

Testing a Source Release



Become a contributor

You need to be added as a contributor to edit the wiki. But don't worry! Just email any [Mailing List](#) or grab us on [IRC](#) and let us know your user name.

Any Apache CouchDB user is free to test a source release, and in fact we encourage you to do so!

This might seem like a complex process, but that is only because it has been documented very thoroughly to make sure that we're all doing the same thing. Once you've done this a few times, the whole process becomes very easy to remember.

Setting Up

Remove any existing temporary directory:

```
rm -rf /tmp/couchdb
```

Obviously, don't run this command if you're already using this directory for something else.

The location isn't important, but this is a sensible default.

Create a new temporary directory to work in:

```
mkdir /tmp/couchdb
```

You can safely remove this directory and all of its contents when you're done.

Downloading the Release Candidate

You should have a copy of the vote email with links to the candidate files.

You should now create a another temporary directory:

```
mkdir /tmp/couchdb/dist
```

Change into it:

```
cd /tmp/couchdb/dist
```

And then download these files:

```
curl -O https://dist.apache.org/repos/dist/dev/couchdb/source/VERSION/CANDIDATE/apache-couchdb-VERSION.tar.gz
curl -O https://dist.apache.org/repos/dist/dev/couchdb/source/VERSION/CANDIDATE/apache-couchdb-VERSION.tar.gz.asc
curl -O https://dist.apache.org/repos/dist/dev/couchdb/source/VERSION/CANDIDATE/apache-couchdb-VERSION.tar.gz.sha256
```

The appropriate commands should be in the email for you to copy and paste.

The VERSION and CANDIDATE bits will change between releases.

If you don't have curl installed, you can use wget which should do the same thing.

Verifying the Candidate

All release managers should have their keys in this file:

- <http://apache.org/dist/couchdb/KEYS>

You will need to import the keys into your local keychain before you can continue:

```
curl -L https://apache.org/dist/couchdb/KEYS | gpg --import -
```

GPG

If you don't have [GPG](#) installed, you will get this message:

```
curl: (23) Failed writing body (0 != 16384)
```

Install GPG to continue.

Once that is done, you can verify the release signature:

```
gpg --verify apache-couchdb-*.tar.gz.asc
```

Check for:

- "Good signature from" message.
- The release manager's name, email, and key.

If anything at all about this output look strange to you, please flag it up on the mailing list.

You must then check the SHA256 checksum of the release:

```
sha256sum --check apache-couchdb-*.tar.gz.sha256
```

Check for an "OK" message.

macOS

If you're using macOS and you don't have `sha256sum` installed, you can install it with [Homebrew](#), like so:

```
brew install coreutils
```

Afterwards, you may need to also do this:

```
sudo ln -s /usr/local/bin/gsha256sum /usr/local/bin/sha256sum
```

Checking the Candidate

Skip this section if you don't know anything about the build system.

You should have an `ish` file which contains the Git tree-ish used to build the release.

Create two new temporary directories:

```
mkdir /tmp/couchdb/git
mkdir /tmp/couchdb/tree
```

Make a pristine copy from the tree-ish.

```
git clone https://git-wip-us.apache.org/repos/asf/couchdb.git /tmp/couchdb/git
cd /tmp/couchdb/git
git archive --prefix=/tmp/couchdb/tree/ `cat /tmp/couchdb/dist/apache-couchdb-*.tar.gz.ish` | tar -Pxf -
```

Change into your dist directory:

```
cd /tmp/couchdb/dist
```

Unpack the tarball:

```
tar -xvzf apache-couchdb-*.tar.gz
```

Compare the contents of the tarball to the release tag:

```
diff -r apache-couchdb-VERSION ../tree
```

Replace VERSION with the version you are testing.

Lines starting with "Only in ../tree" should be part of bootstrap mechanism or source control configuration.

Lines starting with "Only in apache-couchdb-VERSION" should be files produced by the bootstrap mechanism.

Nothing else should ever appear in this output.

If you have no idea what any of this means, you probably should have skipped this section!

Sample healthy output:

```
Only in ../tree: .gitignore
Only in apache-couchdb-1.2.0: INSTALL
Only in apache-couchdb-1.2.0: Makefile.in
Only in apache-couchdb-1.2.0: acinclude.m4
Only in ../tree: acinclude.m4.in
Only in apache-couchdb-1.2.0: aclocal.m4
Only in apache-couchdb-1.2.0/bin: Makefile.in
Only in ../tree: bootstrap
Only in apache-couchdb-1.2.0: build-aux
Only in apache-couchdb-1.2.0: config.h.in
Only in apache-couchdb-1.2.0: configure
Only in apache-couchdb-1.2.0/etc: Makefile.in
[...]
Only in apache-couchdb-1.2.0/etc/windows: Makefile.in
Only in apache-couchdb-1.2.0/m4: ac_check_curl.m4
Only in ../tree/m4: ac_check_curl.m4.gz
Only in apache-couchdb-1.2.0/m4: ac_check_icu.m4
Only in ../tree/m4: ac_check_icu.m4.gz
Only in apache-couchdb-1.2.0/m4: libtool.m4
[...]
Only in apache-couchdb-1.2.0/m4: lt~obsolete.m4
Only in apache-couchdb-1.2.0/share: Makefile.in
[...]
Only in apache-couchdb-1.2.0/var: Makefile.in
```

These are the standard checks and are also performed by the release scripts.

However, assume that you cannot trust:

- The source code the archive was built from.
- The host operating system the archive was built on.

An attacker may have compromised either.

Accordingly, you should subject the release candidate to a number of your own tests.

Some ideas:

- Verify the contents of the generated files.
 - Spot check a few of the "Only in apache-couchdb-VERSION" files.
 - Prepare your own reference candidate and compare the files.
- Audit the types of file contained within the archive.
- Run a virus scanner on the archive.

This part of the process is left to your discretion.

Checking the Code

Change into your dist directory:

```
cd /tmp/couchdb/dist
```

Unpack the tarball (if you haven't already done so in the previous steps):

```
tar -xvzf apache-couchdb-*.tar.gz
```

Obviously, skip this step if you've already unpacked the tarball.

Change into the tarball directory:

```
cd apache-couchdb-VERSION
```

Replace VERSION with the version you are testing.

Configure CouchDB to install into your temporary directory:

```
./configure
```

To use a non-default spidermonkey (mozjs) version, for example 60, may have to run:

```
./configure --spidermonkey-version 60
```

Test the whole package:

```
make check
```

If you passed additional arguments to ./configure, you will need to pass them as an environment variable to make.

Do something like this:

```
DISTCHECK_CONFIGURE_FLAGS="--additional-argument" make check
```

This will take a while. You may want to grab a hot beverage.

If this is successful, it should output something like:

```
=====
apache-couchdb-VERSION archives ready for distribution:
apache-couchdb-VERSION.tar.gz
=====
```

If a test fails, you can run it again:

```
$ make eunit apps=chttpd suites=chttpd_view_test
```

Where chttpd_view_test is the name of the test file and chttpd is the Erlang application name, which is usually just the test module prefix.

If this step is failing consistently, you should see the *Troubleshooting* section below.

Install CouchDB into your temporary directory:

```
make release
```

Edit `./rel/couchdb/etc/local.ini` and add an admin user and password

```
[admins]
adm = pass
```

Start CouchDB:

```
/tmp/couchdb/dist/apache-couchdb-VERSION/rel/couchdb/bin/couchdb
```

If you get the following error:

```
Apache CouchDB 1.2.0 (LogLevel=info) is starting.
Failure to start Mochiweb: eaddrinuse
```

You already have a CouchDB server listening on the default port.

Edit your `local.ini` file:

```
$EDITOR /tmp/couchdb/install/etc/couchdb/local.ini
```

Look for this:

```
[httpd]
;port = 5984
```

And change it to this:

```
[httpd]
port = 5985
```

Restart CouchDB.

CouchDB should now be running and listening for connections.

Visit:

- http://127.0.0.1:5984/_utils/

If you had to change the port number in the previous step, you must update it here too.

Locate the "Verify Installation" link in the sidebar.

Verify your installation.

To avoid the `{database_does_not_exist,[{mem3_shards,load_shards_from_db,"_users" error being emitted in the log, create the _users database`

Wrapping it Up

If you manage to get all the way through this without a single error, the release is good. Please reply to the vote email with a summary of the steps you completed. If anything strange happens, or you run into any problems, please reply to the vote email with an outline of the steps you took, and the problems you noticed. In some cases, it might be due to a slight difference in test setup. If that is the case, please update this document to help future testers.

Troubleshooting

The test suite can be a little brittle at times.

This is most noticeable when you are running the distcheck target, as the test suite is run multiple times.

If you want to automatically run the distcheck target multiple times, there is a script for that.

Change into the temporary directory:

```
cd /tmp/couchdb
```

Grab a copy of the admin resources:

```
git clone http://git-wip-us.apache.org/repos/asf/couchdb-admin.git
```

Change into the dist directory:

```
cd /tmp/couchdb/dist
```

Run the script:

```
../couchdb-admin/auto_distcheck.sh VERSION
```

Replace VERSION with the version you are testing.

This script will run the distcheck target in a loop forever, or until the build succeeds — whichever comes first.

Make sure you understand and are comfortable with this before you run it.