/DBRelease
```

check that the current DBRelease is the new one:

```
cd DBRelease
ls -l
```

- Step 3. Check once more the installation:

```
export CMTCONFIG=i686-slc4-gcc34-opt
source /afs/cern.ch/atlas/software/releases/15.1.0/cmtsite/setup.sh -tag=AtlasProduction,15.1.0.8
```

go to a tmp directory and run at least one check from the FCT:

```
cd /tmp/$USER
nohup Reco_trf.py inputRDOFile=/afs/cern.ch/atlas/offline/external/FullChainTest/long/13.0.30.2/r
tail -f nohup.out
```

Finally make the cache public (after the documentation is ready, see next step)

- Go to

```
cd /afs/cern.ch/atlas/software/kits/Production/slc4_0
fs listquota
```

(remember to check the space in afs destination!!)

```
cp -v ../unvalidated/cache/AtlasProduction_15_1_0_8_* cache
cp -v ../unvalidated/kits/AtlasProduction_15_1_0_8_*  kits
```

(then make the simlink to the production area)

```
cd /afs/cern.ch/atlas/software/kits/Production
lndir slc4_0
```

(then copy the rpms ... not the kits)

```
ssh alibrari@lxplus
cd ~alibrari/scripts
copyKitRPM.py -r rel_6 -b 15.3.X.Y-VAL-Prod
```

Don't move the lock from the unvalidated cache directory!

(then create the pacballs)

```
~alibrari/scripts/create_and_copy_pacball.sh -C i686_slc4_gcc34_opt -r 15.1.0.8 -c Production
~alibrari/scripts/create_and_copy_pacball.sh -C i686_slc5_gcc43_opt -r 15.1.0.8 -c Production
~alibrari/scripts/create_and_copy_pacball.sh -C x86_64_slc5_gcc43_opt -r 15.1.0.8 -c Production
```

Cache announcement

Once the cache is available in the http repository it has to be announced by sending an email to the following lists: hn-atlas-releaseKitAnnounce, atlas-sw-prodcaches,hn-atlas-physics-software-validation,hn-atlas-job-transformations. At this time the cache documentation has to be available (see below). Only after this official announcement the cache can be used

for GRID validation or production.

After Dec 15th, 2008 add sacha.vaniachine@cernNOSPAMPLEASE.ch to new kit announcements.

Cache documentation

The cache has to be documented in the release status web page. As in the example of 14.2.10.2, the documentation should include: the list of tags, tag differences, expectations in terms of functionality (according with the FCT tests) and ERROR filtering used.

The AtlasProduction release coordinator has access to the afs directory:

```
/afs/cern.ch/atlas/www/GROUPS/COMPUTING/projects/releases/status
```

granted by the AtlasRelease librarian and which corresponds to the url:

```
http://atlas-computing.web.cern.ch/atlas-computing/projects/releases/status/
```

```
cd /afs/cern.ch/atlas/www/GROUPS/COMPUTING/projects/releases/status
mkdir -p 15.1.0.8/AtlasProduction
cd 15.1.0.8/AtlasProduction
cp ../../15.1.0.7/AtlasProduction/* .
pico -w index.html
cd ../../
pico -w index.html
```

Basically there it is only needed to update the number of the cache and base release. Another important thing is to point to the "Error Filtering" done in the cache. This can be found in the package PyJobTransformsCore and it is needed to select the right tag associated to the cache.

Create next production cache

Autotag 15.1.0.8 AtlasProduction in TC using the `AutoTagRelease` command, but do not select the "Apply recursively to all release dependencies ?" check box. Terminate the release using the `TerminateRelease` command, again not applying this to the release dependencies. Create release 15.1.0.9 AtlasProduction in TC, using 15.0.0.1 AtlasProduction as the *"Release parent name:"*, and ensuring that the *"Is a patch release:"*, *"Import Tag Approvals from parent release:"* and *"Leaf based"* checkboxes are all set., and that the *"Create an empty release"* checkbox is **not** set. Other check boxes and text entry boxes can be left to their default values.

Adding Trigger tests in each new production cache

In order to maintain tests running the following tags have to be present in each **first** cache. The tag versions must be the same as in the base release.

```
Trigger/TrigValidation/TriggerTest
Trigger/TrigValidation/TrigAnalysisTest
Reconstruction/RecJobTransforms
Reconstruction/RecExample/RecJobTransformTests
Reconstruction/RecExample/RecExAnaTest
PhysicsAnalysis/PATJobTransforms
Generators/EvgenJobTransforms
Simulation/SimuJobTransforms
PhysicsAnalysis/PhysicsValidation/PhysValMon
Trigger/TriggerCommon/TriggerMenuXML
```

Update cronjobs.

Nightlies needs to be switched to the new AtlasProduction 15.1.0.8 release. Change the configuration in http://ami.in2p3.fr/opencms/opencms/AMI/www/NICOS-Configuration/NICOS.html

Interesting links

- Atlas Python production transforms
- ATLAS DBRelease, Replica of the DBRelease on the GRID
- Production System Job Submission/Monitoring web page: PANDA @ CERN, PANDA @ BNL

Major updates:
-- ManuelGallas - 06 May 2008
-- AndresPacheco - 03 Jul 2008

%RESPONSIBLE% Main.unknown
%REVIEW% -- AndresPacheco - 03 Jul 2008

This topic: Main > InstallingRelease14050102
Topic revision: r13 - 2009-11-09 - unknown

Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
Ideas, requests, problems regarding TWiki? Send feedback