

Bitcoin Core version *0.15.1* is now available from:

<https://bitcoincore.org/bin/bitcoin-core-0.15.1/>

or

<https://bitcoin.org/bin/bitcoin-core-0.15.1/>

This is a new minor version release, including various bugfixes and performance improvements, as well as updated translations.

Please report bugs using the issue tracker at GitHub:

<https://github.com/bitcoin/bitcoin/issues>

To receive security and update notifications, please subscribe to:

<https://bitcoincore.org/en/list/announcements/join/>

How to Upgrade

If you are running an older version, shut it down. Wait until it has completely shut down (which might take a few minutes for older versions), then run the installer (on Windows) or just copy over `/Applications/Bitcoin-Qt` (on Mac) or `bitcoind/bitcoin-qt` (on Linux).

The first time you run version 0.15.0 or higher, your chainstate database will be converted to a new format, which will take anywhere from a few minutes to half an hour, depending on the speed of your machine.

The file format of `fee_estimates.dat` changed in version 0.15.0. Hence, a downgrade from version 0.15 or upgrade to version 0.15 will cause all fee estimates to be discarded.

Note that the block database format also changed in version 0.8.0 and there is no automatic upgrade code from before version 0.8 to version 0.15.0. Upgrading directly from 0.7.x and earlier without re-downloading the blockchain is not supported. However, as usual, old wallet versions are still supported.

Downgrading warning

The chainstate database for this release is not compatible with previous releases, so if you run 0.15 and then decide to switch back to any older version, you will need to run the old release with the `-reindex-chainstate` option to rebuild the chainstate data structures in the old format.

If your node has pruning enabled, this will entail re-downloading and processing the entire blockchain.

Compatibility

Bitcoin Core is extensively tested on multiple operating systems using the Linux kernel, macOS 10.8+, and Windows Vista and later. Windows XP is not supported.

Bitcoin Core should also work on most other Unix-like systems but is not frequently tested on them.

Notable changes

Network fork safety enhancements

A number of changes to the way Bitcoin Core deals with peer connections and invalid blocks have been made, as a safety precaution against blockchain forks and misbehaving peers.

- Unrequested blocks with less work than the minimum-chain-work are now no longer processed even if they have more work than the tip (a potential issue during IBD where the tip may have low-work). This prevents peers wasting the resources of a node.
- Peers which provide a chain with less work than the minimum-chain-work during IBD will now be disconnected.
- For a given outbound peer, we now check whether their best known block has at least as much work as our tip. If it doesn't, and if we still haven't heard about a block with sufficient work after a 20 minute timeout, then we send a single getheaders message, and wait 2 more minutes. If after two minutes their best known block has insufficient work, we disconnect that peer. We protect 4 of our outbound peers from being disconnected by this logic to prevent excessive network topology changes as a result of this algorithm, while still ensuring that we have a reasonable number of nodes not known to be on bogus chains.
- Outbound (non-manual) peers that serve us block headers that are already known to be invalid (other than compact block announcements, because BIP 152 explicitly permits nodes to relay compact blocks before fully validating them) will now be disconnected.
- If the chain tip has not been advanced for over 30 minutes, we now assume the tip may be stale and will try to connect to an additional outbound peer. A periodic check ensures that if this extra peer connection is in use, we will disconnect the peer that least recently announced a new block.
- The set of all known invalid-themselves blocks (i.e. blocks which we attempted to connect but which were found to be invalid) are now tracked and used to check if new headers build on an invalid chain. This ensures that everything that descends from an invalid block is marked as such.

Miner block size limiting deprecated

Though `blockmaxweight` has been preferred for limiting the size of blocks returned by `getblocktemplate` since 0.13.0, `blockmaxsize` remained as an option for those who wished to limit their block size directly. Using this option resulted in a few UI issues as well as non-optimal fee selection and ever-so-slightly worse performance, and has thus now been deprecated. Further, the `blockmaxsize` option is now used only to calculate an implied `blockmaxweight`, instead of limiting block size directly. Any miners who wish to limit their blocks by size, instead of by weight, will have to do so manually by removing transactions from their block template directly.

GUI settings backed up on reset

The GUI settings will now be written to `guisettings.ini.bak` in the data directory before wiping them when the `-resetguisettings` argument is used. This can be used to retroactively troubleshoot issues due to the GUI settings.

Duplicate wallets disallowed

Previously, it was possible to open the same wallet twice by manually copying the wallet file, causing issues when both were opened simultaneously. It is no longer possible to open copies of the same wallet.

Debug `-minimumchainwork` argument added

A hidden debug argument `-minimumchainwork` has been added to allow a custom minimum work value to be used when validating a chain.

Low-level RPC changes

- The “currentblocksize” value in `getmininginfo` has been removed.
- `dumpwallet` no longer allows overwriting files. This is a security measure as well as prevents dangerous user mistakes.
- `backupwallet` will now fail when attempting to backup to source file, rather than destroying the wallet.
- `listsinceblock` will now throw an error if an unknown `blockhash` argument value is passed, instead of returning a list of all wallet transactions since the genesis block. The behaviour is unchanged when an empty string is provided.

0.15.1 Change log

Mining

- #11100 7871a7d Fix confusing blockmax{size,weight} options, dont default to throwing away money (TheBlueMatt)

RPC and other APIs

- #10859 2a5d099 gettxout: Slightly improve doc and tests (jtimon)
- #11267 b1a6c94 update cli for estimate*fee argument rename (laanwj)
- #11483 20cdc2b Fix importmulti bug when importing an already imported key (pedrobranco)
- #9937 a43be5b Prevent `dumpwallet` from overwriting files (laanwj)
- #11465 405e069 Update named args documentation for importprivkey (dusty-wil)
- #11131 b278a43 Write authcookie atomically (laanwj)
- #11565 7d4546f Make listsinceblock refuse unknown block hash (ryanofsky)
- #11593 8195cb0 Work-around an upstream libevent bug (theuni)

P2P protocol and network code

- #11397 27e861a Improve and document SOCKS code (laanwj)
- #11252 0fe2a9a When clearing addrman clear mapInfo and mapAddr (instagibbs)
- #11527 a2bd86a Remove my testnet DNS seed (schildbach)
- #10756 0a5477c net processing: swap out signals for an interface class (theuni)
- #11531 55b7abf Check that new headers are not a descendant of an invalid block (more effeciently) (TheBlueMatt)
- #11560 49bf090 Connect to a new outbound peer if our tip is stale (sdaftuar)
- #11568 fc966bb Disconnect outbound peers on invalid chains (sdaftuar)
- #11578 ec8dedf Add missing lock in ProcessHeadersMessage(...) (practicalswift)
- #11456 6f27965 Replace relevant services logic with a function suite (TheBlueMatt)
- #11490 bf191a7 Disconnect from outbound peers with bad headers chains (sdaftuar)

Validation

- #10357 da4908c Allow setting nMinimumChainWork on command line (sdaftuar)
- #11458 2df65ee Don't process unrequested, low-work blocks (sdaftuar)

Build system

- #11440 b6c0209 Fix validationinterface build on super old boost/clang (TheBlueMatt)
- #11530 265bb21 Add share/rpcuser to dist. source code archive (MarcoFalke)

GUI

- #11334 19d63e8 Remove custom fee radio group and remove nCustom-FeeRadio setting (achow101)
- #11198 7310f1f Fix display of package name on ‘open config file’ tooltip (esotericnonsense)
- #11015 6642558 Add delay before filtering transactions (lcl)
- #11338 6a62c74 Backup former GUI settings on `-resetguisettings` (laanwj)
- #11335 8d13b42 Replace save|restoreWindowGeometry with Qt functions (MeshCollider)
- #11237 2e31b1d Fixing division by zero in time remaining (MeshCollider)
- #11247 47c02a8 Use IsMine to validate custom change address (MarcoFalke)

Wallet

- #11017 9e8aae3 Close DB on error (kallewoof)
- #11225 6b4d9f2 Update stored witness in AddToWallet (sdaftuar)
- #11126 2cb720a Acquire cs_main lock before cs_wallet during wallet initialization (ryanofsky)
- #11476 9c8006d Avoid opening copied wallet databases simultaneously (ryanofsky)
- #11492 de7053f Fix leak in CDB constructor (promag)
- #11376 fd79ed6 Ensure backupwallet fails when attempting to backup to source file (tomasvdw)
- #11326 d570aa4 Fix crash on shutdown with invalid wallet (MeshCollider)

Tests and QA

- #11399 a825d4a Fix bip68-sequence rpc test (jl2012)
- #11150 847c75e Add getmininginfo test (mess110)
- #11407 806c78f add functional test for mempoolreplacement command line arg (instagibbs)
- #11433 e169349 Restore bitcoin-util-test py2 compatibility (MarcoFalke)
- #11308 2e1ac70 zapwallettxes: Wait up to 3s for mempool reload (MarcoFalke)
- #10798 716066d test bitcoin-cli (jnewbery)
- #11443 019c492 Allow “make cov” out-of-tree; Fix rpc mapping check (MarcoFalke)

- #11445 51bad91 0.15.1 Backports (MarcoFalke)
- #11319 2f0b30a Fix error introduced into p2p-segwit.py, and prevent future similar errors (sdaftuar)
- #10552 e4605d9 Tests for zmqpubrawtx and zmqpubrawblock (achow101)
- #11067 eeb24a3 TestNode: Add wait_until_stopped helper method (MarcoFalke)
- #11068 5398f20 Move wait_until to util (MarcoFalke)
- #11125 812c870 Add bitcoin-cli -stdin and -stdinrpcpass functional tests (promag)
- #11077 1d80d1e fix timeout issues from TestNode (jnewbery)
- #11078 f1ced0d Make p2p-leaktests.py more robust (jnewbery)
- #11210 f3f7891 Stop test_bitcoin-qt touching ~/.bitcoin (MeshCollider)
- #11234 f0b6795 Remove redundant testutil.cpp|h files (MeshCollider)
- #11215 cef0319 fixups from set_test_params() (jnewbery)
- #11345 f9cf7b5 Check connectivity before sending in assumevalid.py (jnewbery)
- #11091 c276c1e Increase initial RPC timeout to 60 seconds (laanwj)
- #10711 fc2aa09 Introduce TestNode (jnewbery)
- #11230 d8dd8e7 Fixup dbcrash interaction with add_nodes() (jnewbery)
- #11241 4424176 Improve signmessages functional test (mess110)
- #11116 2c4ff35 Unit tests for script/standard and IsMine functions (jimpo)
- #11422 a36f332 Verify DBWrapper iterators are taking snapshots (TheBlueMatt)
- #11121 bb5e7cb TestNode tidyups (jnewbery)
- #11521 ca0f3f7 travis: move back to the minimal image (theuni)
- #11538 adbc9d1 Fix race condition failures in replace-by-fee.py, sendheaders.py (sdaftuar)
- #11472 4108879 Make tmpdir option an absolute path, misc cleanup (MarcoFalke)
- #10853 5b728c8 Fix RPC failure testing (again) (jnewbery)
- #11310 b6468d3 Test listwallets RPC (mess110)

Miscellaneous

- #11377 75997c3 Disallow uncompressed pubkeys in bitcoin-tx [multisig] output adds (TheBlueMatt)
- #11437 dea3b87 [Docs] Update Windows build instructions for using WSL and Ubuntu 17.04 (fanquake)
- #11318 8b61aee Put back inadvertently removed copyright notices (gmaxwell)
- #11442 cf18f42 [Docs] Update OpenBSD Build Instructions for OpenBSD 6.2 (fanquake)
- #10957 50bd3f6 Avoid returning a BIP9Stats object with uninitialized values (practicalswift)
- #11539 01223a0 [verify-commits] Allow revoked keys to expire (TheBlue-

Matt)

Credits

Thanks to everyone who directly contributed to this release:

- Andreas Schildbach
- Andrew Chow
- Chris Moore
- Cory Fields
- Cristian Mircea Messel
- Daniel Edgecumbe
- Donal OConnor
- Dusty Williams
- fanquake
- Gregory Sanders
- Jim Posen
- John Newbery
- Johnson Lau
- João Barbosa
- Jorge Timón
- Karl-Johan Alm
- Lucas Betschart
- MarcoFalke
- Matt Corallo
- Paul Berg
- Pedro Branco
- Pieter Wuille
- practicalswift
- Russell Yanofsky
- Samuel Dobson
- Suhas Daftuar
- Tomas van der Wansem
- Wladimir J. van der Laan

As well as everyone that helped translating on Transifex.