Dogecoin Core [DOGE, Ð]

==========

What is Dogecoin? – Such coin

Dogecoin is a cryptocurrency like Bitcoin, although it does not use SHA256 as its proof of work (POW). Taking development cues from Tenebrix and Litecoin, Dogecoin currently employs a simplified variant of scrypt.

http://dogecoin.com/

License – Much license

Dogecoin is released under the terms of the MIT license. See COPYING for more information or see http://opensource.org/licenses/MIT.

Development and contributions – omg developers

Development is ongoing, and the development team, as well as other volunteers, can freely work in their own trees and submit pull requests when features or bug fixes are ready.

Version strategy

Version numbers are following `major.minor.patch` semantics.

Branches

There are 3 types of branches in this repository:

- **master:** Stable, contains the latest version of the latest *major.minor* release.
- **maintenance:** Stable, contains the latest version of previous releases, which are still under active maintenance. Format: `<version>-maint`
- **development:** Unstable, contains new code for planned releases. Format: `<version>-dev`

Master and maintenance branches are exclusively mutable by release. Planned releases will always have a development branch and pull requests should be submitted against those. Maintenance branches are there for ***bug fixes only,*** *please submit new features against the development branch with the highest version.*

Very Much Frequently Asked Questions

How much doge can exist? – So many puppies!

Early 2015 (approximately a year and a half after release) there will be approximately 100,000,000,000 coins. Each subsequent block will grant 10,000 coins to encourage miners to continue to secure the network and make up for lost wallets on hard drives/phones/lost encryption passwords/etc.

How to get doge? – To the moon!

Dogecoin uses a simplified variant of the scrypt key derivation function as its proof of work with a target time of one minute per block and difficulty readjustment after every block. The block rewards are fixed and halve every 100,000 blocks. Starting with the 600,000th block, a permanent reward of 10,000 Dogecoin per block will be paid.

Originally, a different payout scheme was envisioned with block rewards being determined by taking the maximum reward as per the block schedule and applying the result of a Mersenne Twister pseudo-random number generator to arrive at a number between 0 and the maximum reward. This was changed, starting with block 145,000, to prevent large pools from gaming the system and mining only high reward blocks. At the same time, the difficulty retargeting was also changed from four hours to once per block (every minute), implementing an algorithm courtesy of

the DigiByte Coin development team, to lessen the impact of sudden increases and decreases of network hashing rate.

The current block reward schedule:

1–99,999: 0–1,000,000 Dogecoin

100,000–144,999: 0–500,000 Dogecoin

145,000–199,999: 250,000 Dogecoin

200,000–299,999: 125,000 Dogecoin

300,000–399,999: 62,500 Dogecoin

400,000–499,999: 31,250 Dogecoin

500,000–599,999: 15,625 Dogecoin

600,000+: 10,000 Dogecoin

The original block reward schedule, with one-minute block targets and four-hour difficulty readjustment:

1–99,999: 0–1,000,000 Dogecoin

100,000–199,999: 0–500,000 Dogecoin

200,000–299,999: 0–250,000 Dogecoin

300,000–399,999: 0–125,000 Dogecoin

400,000–499,999: 0–62,500 Dogecoin

500,000–599,999: 0–31,250 Dogecoin

600,000+: 10,000 Dogecoin

Wow plz make dogecoind/dogecoin-cli/dogecoin-qt

The following are developer notes on how to build Dogecoin on your native platform. They are not complete guides, but include notes on the necessary libraries, compile flags, etc.

- OSX Build Notes
- Unix Build Notes
- Windows Build Notes

Such ports

RPC 22555 P2P 22556

Translations

Changes to translations, as well as new translations, can be submitted to Bitcoin Core's Transifex page.

Periodically the translations are pulled from Transifex and merged into the git repository. See the translation process for details on how this works.

If the changes are Dogecoin specific, they can be submitted as pull requests against this repository. If it is a general translation, consider submitting it through upstream, as we will pull these changes later on.

Development tips and tricks

compiling for debugging

Run configure with the --enable-debug option, then make. Or run configure with CXXFLAGS="-g -ggdb -O0" or whatever debug flags you need.

debug.log

If the code is behaving strangely, take a look in the debug.log file in the data directory; error and debugging messages are written there.

The -debug=... command-line option controls debugging; running with just -debug will turn on all categories (and give you a very large debug.log file).

The Qt code routes qDebug() output to debug.log under category "qt": run with -debug=qt to see it.

testnet and regtest modes

Run with the -testnet option to run with "play dogecoins" on the test network, if you are testing multi-machine code that needs to operate across the internet.

If you are testing something that can run on one machine, run with the -regtest option. In regression test mode, blocks can be created on-demand; see qa/rpc-tests/ for tests that run in -regtest mode.

DEBUG_LOCKORDER

Dogecoin Core is a multithreaded application, and deadlocks or other multithreading bugs can be very difficult to track down. Compiling with -DDEBUG_LOCKORDER (configure CXXFLAGS="-DDEBUG_LOCKORDER -g") inserts run-time checks to keep track of which locks are held, and adds warnings to the debug.log file if inconsistencies are detected.