Table of Contents

Concept of a dev-version

Over the last two years, it's typically been the case dune-artdaq's requirements aren't satisfied by the official, versioned products in depends on (e.g., artdaq, artdaq-core, etc.). As a consequence, patches have been applied to the versioned code by Fermilab developers on special development branches. Generally, these patches are made possible by building the actual git repos of the patched packages alongside dune-artdaq. However, a new dev-version products area has been created on the cluster, /nfs/sw/artdaq/products_dev, where patched versions of packages exist in a manner such that they're clearly and consistently tagged, and so that all dune-artdaq development areas can use them without needing to use an actual repo. This TWiki describes how to create a dev-version of a package.

Performing a test run

Before creating a dev-version of a package, you need to put its code through at least one run. This means you'll want to have a dune-artdaq development area which contains a git repo for the package you'll be dev-versioning, where the code's been at least locally committed. It should go without saying that you're satisfied with the results of the test run before you proceed with dev-versioning the code!

Automatically installing the dev-version

Once you've performed one or more test runs, you can then dev-version the package with a command that looks like the following:

```
cd <base directory of the dune-artdaq development area in np04-srv-023:/scratch you used for the
. ~np04daq/bin/web_proxy.sh  # Script needs to access the outside world
~np04daq/.jcfree/bin/create_and_install_dev_version.sh <name of package> <descriptive token for t
```

Note that the package name should be given with underscores, e.g., "artdaq_core" rather than "artdaq-core". A concrete example of running the script might be:

```
cd /nfs/sw/work_dirs/dune-artdaq_example_workarea14
. ~np04daq/bin/web_proxy.sh  # Script needs to access the outside world
~np04daq/.jcfree/bin/create_and_install_dev_version.sh artdaq proof_of_concept 7900
```

The script will perform a bunch of sanity checks before taking any concrete action, including whether the dune-artdaq development area you're running out of was actually the one used for the test run. It will then modify the product_deps file of the given package's code repo in the development area so that its listed dependencies (such as artdaq-utilities and artdaq-core, in the case of artdaq's product_deps file) use the versions (or dev-versions) which correspond to the code which was used in the run. Note what this implies: if, say, you performed a test run because you want a dev-version of artdaq, but you used a local git repo of artdaq-core which had modifications made, you'll want to dev-version artdaq-core first before dev-versioning artdaq - otherwise the script will fail with a message like the following:

```
Unable to determine the tag of dependent package artdaq-core which repesents the code used in run
```

After modifying the product_deps file, the script will locally commit it, and then locally tag the code with a tag which looks like:

```
<date expressed in version form>_<branch the code was committed on>_<descriptive token for tag>
```

where again, a concrete version for this might look like "v2019_05_20_for_dune-artdaq_proof_of_concept". It will then source the setupDUNEARTDAQ_forBuilding script in the directory, point MRB_INSTALL to /nfs/sw/artdaq/products_dev, and install the code in /nfs/sw/artdaq/products_dev. Please note that because of this you need to run this script under the same user account that has ownership of the dune-artdaq development area.

Once this is done, you can check that you've now got your new dev-versioned package by running the following:

```
. /nfs/sw/artdaq/products/setup
. /nfs/sw/artdaq/products_dev/setup
ups list -aK+ <package you dev-versioned> <tag of the dev-version you just made>
```

Now you can proceed to push the code and its tag to the central repo.

-- JohnChristianFreeman - 2019-11-13

This topic: CENF > InstallingDevVersion
Topic revision: r6 - 2019-11-13 - JohnChristianFreeman