

Bitcoin Core integration/staging tree

<https://bitcoincore.org>

For an immediately usable, binary version of the Bitcoin Core software, see <https://bitcoincore.org/en/download/>.

Further information about Bitcoin Core is available in the [doc folder](#).

What is Bitcoin?

Bitcoin is an experimental digital currency that enables instant payments to anyone, anywhere in the world. Bitcoin uses peer-to-peer technology to operate with no central authority: managing transactions and issuing money are carried out collectively by the network. Bitcoin Core is the name of open source software which enables the use of this currency.

For more information read the original Bitcoin whitepaper.

License

Bitcoin Core is released under the terms of the MIT license. See [COPYING](#) for more information or see <https://opensource.org/licenses/MIT>.

Development Process

The `master` branch is regularly built (see `doc/build-*.md` for instructions) and tested, but it is not guaranteed to be completely stable. [Tags](#) are created regularly from release branches to indicate new official, stable release versions of Bitcoin Core.

The <https://github.com/bitcoin-core/gui> repository is used exclusively for the development of the GUI. Its master branch is identical in all monotree repositories. Release branches and tags do not exist, so please do not fork that repository unless it is for development reasons.

The contribution workflow is described in [CONTRIBUTING.md](#) and useful hints for developers can be found in [doc/developer-notes.md](#).

Testing

Testing and code review is the bottleneck for development; we get more pull requests than we can review and test on short notice. Please be patient and help out by testing other people's pull requests, and remember this is a security-critical project where any mistake might cost people lots of money.

Automated Testing

Developers are strongly encouraged to write [unit tests](#) for new code, and to submit new unit tests for old code. Unit tests can be compiled and run (assuming they weren't disabled in configure) with: `make check`. Further details on running and extending unit tests can be found in [/src/test/README.md](#).

There are also [regression and integration tests](#), written in Python. These tests can be run (if the [test dependencies](#) are installed) with: `test/functional/test_runner.py`

The CI (Continuous Integration) systems make sure that every pull request is built for Windows, Linux, and macOS, and that unit/sanity tests are run automatically.

Manual Quality Assurance (QA) Testing

Changes should be tested by somebody other than the developer who wrote the code. This is especially important for large or high-risk changes. It is useful to add a test plan to the pull request description if testing the changes is not straightforward.

Translations

Changes to translations as well as new translations can be submitted to [Bitcoin Core's Transifex page](#).

Translations are periodically pulled from Transifex and merged into the git repository. See the [translation process](#) for details on how this works.

Important: We do not accept translation changes as GitHub pull requests because the next pull from Transifex would automatically overwrite them again.