Jenkins
Continuous Integration
Build Service

FIFE WORKSHOP
MAY 28, 2015

PATRICK GARTUNG
SCD/SSA/SSI

Jenkins

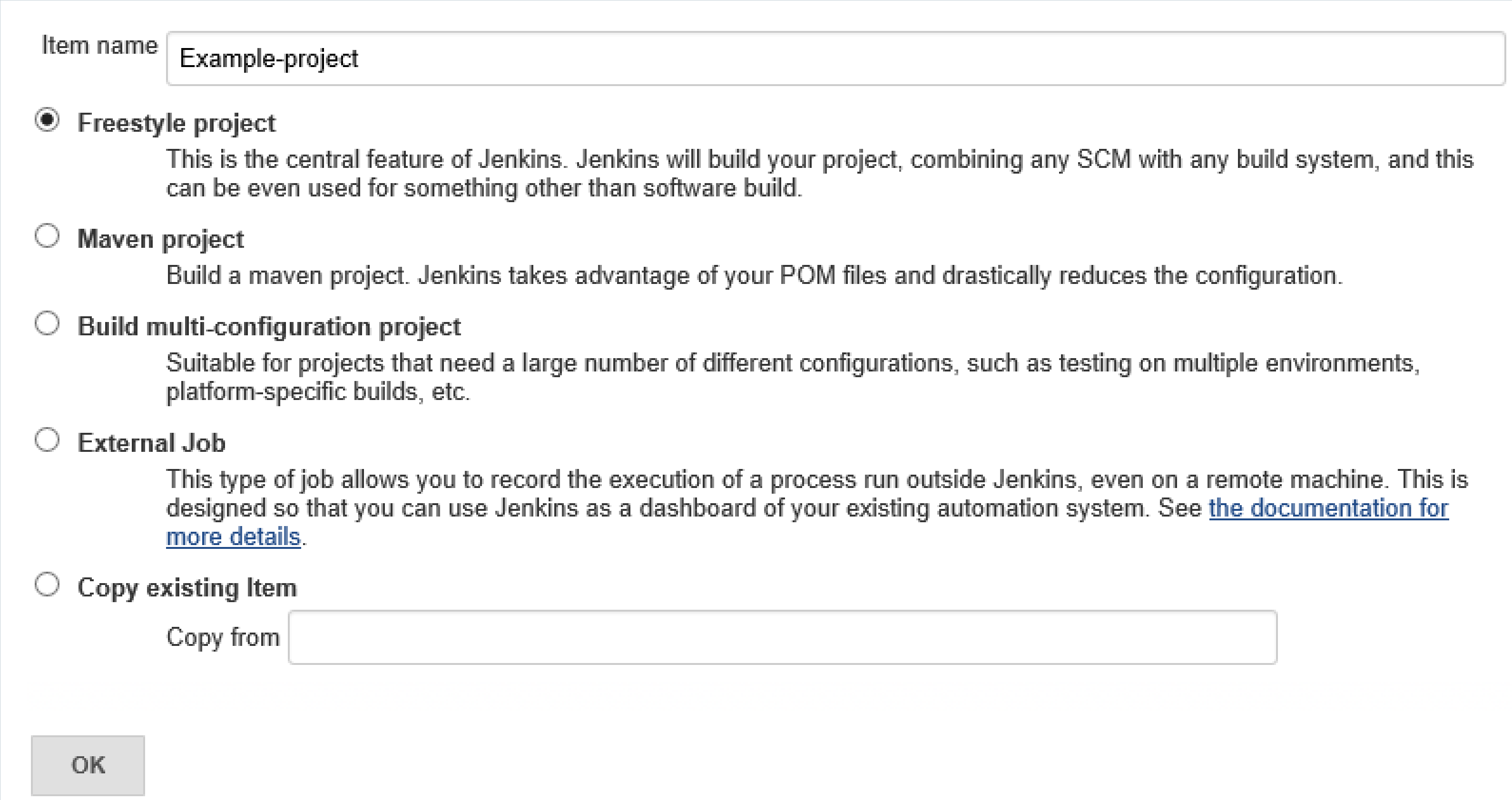
- Continuous Integration:
  - Building and testing your software project
    - After an SCM commit
    - Periodically
    - On demand
- Requirements
  - Web accessible
  - Archiving console log and build results
  - Multiple platforms
  - Dedicated bare-metal servers
- Jenkins meets all of these

Getting started

- Get your Kerberos kca cert and load it in your browser.
- Go to https://buildmaster.fnal.gov/
- You should be prompted to select a certificate.
- You will be recognized as your Kerberos principal if authentication succeeds.
- To begin editing jobs your Kerberos principal needs to be assigned the job-creator role.
- Put in a service desk ticket to request this.

Creating New Project

- Click on New Item; Select Freestyle Project and Enter Item name



Item name: Example-project

● **Freestyle project**

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

○ **Maven project**

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

○ **Build multi-configuration project**

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

○ **External Job**

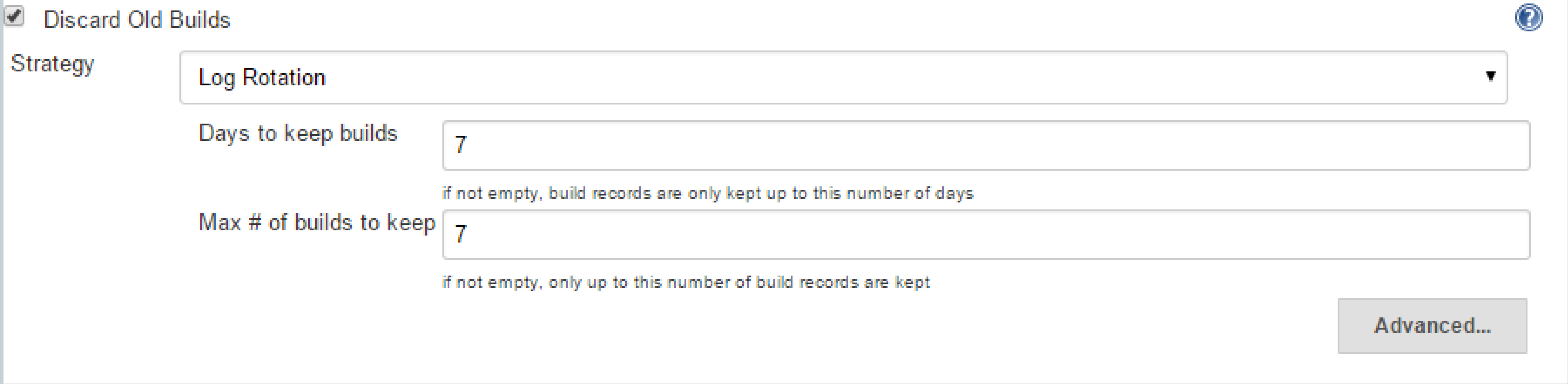
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. See the documentation for more details.

○ **Copy existing Item**

Copy from

Configuring New Project

- Optional: Check Discard Old Builds and enter values



☑ Discard Old Builds

Strategy: Log Rotation

Days to keep builds: 7

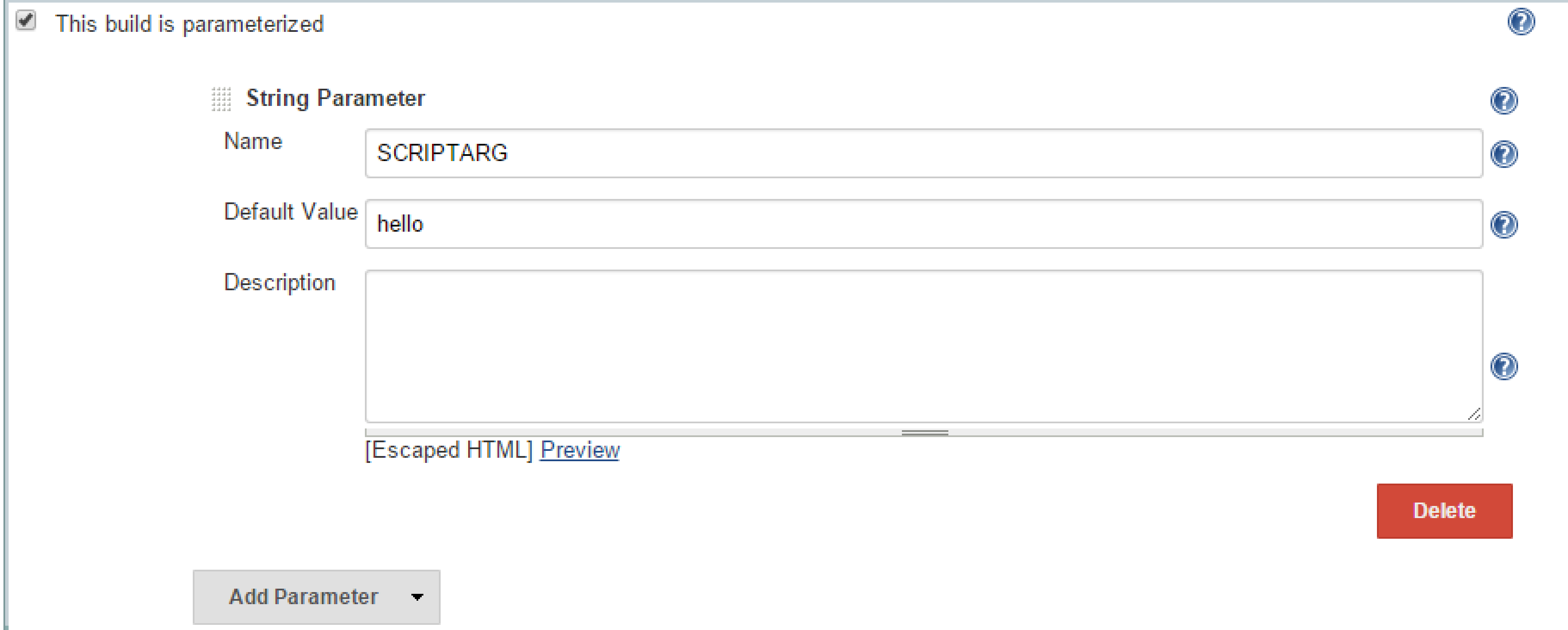
if not empty, build records are only kept up to this number of days

Max # of builds to keep: 7

if not empty, only up to this number of build records are kept

Configuring Project Parameters

- Optional: check "This build is parameterized" then select Add Parameter to add any parameters that your project might need. For example a text parameter could be passed to scripts executed by the project



☑ This build is parameterized

String Parameter

Name: SCRIPTARG

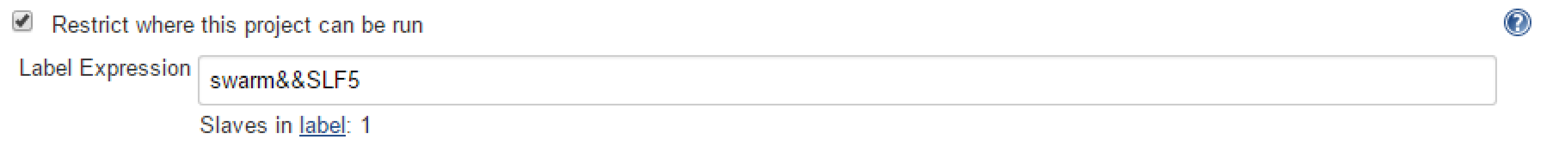
Default Value: hello

Description:

[Escaped HTML] Preview

Selecting where project builds

- Check "Restrict where this project can be run and in Label Expression enter any label to restrict where your job can run
- Current labels
  - swarm
  - SLF5, SLF6,
  - OSX-10.9 ,OSX10.10



☑ Restrict where this project can be run

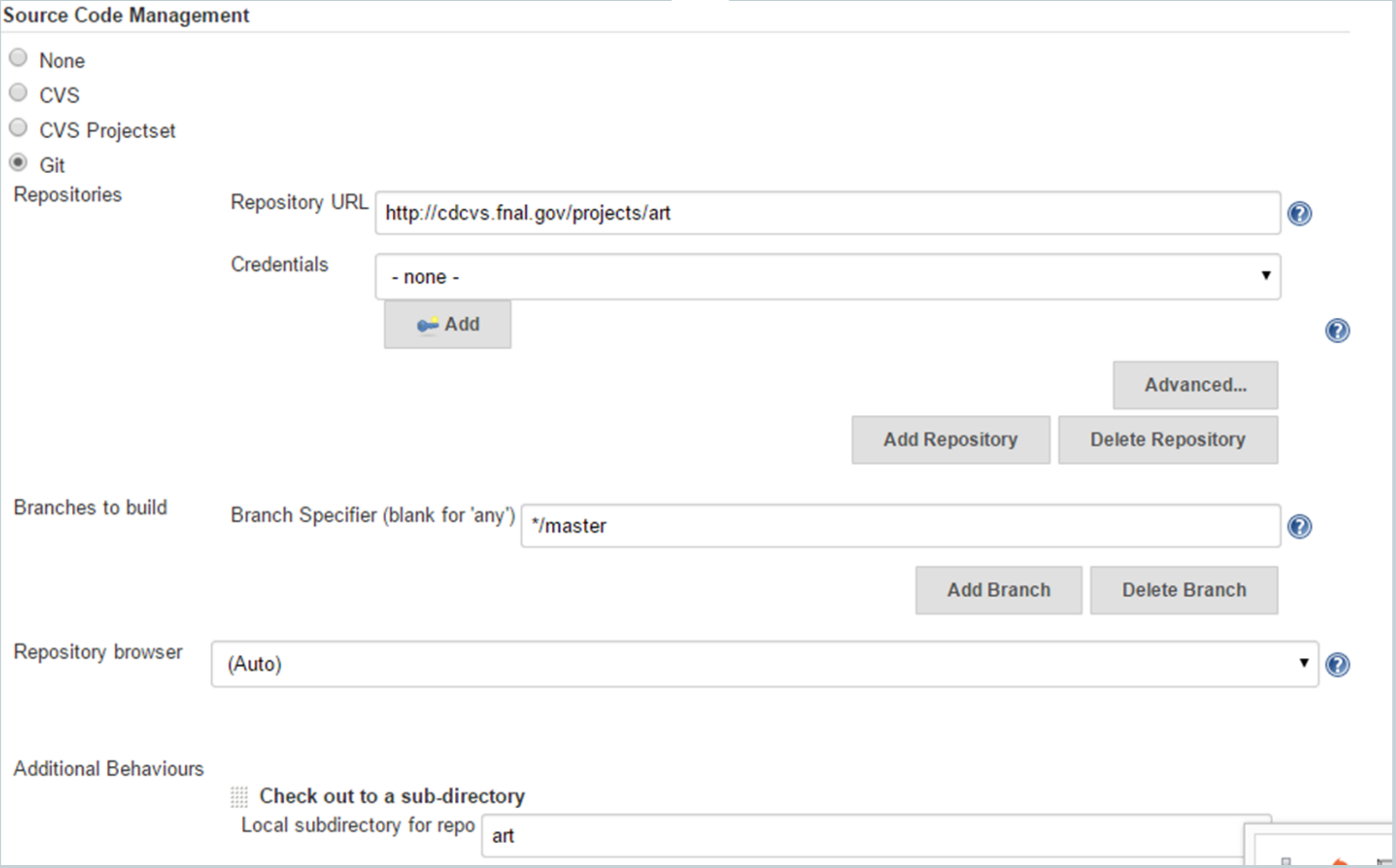
Label Expression: swarm&&SLF5

Slaves in label: 1

Source Code Management

- Optional: allows Jenkins to keep track of changes built
  - Under Source Code Management select Git
  - Enter Repository URL
    - https://cdcvs.fnal.gov/projects/xyz
  - Change Branches to build from */master to */develop to track changes on develop
  - Click Additional Behaviors, select Check out to a sub-directory and enter the subdirectory name

Source Code Management



Source Code Management

○ None
○ CVS
○ CVS Projectset
● Git

Repositories

Repository URL: http://cdcvs.fnal.gov/projects/art

Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

Additional Behaviours

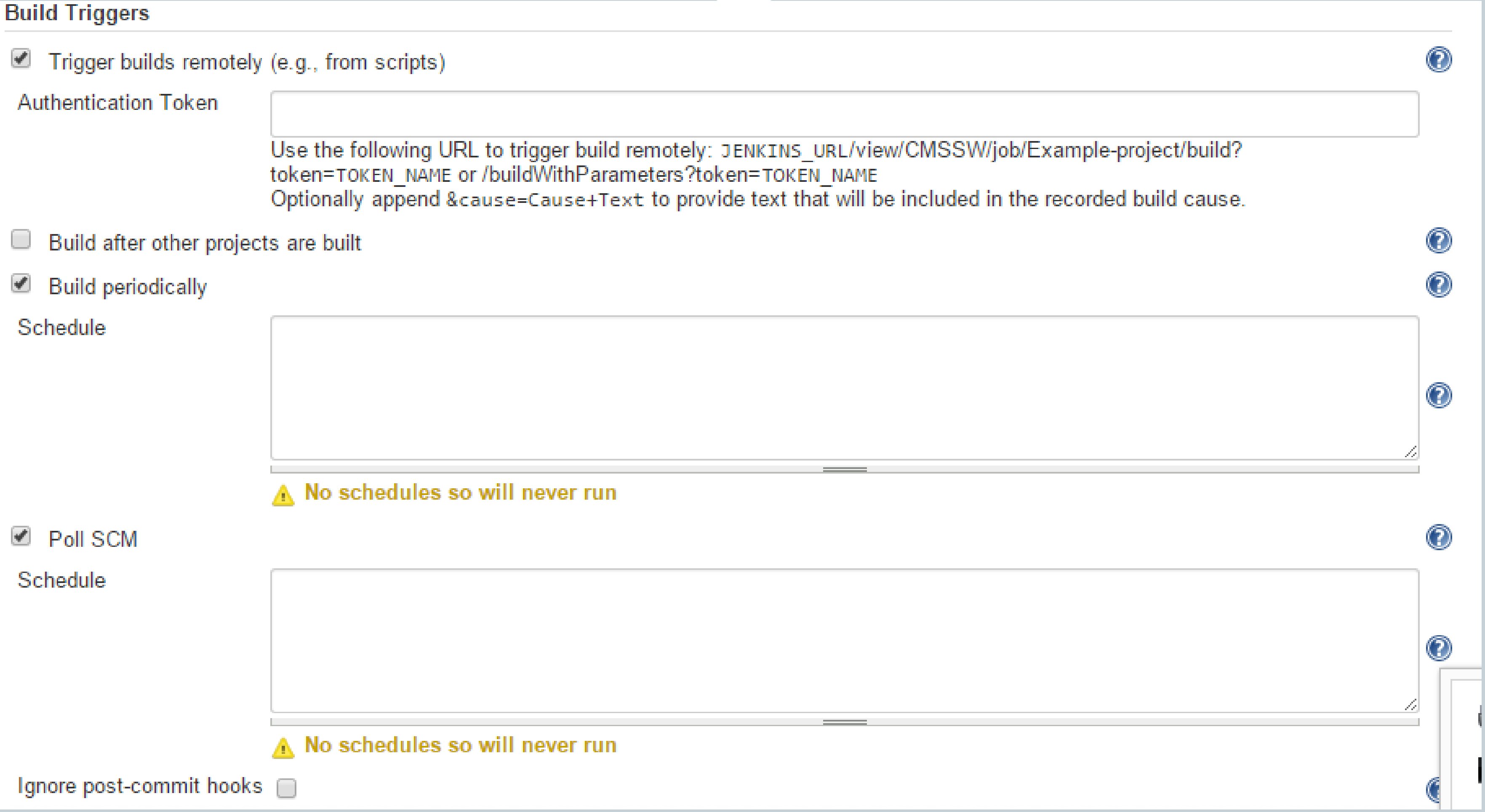
Check out to a sub-directory

Local subdirectory for repo: art

Build Triggers

- Under Build Triggers
  - Select Trigger builds remotely and enter Authentication Token
    - This token should be a string that can be used from the command line and sent in an http request
  - Select Poll SCM and enter a cron-like time pattern
    - This controls how often Jenkins checks the repo for changes
  - Select Build Periodically and enter a cron-like time pattern
    - For example a nightly build

Build Triggers



Build Triggers

☑ Trigger builds remotely (e.g., from scripts)

Authentication Token

Use the following URL to trigger build remotely: `JENKINS_URL`/view/CMSSW/job/Example-project/build?token=`TOKEN_NAME` or /buildWithParameters?token=`TOKEN_NAME`
Optionally append `&cause=Cause+Text` to provide text that will be included in the recorded build cause.

□ Build after other projects are built

☑ Build periodically

Schedule

⚠ No schedules so will never run

☑ Poll SCM

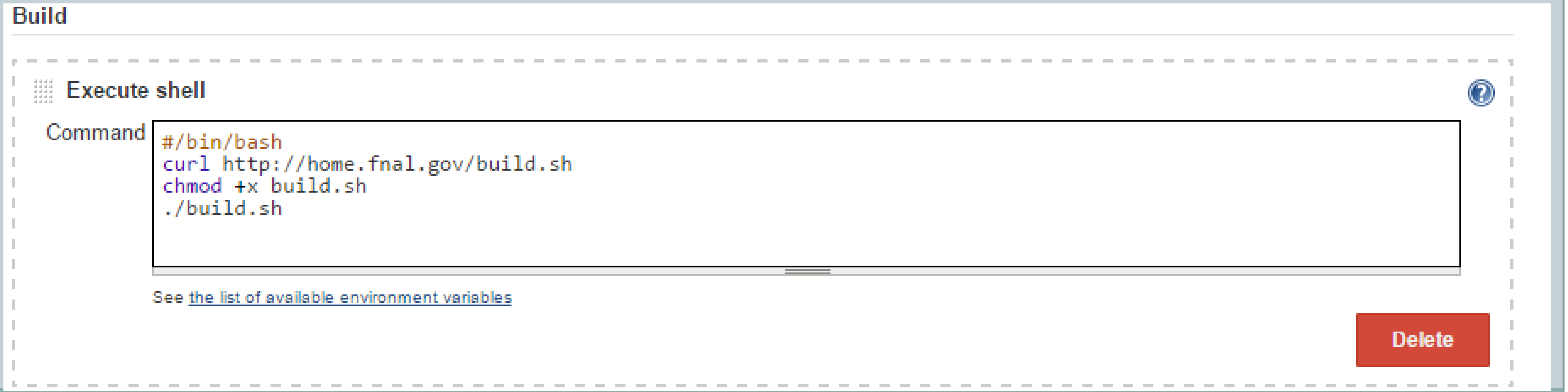
Schedule

⚠ No schedules so will never run

Ignore post-commit hooks □

Build

- Under Build click Add build step and select Execute Shell. This is where the actual build happens.
- Enter a short script
  - You can use #/bin/bash
  - Last exit code is the result of build step.
  - Exit 0 = green anything else red
- You can add multiple build steps



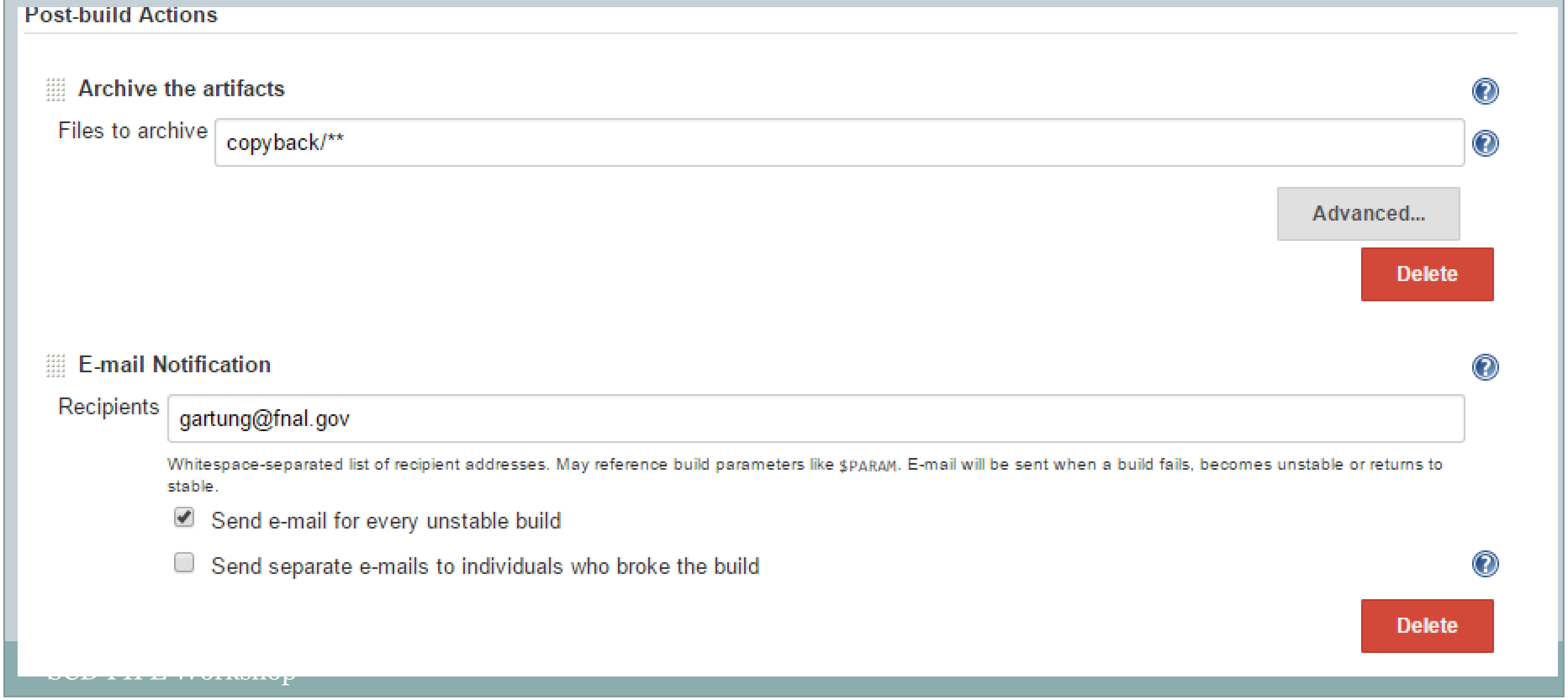
Command

```
#/bin/bash
curl http://home.fnal.gov/build.sh
chmod +x build.sh
./build.sh
```

See the list of available environment variables

Post-build actions

- Under Post-build Actions select Archive the artifacts
- Enter the specific files or wildcards (** = * in java)
- Select E-mail Notification and enter an email address



Post-build Actions

Archive the artifacts

Files to archive: copyback/**

E-mail Notification

Recipients: gartung@fnal.gov

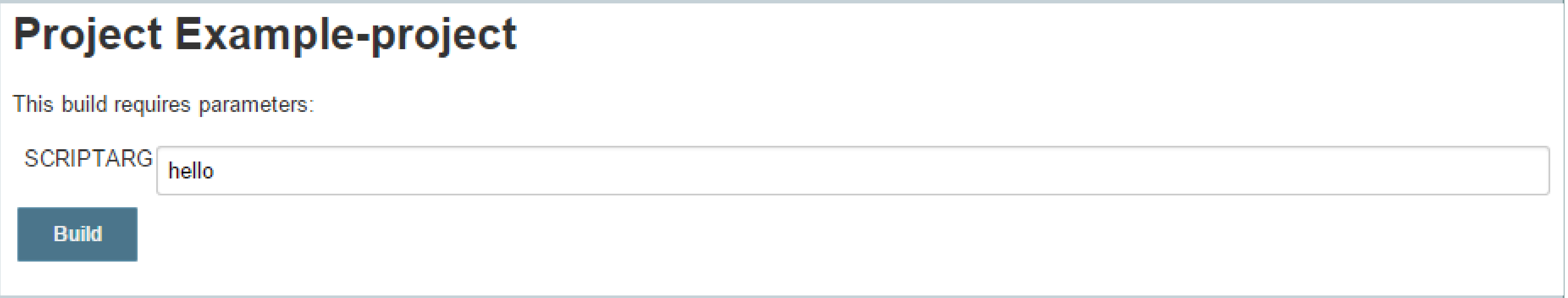
Whitespace-separated list of recipient addresses. May reference build parameters like $PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

☑ Send e-mail for every unstable build

□ Send separate e-mails to individuals who broke the build

Save and Build

- Click Save and the job will now appear in the list.
- Select the job and click Build with Parameters
- Change the parameter if needed and click Build

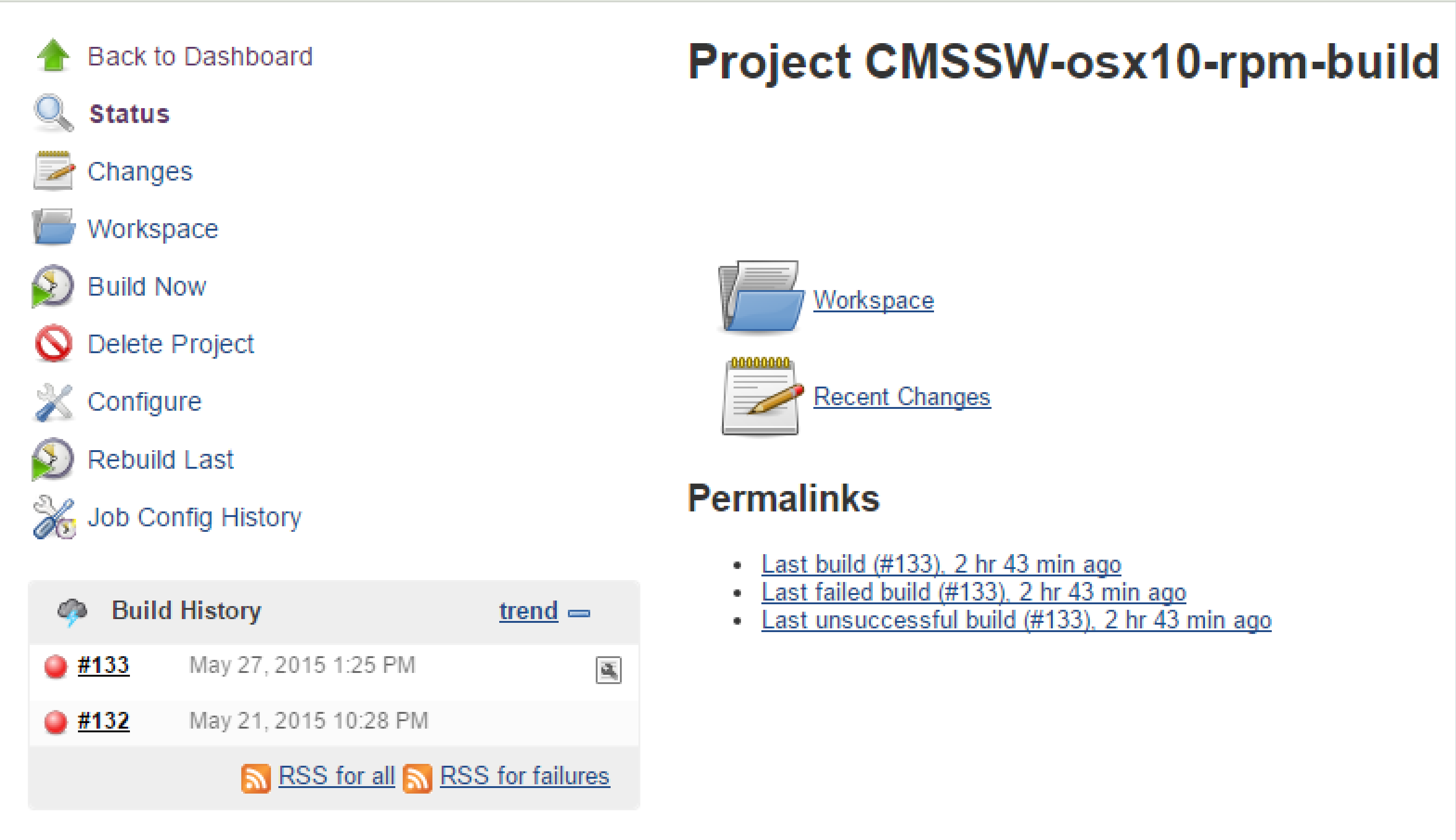


Project Example-project

This build requires parameters:

SCRIPTARG hello

Project Page



Back to Dashboard
Status
Changes
Workspace
Build Now
Delete Project
Configure
Rebuild Last
Job Config History

#133	May 27, 2015 1:25 PM
#132	May 21, 2015 10:28 PM

Project CMSSW-osx10-rpm-build

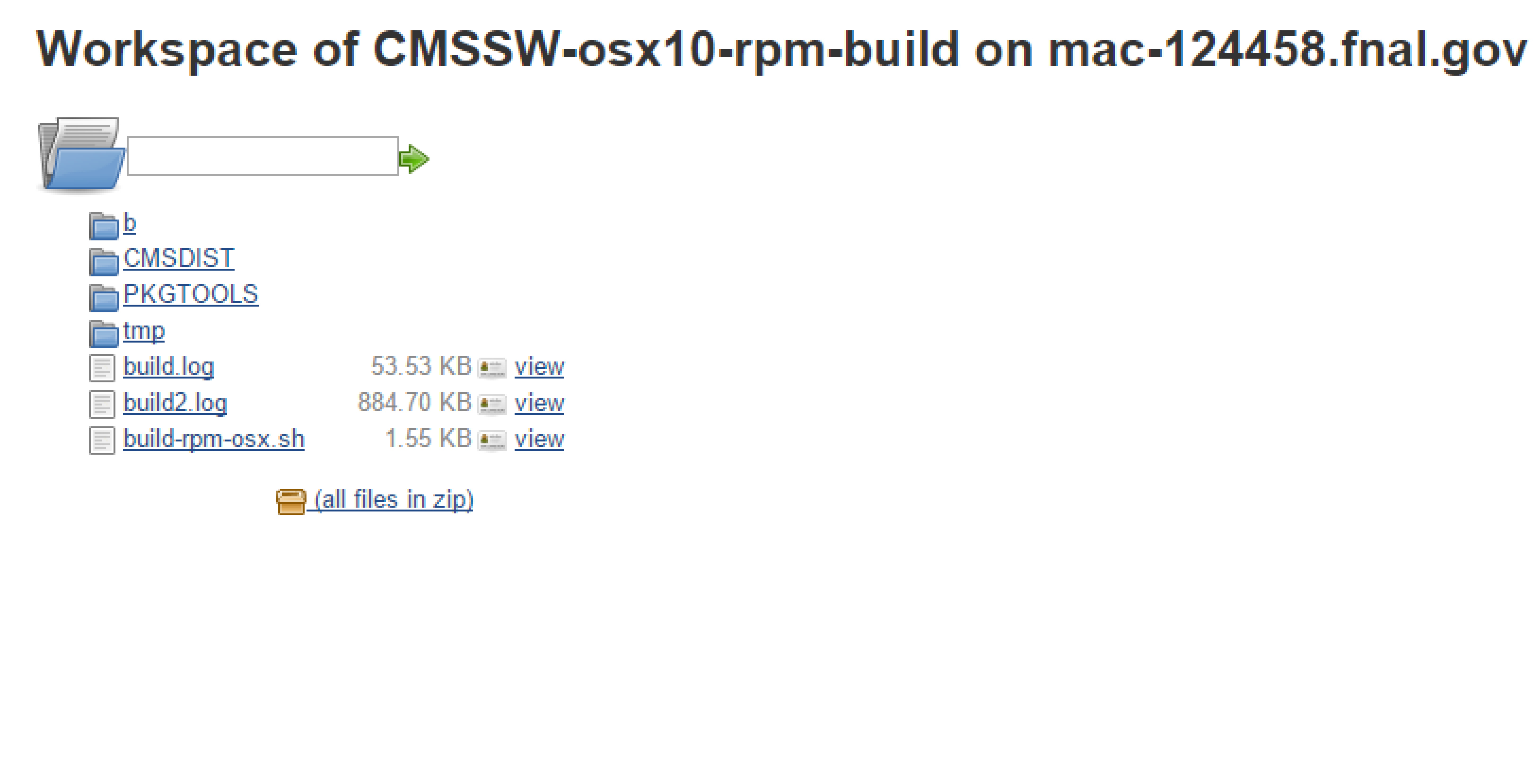
Workspace

Recent Changes

Permalinks

- Last build (#133), 2 hr 43 min ago
- Last failed build (#133), 2 hr 43 min ago
- Last unsuccessful build (#133), 2 hr 43 min ago

Workspace



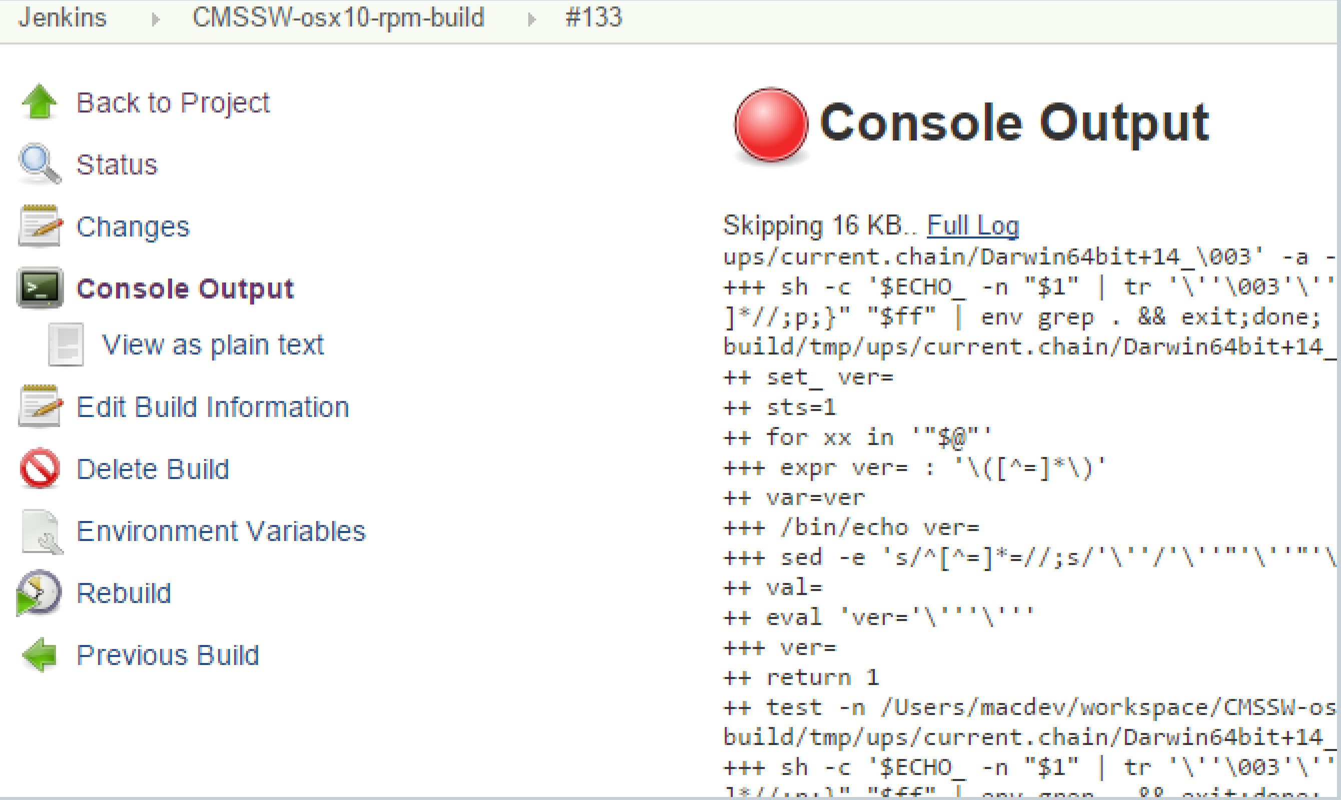
Workspace of CMSSW-osx10-rpm-build on mac-124458.fnal.gov

- b
- CMSDIST
- PKGTOOLS
- tmp

build.log	53.53 KB	view
build2.log	884.70 KB	view
build-rpm-osx.sh	1.55 KB	view

(all files in zip)

Console Output



Console Output

Skipping 16 KB.. Full Log

```
ups/current.chain/Darwin64bit+14_\003' -a -
+++ sh -c '$ECHO_ -n "$1" | tr '\''\003'\''
]*//;p;}" "$ff" | env grep . && exit;done;
build/tmp/ups/current.chain/Darwin64bit+14_
++ set_ ver=
++ sts=1
++ for xx in '"$@"'
+++ expr ver= : '\([^=]*\)'
++ var=ver
+++ /bin/echo ver=
+++ sed -e 's/^[^=]*=//;s/'\''/'\''"'\''"'\
++ val=
++ eval 'ver='\'''\'''
+++ ver=
++ return 1
++ test -n /Users/macdev/workspace/CMSSW-os
build/tmp/ups/current.chain/Darwin64bit+14_
+++ sh -c '$ECHO_ -n "$1" | tr '\''\003'\''
]*//;p;}" "$ff" | env grep . && exit;done;
```

Best Practices

- Clone another project instead of starting from scratch
- Create a script to run the build
- Put that script in SCM or on a web page
- Checkout or fetch that script and run it
- Create a copyback directory and put artifacts there
- Use relative output paths in scripts
- Make use of cvmfs to bootstrap the build
- **Backup job configuration xml**

curl -E /tmp/x509up_u1000 https://buildmaster.fnal.gov/job/{job-name}/config.xml >{job-name}.xml

where /tmp/x509up_u1000 is the KCA cert produced by get-cert.sh

Note: your userid may be something other than 1000.

Jenkins Documentation

- Wiki
  - https://cdcvs.fnal.gov/redmine/projects/build_service/wiki
- Mailing list build-service-users@listserv.fnal.gov
  - Archive http://listserv.fnal.gov/archives/build-service-users.html