

Multi-configuration project name

Description

```
Runs the v8 tests from within the Node tree. Should be run when adding/modifying
```

[Safe HTML] [Preview](#)

Enable project-based security

Compress build logs ?

Discard old builds ?

GitHub project

Project url ?

ADVANCED...

Groovy script to restrict where this project can be run

Notify when Job configuration changes

Permission to Copy Artifact

Preference of Node

Rebuild options: Rebuild Without Asking For Parameters ?

Disable Rebuilding for this job

This project is parameterized ?

String Parameter X ?

Name ?

Default Value ?

Description

```
The user/org of the GitHub repo
```

?

[Safe HTML] [Preview](#)

Trim the string ?

String Parameter X ?

Name ?

Default Value ?

Description

```
The name of the repo
```

?

[Safe HTML] [Preview](#)

Trim the string ?

SAVE

APPLY

X

Default Value ?

Description

```
The remote portion of the Git refs spec to fetch and test
```

 ?
[Safe HTML] [Preview](#)

Trim the string ?

String Parameter X ?

Name ?

Default Value ?

Description

```
Optionally, rebase onto the given ref before testing. Leave blank to not rebase.
```

 ?
[Safe HTML] [Preview](#)

Trim the string ?

String Parameter X ?

Name ?

Default Value ?

Description

```
After rebasing, check that the resulting commit sha1 matches the original commit sha1.
```

 ?
[Safe HTML] [Preview](#)

Trim the string ?

Boolean Parameter X ?

Name ?

Default Value ?

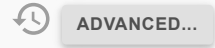
Description

```
Mark the build as unstable instead of failure if only flaky tests failed.
```

 ?
[Safe HTML] [Preview](#)

ADD PARAMETER ▾

- Throttle Concurrent Builds ?
 - Throttle builds ?
 - Use another job as a template ?
 - Allow this job to be used as a template ?
 - Prepare an environment for ?
 - Disable this project ?
-



Source Code Management

- None
- Git

Repositories

Repository URL

Credentials

Name

Refspec

ADD REPOSITORY

Branches to build

Branch Specifier (blank for 'any')

ADD BRANCH

Repository browser

URL

Additional Behaviours

Clean after checkout

Advanced clone behaviours

Fetch tags

Honor refspec on initial clone

Shallow clone

Shallow clone depth

Path of the reference repo to use during clone

Timeout (in minutes) for clone and fetch operations

Calculate changelog against a specific branch

Name of repository

Name of branch

ADD

- Mercurial
- Multiple SCMs
- Subversion

Build Triggers

SAVE

APPLY

Build periodically ?

Schedule ?

Would last have run at Friday, June 15, 2018 5:34:08 AM EDT; would next run at Saturday, June 16, 2018 5:34:08 AM EDT.

GitHub Pull Request Builder ?

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Configuration Matrix

Slaves X ?

Name

Node/Label Labels Individual nodes

Slaves X ?

Name

Node/Label Labels Individual nodes

ADD AXIS ▾

Combination Filter ?

Execution Strategy ▾

Run each configuration sequentially ?

Execute touchstone builds first ?

Build Environment

Use secret text(s) or file(s) ?

Abort the build if it's stuck

Time-out strategy ▾ ?

Timeout seconds ?

Time-out variable

Set a build timeout environment variable

Time-out actions X ? ?

Color ANSI Console Output

ANSI color map

xterm

 Generate Release Notes Inject environment variables to the build process Inject passwords to the build as environment variables SSH Agent Set GitHub commit status with custom context and message (Must configure upstream job using GHPRB trigger) Set Jenkins user build variables With Ant

Build

Execute shell

Command

```
git --version

# Name and email in git config need to be the same across all jobs
# so that git rebase --commit-date-is-author-date will produce
# the same commit SHA1
git config --replace-all user.name Dummy
git config --replace-all user.email dummy@dummy.com
git config user.name
git config user.email
echo $GIT_COMMITTER_NAME
echo $GIT_AUTHOR_NAME

git rebase --abort || true
git checkout -f refs/remotes/origin/_jenkins_local_branch
git config user.name
git config user.email
echo $GIT_COMMITTER_NAME
echo $GIT_AUTHOR_NAME

git status
git rev-parse HEAD
git rev-parse $REBASE_ONTO

if [ -n "${REBASE_ONTO}" ]; then
  git rebase --commit-date-is-author-date $REBASE_ONTO
fi

if [ -n "${POST_REBASE_SHA1_CHECK}" ]; then
  check_sha1=${POST_REBASE_SHA1_CHECK}
  head_sha1=$(git rev-parse HEAD)
  if [ "$head_sha1" != "$check_sha1" ]; then
    exit 1
  fi
fi
```

See [the list of available environment variables](#)

ADVANCED...

Execute shell

Command

```
curl https://raw.githubusercontent.com/nodejs/build/master/jenkins/scripts/node-test-commit-
```

See [the list o](#)

SAVE

APPLY

Execute shell

```
Command # required because otherwise we get clang errors ?
rm -rf deps/v8/tools/clang || true
```

See [the list of available environment variables](#)

ADVANCED...

Execute shell

```
Command # setup depot tools
rm -rf depot_tools || true
git clone https://chromium.googlesource.com/chromium/tools/depot_tools.git
```

See [the list of available environment variables](#)

ADVANCED...

Execute shell

```
Command SKIP_MESSAGE= OK # skipped s390 not supported for versions less than 6, skipping
if [ "$DESTCPU" = "X" ]; then
  cd tools
  echo "from utils import *" >getArch.py
  echo "print GuessArchitecture()" >> getArch.py
  # assume 64 bit unless set specifically
  DESTCPU= python getArch.py | sed 's/ia32/x64/' | sed 's/ppc/ppc64/' | sed 's/arm/arm64/'
  cd ..
fi
echo [DESTCPU=$DESTCPU]

if [ "$DESTCPU" = "XNone" ]; then
  # we only support running v8 tests on recognized platforms. For example
  # if you run this for 4.X on s390 the platform will not be recognized
  # as 4.X does not support s390.
  RUN_TESTS="DONT_RUN"
fi

if [ "$DESTCPU" = "Xppc64" ]; then
  # select the appropriate compiler
  rm -rf build
  git clone https://github.com/nodejs/build.git
  ./build/jenkins/scripts/select-compiler.sh

  # temporarily patch make-v8.sh until we upstream the required
  # changes
  git apply $HOME/build-tools/make-v8.patch

  echo "import platform" > getMachine.py
  echo "print platform.machine()" >> getMachine.py
  ENDIAN= python getMachine.py | sed 's/ppc64/g'
  if [ "$ENDIAN" != "Xle" ]; then

    # ICU tests are disabled for BE platforms because the ICU included by V8 is le only
    # This is not an issue when we build in Node as we include the proper data files for BE
    # it does mean that we need to build/test without ICU when running the v8 tests.
    DISABLE_V8_I18N_OPTION="DISABLE_V8_I18N=1"
  fi
fi
if [ "$DESTCPU" = "Xppc64" ]; then
```

SAVE

APPLY

```
fi

if [ "X$DESTCPU" = "Xs390x" ]; then
# ICU tests are disabled for BE platforms because the ICU included by V8 is le only
# This is not an issue when we build in Node as we include the proper data files for BE bu
# it does mean that we need to build/test without ICU when running the v8 tests.
DISABLE_V8_I18N_OPTION="DISABLE_V8_I18N=1"

# select the appropriate compiler
rm -rf build
git clone https://github.com/nodejs/build.git
. ./build/jenkins/scripts/select-compiler.sh

# temporarily patch make-v8.sh until we upstream the required
# changes
git apply $HOME/build-tools/make-v8.patch

# 390 is only supported on v6 and later for now
MAJOR_VERSION=`cat src/node_version.h |grep "#define NODE_MAJOR_VERSION" | awk '{ print $3 }'`
if [ "$MAJOR_VERSION" -lt "6" ]; then
    RUN_TESTS="DONT_RUN"
fi
fi

if [ "$RUN_TESTS" = "RUN" ]; then
#make -j $(getconf _NPROCESSORS_ONLN) v8 $DISABLE_V8_I18N_OPTION DESTCPU=$DESTCPU ARCH=$ARCH

# when run under jenkins the d8-os.js test hangs because signals don't set to the process
# it works just fine when run from the command line so its not an issue with v8 itself.
# it fails under jenkins but so far have not managed to so remove this test before running
rm deps/v8/test/mjsunit/d8-os.js || true
./configure
make -j $(getconf _NPROCESSORS_ONLN) test-v8 V=1 $DISABLE_V8_I18N_OPTION DESTCPU=$DESTCPU
else
echo $SKIP_MESSAGE
# fake out so we don't get failures
echo '<testsuite tests="1"><testcase name="none/none.none test were skipped" time="0.001"
fi

# clean up so that the v8 directory is always fresh
rm -rf deps/v8 || true
```

See [the list of available environment variables](#)

ADVANCED...

Execute shell

Command `curl https://raw.githubusercontent.com/nodejs/build/master/jenkins/scripts/node-test-commit-d`

See [the list of available environment variables](#)

ADVANCED...

ADD BUILD STEP ▼

Post-build Actions

SAVE

APPLY

'v8-tap.xml' doesn't match anything

Fileset 'includes' setting that specifies the generated raw XML report files, such as 'myproject/target/test-reports/*.xml'. Basedir of the fileset is [the workspace root](#).

Retain long standard output/error

Health report amplification factor

1% failing tests scores as 99% health. 5% failing tests scores as 95% health

Allow empty results Do not fail the build on empty test results

E-mail Notification

Recipients

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

Trigger parameterized build on other projects

Build Triggers

Projects to build

Trigger when build is

Trigger build without parameters

Trigger build from child projects

Current build parameters

ADD PARAMETERS ▾

ADD TRIGGER...

ADD POST-BUILD ACTION ▾

SAVE

APPLY