

Bitcoin Core version 0.11.0 is now available from:

<https://bitcoin.org/bin/bitcoin-core-0.11.0/>

This is a new major version release, bringing both new features and bug fixes.

Please report bugs using the issue tracker at github:

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

## Upgrading and downgrading

### 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).

### Downgrade warning

Because release 0.10.0 and later makes use of headers-first synchronization and parallel block download (see further), the block files and databases are not backwards-compatible with pre-0.10 versions of Bitcoin Core or other software:

- Blocks will be stored on disk out of order (in the order they are received, really), which makes it incompatible with some tools or other programs. Reindexing using earlier versions will also not work anymore as a result of this.
- The block index database will now hold headers for which no block is stored on disk, which earlier versions won't support.

If you want to be able to downgrade smoothly, make a backup of your entire data directory. Without this your node will need start syncing (or importing from `bootstrap.dat`) anew afterwards. It is possible that the data from a completely synchronised 0.10 node may be usable in older versions as-is, but this is not supported and may break as soon as the older version attempts to reindex.

This does not affect wallet forward or backward compatibility. There are no known problems when downgrading from 0.11.x to 0.10.x.

## Important information

### Transaction flooding

At the time of this release, the P2P network is being flooded with low-fee transactions. This causes a ballooning of the mempool size.

If this growth of the mempool causes problematic memory use on your node, it is possible to change a few configuration options to work around this. The growth of the mempool can be monitored with the RPC command `getmempoolinfo`.

One is to increase the minimum transaction relay fee `minrelaytxfee`, which defaults to 0.00001. This will cause transactions with fewer BTC/kB fee to be rejected, and thus fewer transactions entering the mempool.

The other is to restrict the relaying of free transactions with `limitfreerelay`. This option sets the number of kB/minute at which free transactions (with enough priority) will be accepted. It defaults to 15. Reducing this number reduces the speed at which the mempool can grow due to free transactions.

For example, add the following to `bitcoin.conf`:

```
minrelaytxfee=0.00005
limitfreerelay=5
```

More robust solutions are being worked on for a follow-up release.

## Notable changes

### Block file pruning

This release supports running a fully validating node without maintaining a copy of the raw block and undo data on disk. To recap, there are four types of data related to the blockchain in the bitcoin system: the raw blocks as received over the network (`blk???.dat`), the undo data (`rev???.dat`), the block index and the UTXO set (both LevelDB databases). The databases are built from the raw data.

Block pruning allows Bitcoin Core to delete the raw block and undo data once it's been validated and used to build the databases. At that point, the raw data is used only to relay blocks to other nodes, to handle reorganizations, to look up old transactions (if `-txindex` is enabled or via the RPC/REST interfaces), or for rescanning the wallet. The block index continues to hold the metadata about all blocks in the blockchain.

The user specifies how much space to allot for block & undo files. The minimum allowed is 550MB. Note that this is in addition to whatever is required for the block index and UTXO databases. The minimum was chosen so that Bitcoin Core will be able to maintain at least 288 blocks on disk (two days worth of blocks at 10 minutes per block). In rare instances it is possible that the amount of space used will exceed the pruning target in order to keep the required last 288 blocks on disk.

Block pruning works during initial sync in the same way as during steady state, by deleting block files "as you go" whenever disk space is allocated. Thus, if the user specifies 550MB, once that level is reached the program will begin deleting the oldest block and undo files, while continuing to download the blockchain.

For now, block pruning disables block relay. In the future, nodes with block pruning will at a minimum relay "new" blocks, meaning blocks that extend their active chain.

Block pruning is currently incompatible with running a wallet due to the fact that block data is used for rescanning the wallet and importing keys or addresses (which require a rescan.) However, running the wallet with block pruning will be supported in the near future, subject to those limitations.

Block pruning is also incompatible with `-txindex` and will automatically disable it.

Once you have pruned blocks, going back to unpruned state requires re-downloading the entire blockchain. To do this, re-start the node with `-reindex`. Note also that any problem that would cause a user to reindex (e.g., disk corruption) will cause a pruned node to redownload the entire blockchain. Finally, note that when a pruned node reindexes, it will delete any `blk???.dat` and `rev???.dat` files in the data directory prior to restarting the download.

To enable block pruning on the command line:

- `-prune=N` : where N is the number of MB to allot for raw block & undo data.

Modified RPC calls:

- `getblockchaininfo` now includes whether we are in pruned mode or not.
- `getblock` will check if the block's data has been pruned and if so, return an error.
- `getrawtransaction` will no longer be able to locate a transaction that has a UTXO but whose block file has been pruned.

Pruning is disabled by default.

## Big endian support

Experimental support for big-endian CPU architectures was added in this release. All little-endian specific code was replaced with endian-neutral constructs. This has been tested on at least MIPS and PPC hosts. The build system will automatically detect the endianness of the target.

## Memory usage optimization

There have been many changes in this release to reduce the default memory usage of a node, among which:

- Accurate UTXO cache size accounting (#6102); this makes the option `-dbcache` precise where this grossly underestimated memory usage before
- Reduce size of per-peer data structure (#6064 and others); this increases the number of connections that can be supported with the same amount of memory
- Reduce the number of threads (#5964, #5679); lowers the amount of (esp. virtual) memory needed

## Fee estimation changes

This release improves the algorithm used for fee estimation. Previously, -1 was returned when there was insufficient data to give an estimate. Now, -1 will also be returned when there is no fee or priority high enough for the desired confirmation target. In those cases, it can help to ask for an estimate for a higher target number of blocks. It is not uncommon for there to be no fee or priority high enough to be reliably (85%) included in the next block and for this reason, the default for `-txconfirmtarget=n` has changed from 1 to 2.

## Privacy: Disable wallet transaction broadcast

This release adds an option `-walletbroadcast=0` to prevent automatic transaction broadcast and rebroadcast (#5951). This option allows separating transaction submission from the node functionality.

Making use of this, third-party scripts can be written to take care of transaction (re)broadcast:

- Send the transaction as normal, either through RPC or the GUI
- Retrieve the transaction data through RPC using `gettransaction` (NOT `getrawtransaction`). The `hex` field of the result will contain the raw hexadecimal representation of the transaction
- The transaction can then be broadcasted through arbitrary mechanisms supported by the script

One such application is selective Tor usage, where the node runs on the normal internet but transactions are broadcasted over Tor.

For an example script see [bitcoin-submittx](#).

## Privacy: Stream isolation for Tor

This release adds functionality to create a new circuit for every peer connection, when the software is used with Tor. The new option, `-proxyrandomize`, is on by default.

When enabled, every outgoing connection will (potentially) go through a different exit node. That significantly reduces the chance to get unlucky and pick a single exit node that is either malicious, or widely banned from the P2P network. This improves connection reliability as well as privacy, especially for the initial connections.

**Important note:** If a non-Tor SOCKS5 proxy is configured that supports authentication, but doesn't require it, this change may cause that proxy to reject connections. A user and password is sent where they weren't before. This setup is exceedingly rare, but in this case `-proxyrandomize=0` can be passed to disable the behavior.

## 0.11.0 Change log

Detailed release notes follow. This overview includes changes that affect behavior, not code moves, refactors and string updates. For convenience in locating the code changes and accompanying discussion, both the pull request and git merge commit are mentioned.

### RPC and REST

- #5461 5f7279a signrawtransaction: validate private key
- #5444 103f66b Add /rest/headers/.
- #4964 95ecc0a Add scriptPubKey field to validateaddress RPC call
- #5476 c986972 Add time offset into getpeerinfo output
- #5540 84eba47 Add unconfirmed and immature balances to getwalletinfo
- #5599 40e96a3 Get rid of the internal miner's hashmeter
- #5711 87ecfb0 Push down RPC locks
- #5754 1c4e3f9 fix getblocktemplate lock issue
- #5756 5d901d8 Fix getblocktemplate\_proposals test by mining one block
- #5548 d48ce48 Add /rest/chaininfos
- #5992 4c4f1b4 Push down RPC reqWallet flag
- #6036 585b5db Show zero value txouts in listunspent
- #5199 6364408 Add RPC call `gettxoutproof` to generate and verify merkle blocks
- #5418 16341cc Report missing inputs in sendrawtransaction
- #5937 40f5e8d show script verification errors in signrawtransaction result
- #5420 1fd2d39 getutxos REST command (based on Bip64)
- #6193 42746b0 [REST] remove json input for getutxos, limit to query max. 15 outpoints
- #6226 5901596 json: fail read\_string if string contains trailing garbage

### Configuration and command-line options

- #5636 a353ad4 Add option `-allowselfsignedrootcertificate` to allow self signed root certs (for testing payment requests)
- #5900 3e8a1f2 Add a consistency check `-checkblockindex` for the block chain data structures
- #5951 7efc9cf Make it possible to disable wallet transaction broadcast (using `-walletbroadcast=0` )
- #5911 b6ea3bc privacy: Stream isolation for Tor (on by default, use `-proxyrandomize=0` to disable)
- #5863 c271304 Add autopruner functionality ( `-prune=<size>` )
- #6153 0bcf04f Parameter interaction: disable upnp if -proxy set
- #6274 4d9c7fe Add option `-alerts` to opt out of alert system

### Block and transaction handling

- #5367 dcc1304 Do all block index writes in a batch

- #5253 203632d Check against MANDATORY flags prior to accepting to mempool
- #5459 4406c3e Reject headers that build on an invalid parent
- #5481 055f3ae Apply AreSane() checks to the fees from the network
- #5580 40d65eb Preemptively catch a few potential bugs
- #5349 f55c5e9 Implement test for merkle tree malleability in CPartialMerkleTree
- #5564 a89b837 clarify obscure uses of EvalScript()
- #5521 8e4578a Reject non-final txs even in testnet/regtest
- #5707 6af674e Change hardcoded character constants to descriptive named constants for db keys
- #5286 fcfc646c Change the default maximum OP\_RETURN size to 80 bytes
- #5710 175d86e Add more information to errors in ReadBlockFromDisk
- #5948 b36f1ce Use GetAncestor to compute new target
- #5959 a0bfc69 Add additional block index consistency checks
- #6058 7e0e7f8 autopruning minor post-merge improvements
- #5159 2cc1372 New fee estimation code
- #6102 6fb90d8 Implement accurate UTXO cache size accounting
- #6129 2a82298 Bug fix for clearing fCheckForPruning
- #5947 e9af4e6 Alert if it is very likely we are getting a bad chain
- #6203 c00ae64 Remove P2SH coinbase flag, no longer interesting
- #5985 37b4e42 Fix removing of orphan transactions
- #6221 6cb70ca Prune: Support noncontiguous block files
- #6256 fce474c Use best header chain timestamps to detect partitioning
- #6233 a587606 Advance pindexLastCommonBlock for blocks in chainActive

## P2P protocol and network code

- #5507 844ace9 Prevent DOS attacks on in-flight data structures
- #5770 32a8b6a Sanitize command strings before logging them
- #5859 dd4ffce Add correct bool combiner for net signals
- #5876 8e4fd0c Add a NODE\_GETUTXO service bit and document NODE\_NETWORK
- #6028 b9311fb Move nLastTry from CAddress to CAddrInfo
- #5662 5048465 Change download logic to allow calling getdata on inbound peers
- #5971 18d2832 replace absolute sleep with conditional wait
- #5918 7bf5d5e Use equivalent PoW for non-main-chain requests
- #6059 f026ab6 chainparams: use SeedSpec6's rather than CAddress's for fixed seeds
- #6080 31c0bfl Add jonasschnellis dns seeder
- #5976 9f7809f Reduce download timeouts as blocks arrive
- #6172 b4bbad1 Ignore getheaders requests when not synced
- #5875 304892f Be stricter in processing unrequested blocks
- #6333 41bbc85 Hardcoded seeds update June 2015

## Validation

- #5143 48e1765 Implement BIP62 rule 6
- #5713 41e6e4c Implement BIP66

## Build system

- #5501 c76c9d2 Add mips, mipsel and aarch64 to depends platforms
- #5334 cf87536 libbitcoinconsensus: Add pkg-config support
- #5514 ed11d53 Fix 'make distcheck'

- #5505 a99ef7d Build winshutdownmonitor.cpp on Windows only
- #5582 e8a6639 Osx toolchain update
- #5684 ab64022 osx: bump build sdk to 10.9
- #5695 23ef5b7 depends: latest config.guess and config.sub
- #5509 31dedb4 Fixes when compiling in c++11 mode
- #5819 f8e68f7 release: use static libstdc++ and disable reduced exports by default
- #5510 7c3fbc3 Big endian support
- #5149 c7abfa5 Add script to verify all merge commits are signed
- #6082 7abbb7e qt: disable qt tests when one of the checks for the gui fails
- #6244 0401aa2 configure: Detect (and reject) LibreSSL
- #6269 95aca44 gitian: Use the new bitcoin-detached-sigs git repo for OSX signatures
- #6285 ef1d506 Fix scheduler build with some boost versions.
- #6280 25c2216 depends: fix Boost 1.55 build on GCC 5
- #6303 b711599 gitian: add a gitian-win-signer descriptor
- #6246 8ea6d37 Fix build on FreeBSD
- #6282 daf956b fix crash on shutdown when e.g. changing -txindex and abort action
- #6354 bdf0d94 Gitian windows signing normalization

## Wallet

- #2340 811c71d Discourage fee sniping with nLockTime
- #5485 d01bcc4 Enforce minRelayTxFee on wallet created tx and add a maxtxfee option
- #5508 9a5cabf Add RandAddSeedPerfmon to MakeNewKey
- #4805 8204e19 Do not flush the wallet in AddToWalletIfInvolvingMe(..)
- #5319 93b7544 Clean up wallet encryption code
- #5831 df5c246 Subtract fee from amount
- #6076 6c97fd1 wallet: fix boost::get usage with boost 1.58
- #5511 23c998d Sort pending wallet transactions before reaccepting
- #6126 26e08a1 Change default nTxConfirmTarget to 2
- #6183 75a4d51 Fix off-by-one error w/ nLockTime in the wallet
- #6276 c9fd907 Fix getbalance \* 0

## GUI

- #5219 f3af0c8 New icons
- #5228 bb3c75b HiDPI (retina) support for splash screen
- #5258 73cbf0a The RPC Console should be a QWidget to make window more independent
- #5488 851dfc7 Light blue icon color for regtest
- #5547 a39aa74 New icon for the debug window
- #5493 e515309 Adopt style colour for button icons
- #5557 70477a0 On close of splashscreen interrupt verifyDB
- #5559 83be8fd Make the command-line-args dialog better
- #5144 c5380a9 Elaborate on signverify message dialog warning
- #5489 d1aa3c6 Optimize PNG files
- #5649 e0cd2f5 Use text-color icons for system tray Send/Receive menu entries
- #5651 848f55d Coin Control: Use U+2248 "ALMOST EQUAL TO" rather than a simple tilde
- #5626 ab0d798 Fix icon sizes and column width
- #5683 c7b22aa add new osx dmg background picture
- #5620 7823598 Payment request expiration bug fix

- #5729 9c4a5a5 Allow unit changes for read-only BitcoinAmountField
- #5753 0f44672 Add bitcoin logo to about screen
- #5629 a956586 Prevent amount overflow problem with payment requests
- #5830 215475a Don't save geometry for options and about/help window
- #5793 d26f0b2 Honor current network when creating autostart link
- #5847 f238add Startup script for centos, with documentation
- #5915 5bd3a92 Fix a static qt5 crash when using certain versions of libxcb
- #5898 bb56781 Fix rpc console font size to flexible metrics
- #5467 bc8535b Payment request / server work - part 2
- #6161 180c164 Remove movable option for toolbar
- #6160 0d862c2 Overviewpage: make sure warning icons gets colored

## Tests

- #5453 2f2d337 Add ability to run single test manually to RPC tests
- #5421 886eb57 Test unexecuted OP\_CODESEPARATOR
- #5530 565b300 Additional rpc tests
- #5611 37b185c Fix spurious windows test failures after 012598880c
- #5613 2eda47b Fix smartfees test for change to relay policy
- #5612 e3f5727 Fix zapwallettxes test
- #5642 30a5b5f Prepare paymentservertests for new unit tests
- #5784 e3a3cd7 Fix usage of NegateSignatureS in script\_tests
- #5813 ee9f2bf Add unit tests for next difficulty calculations
- #5855 d7989c0 Travis: run unit tests in different orders
- #5852 cdae53e Reinitialize state in between individual unit tests.
- #5883 164d7b6 tests: add a BasicTestingSetup and apply to all tests
- #5940 446bb70 Regression test for ResendWalletTransactions
- #6052 cf7adad fix and enable bip32 unit test
- #6039 734f80a tests: Error when setgenerate is used on regtest
- #6074 948beaf Correct the PUSHDATA4 minimal encoding test in script\_invalid.json
- #6032 e08886d Stop nodes after RPC tests, even with --nocleanup
- #6075 df1609f Add additional script edge condition tests
- #5981 da38dc6 Python P2P testing
- #5958 9ef00c3 Add multisig rpc tests
- #6112 fec5c0e Add more script edge condition tests

## Miscellaneous

- #5457, #5506, #5952, #6047 Update libsecp256k1
- #5437 84857e8 Add missing CAutoFile::IsNull() check in main
- #5490 ec20fd7 Replace uint256/uint160 with opaque blobs where possible
- #5654, #5764 Adding jonasschnelli's GPG key
- #5477 5f04d1d OS X 10.10: LSSharedFileListItemResolve() is deprecated
- #5679 beff11a Get rid of DetectShutdownThread
- #5787 9bd8c9b Add fanquake PGP key
- #5366 47a79bb No longer check osx compatibility in RenameThread
- #5689 07f4386 openssl: abstract out OPENSSL\_cleanse
- #5708 8b298ca Add list of implemented BIPs
- #5809 46bfbef Add bitcoin-cli man page

- #5839 86eb461 keys: remove libsecp256k1 verification until it's actually supported
- #5749 d734d87 Help messages correctly formatted (79 chars)
- #5884 7077fe6 BUGFIX: Stack around the variable 'rv' was corrupted
- #5849 41259ca contrib/init/bitcoind.openrc: Compatibility with previous OpenRC init script variables
- #5950 41113e3 Fix locale fallback and guard tests against invalid locale settings
- #5965 7c6bfb1 Add git-subtree-check.sh script
- #6033 1623f6e FreeBSD, OpenBSD thread renaming
- #6064 b46e7c2 Several changes to mruset
- #6104 3e2559c Show an init message while activating best chain
- #6125 351f73e Clean up parsing of bool command line args
- #5964 b4c219b Lightweight task scheduler
- #6116 30dc3c1 [OSX] rename Bitcoin-Qt.app to Bitcoin-Core.app
- #6168 b3024f0 contrib/linearize: Support linearization of testnet blocks
- #6098 7708fcd Update Windows resource files (and add one for bitcoin-tx)
- #6159 e1412d3 Catch errors on datadir lock and pidfile delete
- #6186 182686c Fix two problems in CSubnet parsing
- #6174 df992b9 doc: add translation strings policy
- #6210 dfdb6dd build: disable optional use of gmp in internal secp256k1 build
- #6264 94cd705 Remove translation for -help-debug options
- #6286 3902c15 Remove berkeley-db4 workaround in MacOSX build docs
- #6319 3f8fcc9 doc: update mailing list address

## Credits

Thanks to everyone who directly contributed to this release:

- 21E14
- Adam Weiss
- Alex Morcos
- ayeowch
- azeteki
- Ben Holden-Crowther
- bikinibabe
- BitcoinPRReadingGroup
- Blake Jakopovic
- BtcDrak
- charlescharles
- Chris Arnesen
- Ciemon
- CohibAA
- Corinne Dashjr
- Cory Fields
- Cozz Lovan
- Daira Hopwood
- Daniel Kraft
- Dave Collins
- David A. Harding
- dexX7
- Earlz



- Eric Lombrozo
- Eric R. Schulz
- Everett Forth
- Flavien Charlon
- fsb4000
- Gavin Andresen
- Gregory Maxwell
- Heath
- Ivan Pustogarov
- Jacob Welsh
- Jameson Lopp
- Jason Lewicki
- Jeff Garzik
- Jonas Schnelli
- Jonathan Brown
- Jorge Timón
- joshhr
- jtimon
- Julian Yap
- Luca Venturini
- Luke Dashjr
- Manuel Araoz
- MarcoFalke
- Matt Bogosian
- Matt Corallo
- Micha
- Michael Ford
- Mike Hearn
- mrbandrews
- Nicolas Benoit
- paveljanik
- Pavel Janík
- Pavel Vasin
- Peter Todd
- Philip Kaufmann
- Pieter Wuille
- pstratem
- randy-waterhouse
- rion
- Rob Van Mieghem
- Ross Nicoll
- Ruben de Vries
- sandakersmann
- Shaul Kfir
- Shawn Wilkinson
- sinetek
- Suhas Daftuar
- svost
- Thomas Zander
- Tom Harding

- UdjinM6
- Vitalii Demianets
- Wladimir J. van der Laan

And all those who contributed additional code review and/or security research:

- Sergio Demian Lerner

As well as everyone that helped translating on [Transifex](#).