

TestPatchTips

Introduction

In the Hadoop source tree is `dev-support/bin/test-patch`. This is a wrapper to invoke <https://yetus.apache.org>. It is possible and highly recommended to run this script locally prior to uploading a patch to JIRA. Beyond just these tips, be sure to checkout the Yetus documentation.

In order to get the full power of the tool set, you'll want to make sure that both `findbugs` and `shellcheck` are installed.

Using test-patch.sh

Running `test-patch.sh` will show a usage message that describes all of its options. While there are many listed, there are a few key ones:

- `--basedir` = location of the source repo
- `--dirty-workspace` = the repo isn't pristine, but run anyway
- `--reset-repo` = the repo is allowed to be modified NOTE: This will **DESTROY** any changes in the given repo!
- `--run-tests` = run appropriate unit tests
- filename or JIRA # or HTTP URL = the location of the patch that needs to be tested

Apply and run just the basic checks in a checkout that has other stuff in it:

```
$ dev-support/bin/test-patch --dirty-workspace /tmp/patchfile
```

Apply and run the full unit test:

```
$ dev-support/bin/test-patch --dirty-workspace --run-tests /tmp/patchfile
```

Download a patch from a JIRA and run just the basic checks in a checkout that can be destroyed:

```
$ dev-support/bin/test-patch --resetrepo HADOOP-11820
```

Recommended Usage

In general, the easiest way to use `test-patch.sh` is to use two repos. One repo is used to build patches. The other repo is used to test them.

```
$ git diff --no-prefix trunk > /tmp/1.patch  
  
$ dev-support/bin/test-patch --resetrepo --run-tests --basedir=/test/repo /tmp/1.patch
```

This will run the freshly built patch against the tests in a fresh repo.