

Bitcoin Core version *0.15.0* is now available from:

<https://bitcoin.org/bin/bitcoin-core-0.15.0/>

This is a new major version release, including new features, 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, 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 0.15.0 or upgrade to version 0.15.0 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 redownloading 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.

## Notes for 0.15.0

### Current SegWit support

Version 0.15.0 supports adding a segregated witness address via the `addwitnessaddress` RPC, but please note that this is a testing/expert RPC, which does not guarantee recovery from backup. Only use this RPC if you know

what you are doing. More complete wallet support for segregated witness is coming in a next version.

## Rescanning with encrypted wallets

As in previous versions, when using an encrypted HD wallet, the keypool cannot be topped up without unlocking the wallet. This means that currently, in order to recover from a backup of an encrypted HD wallet, the user must unlock the wallet with a really long timeout and manually trigger a rescan, otherwise they risk missing some keys when auto-topup cannot run. Unfortunately there is no `rescan` RPC in this version, that will be included in a future version, so for now a rescan can be triggered using one of the `import*` commands, using a dummy address generated by another (trusted) wallet.

## Notable changes

### Performance Improvements

Version 0.15 contains a number of significant performance improvements, which make Initial Block Download, startup, transaction and block validation much faster:

- The chainstate database (which is used for tracking UTXOs) has been changed from a per-transaction model to a per-output model (See [PR 10195](#)). Advantages of this model are that it:
  - avoids the CPU overhead of deserializing and serializing the unused outputs;
  - has more predictable memory usage;
  - uses simpler code;
  - is adaptable to various future cache flushing strategies.

As a result, validating the blockchain during Initial Block Download (IBD) and reindex is ~30-40% faster, uses 10-20% less memory, and flushes to disk far less frequently. The only downside is that the on-disk database is 15% larger. During the conversion from the previous format a few extra gigabytes may be used.

- Earlier versions experienced a spike in memory usage while flushing UTXO updates to disk. As a result, only half of the available memory was actually used as cache, and the other half was reserved to accommodate flushing. This is no longer the case (See [PR 10148](#)), and the entirety of the available cache (see `-dbcache`) is now actually used as cache. This reduces the flushing frequency by a factor 2 or more.
- In previous versions, signature validation for transactions has been cached when the transaction is accepted to the mempool. Version 0.15 extends this to cache the entire script validity (See [PR 10192](#)). This means that if a transaction in a block has already been accepted to the mempool, the scriptSig does not need to be re-evaluated. Empirical tests show that this results in new block validation being 40-50% faster.
- LevelDB has been upgraded to version 1.20 (See [PR 10544](#)). This version contains hardware acceleration for CRC on architectures supporting SSE 4.2. As a result, synchronization and block validation are now faster.
- SHA256 hashing has been optimized for architectures supporting SSE 4 (See [PR 10821](#)). SHA256 is around 50% faster on supported hardware, which results in around 5% faster IBD and block validation. In version 0.15, SHA256 hardware optimization is disabled in release builds by default, but can be enabled by using `-enable-experimental-asm` when building.
- Refill of the keypool no longer flushes the wallet between each key which resulted in a ~20x speedup in creating a new wallet. Part of this speedup was used to increase the default keypool to 1000 keys to make recovery more robust. (See [PR 10831](#)).

## Fee Estimation Improvements

Fee estimation has been significantly improved in version 0.15, with more accurate fee estimates used by the wallet and a wider range of options for advanced users of the `estimatesmartfee` and `estimatorawfee` RPCs (See [PR 10199](#)).

### Changes to internal logic and wallet behavior

- Internally, estimates are now tracked on 3 different time horizons. This allows for longer targets and means estimates adjust more quickly to changes in conditions.
- Estimates can now be *conservative* or *economical*. *Conservative* estimates use longer time horizons to produce an estimate which is less susceptible to rapid changes in fee conditions. *Economical* estimates use shorter time horizons and will be more affected by short-term changes in fee conditions. Economical estimates may be considerably lower during periods of low transaction activity (for example over weekends), but may result in transactions being unconfirmed if prevailing fees increase rapidly.
- By default, the wallet will use conservative fee estimates to increase the reliability of transactions being confirmed within the desired target. For transactions that are marked as replaceable, the wallet will use an economical estimate by default, since the fee can be 'bumped' if the fee conditions change rapidly (See [PR 10589](#)).
- Estimates can now be made for confirmation targets up to 1008 blocks (one week).
- More data on historical fee rates is stored, leading to more precise fee estimates.
- Transactions which leave the mempool due to eviction or other non-confirmed reasons are now taken into account by the fee estimation logic, leading to more accurate fee estimates.
- The fee estimation logic will make sure enough data has been gathered to return a meaningful estimate. If there is insufficient data, a fallback default fee is used.

### Changes to fee estimate RPCs

- The `estimatefee` RPC is now deprecated in favor of using only `estimatesmartfee` (which is the implementation used by the GUI)
- The `estimatesmartfee` RPC interface has been changed (See [PR 10707](#)):
  - The `nblocks` argument has been renamed to `conf_target` (to be consistent with other RPC methods).
  - An `estimate_mode` argument has been added. This argument takes one of the following strings: `CONSERVATIVE`, `ECONOMICAL` or `UNSET` (which defaults to `CONSERVATIVE`).
  - The RPC return object now contains an `errors` member, which returns errors encountered during processing.
  - If Bitcoin Core has not been running for long enough and has not seen enough blocks or transactions to produce an accurate fee estimation, an error will be returned (previously a value of -1 was used to indicate an error, which could be confused for a feerate).
- A new `estimatorawfee` RPC is added to provide raw fee data. External clients can query and use this data in their own fee estimation logic.

## Multi-wallet support

Bitcoin Core now supports loading multiple, separate wallets (See [PR 8694](#), [PR 10849](#)). The wallets are completely separated, with individual balances, keys and received transactions.

Multi-wallet is enabled by using more than one `-wallet` argument when starting Bitcoin, either on the command line or in the Bitcoin config file.

**In Bitcoin-Qt, only the first wallet will be displayed and accessible for creating and signing transactions.** GUI selectable multiple wallets will be supported in a future version. However, even in 0.15 other loaded wallets will remain synchronized to the node's current tip in the background. This can be useful if running a pruned node, since loading a wallet where the most recent sync is beyond the pruned height results in having to download and revalidate the whole blockchain. Continuing to synchronize all wallets in the background avoids this problem.

Bitcoin Core 0.15.0 contains the following changes to the RPC interface and `bitcoin-cli` for multi-wallet:

- When running Bitcoin Core with a single wallet, there are **no** changes to the RPC interface or `bitcoin-cli`. All RPC calls and `bitcoin-cli` commands continue to work as before.
- When running Bitcoin Core with multi-wallet, all *node-level* RPC methods continue to work as before. HTTP RPC requests should be send to the normal `<RPC IP address>:<RPC port>` endpoint, and `bitcoin-cli` commands should be run as before. A *node-level* RPC method is any method which does not require access to the wallet.
- When running Bitcoin Core with multi-wallet, *wallet-level* RPC methods must specify the wallet for which they're intended in every request. HTTP RPC requests should be send to the `<RPC IP address>:<RPC port>/wallet/<wallet name>` endpoint, for example `127.0.0.1:8332/wallet/wallet1.dat`. `bitcoin-cli` commands should be run with a `-rpcwallet` option, for example `bitcoin-cli -rpcwallet=wallet1.dat getbalance`.
- A new *node-level* `listwallets` RPC method is added to display which wallets are currently loaded. The names returned by this method are the same as those used in the HTTP endpoint and for the `rpcwallet` argument.

Note that while multi-wallet is now fully supported, the RPC multi-wallet interface should be considered unstable for version 0.15.0, and there may backwards-incompatible changes in future versions.

## Replace-by-fee control in the GUI

Bitcoin Core has supported creating opt-in replace-by-fee (RBF) transactions since version 0.12.0, and since version 0.14.0 has included a `bumpfee` RPC method to replace unconfirmed opt-in RBF transactions with a new transaction that pays a higher fee.

In version 0.15, creating an opt-in RBF transaction and replacing the unconfirmed transaction with a higher-fee transaction are both supported in the GUI (See [PR 9592](#)).

## Removal of Coin Age Priority

In previous versions of Bitcoin Core, a portion of each block could be reserved for transactions based on the age and value of UTXOs they spent. This concept (Coin Age Priority) is a policy choice by miners, and there are no consensus rules around the inclusion of Coin Age Priority transactions in blocks. In practice, only a few miners continue to use Coin Age Priority for transaction selection in blocks. Bitcoin Core 0.15 removes all remaining support for Coin Age Priority (See [PR 9602](#)). This has the following implications:

- The concept of *free transactions* has been removed. High Coin Age Priority transactions would previously be allowed to be relayed even if they didn't attach a miner fee. This is no longer possible since there is no concept of Coin Age Priority. The `-limitfreerelay` and `-relaypriority` options which controlled relay of free transactions have therefore been removed.
- The `-sendfreetransactions` option has been removed, since almost all miners do not include transactions which do not attach a transaction fee.
- The `-blockprioritysize` option has been removed.
- The `estimatepriority` and `estimatesmartpriority` RPCs have been removed.

- The `getmempoolancestors`, `getmempooldescendants`, `getmempoolentry` and `getrawmempool` RPCs no longer return `startingpriority` and `currentpriority`.
- The `prioritisetransaction` RPC no longer takes a `priority_delta` argument, which is replaced by a `dummy` argument for backwards compatibility with clients using positional arguments. The RPC is still used to change the apparent fee-rate of the transaction by using the `fee_delta` argument.
- `-minrelaytxfee` can now be set to 0. If `minrelaytxfee` is set, then fees smaller than `minrelaytxfee` (per kB) are rejected from relaying, mining and transaction creation. This defaults to 1000 satoshi/kB.
- The `-printpriority` option has been updated to only output the fee rate and hash of transactions included in a block by the mining code.

## Mempool Persistence Across Restarts

Version 0.14 introduced mempool persistence across restarts (the mempool is saved to a `mempool.dat` file in the data directory prior to shutdown and restores the mempool when the node is restarted). Version 0.15 allows this feature to be switched on or off using the `-persistmempool` command-line option (See [PR 9966](#)). By default, the option is set to true, and the mempool is saved on shutdown and reloaded on startup. If set to false, the `mempool.dat` file will not be loaded on startup or saved on shutdown.

## New RPC methods

Version 0.15 introduces several new RPC methods:

- `abortrescan` stops current wallet rescan, e.g. when triggered by an `importprivkey` call (See [PR 10208](#)).
- `combinerawtransaction` accepts a JSON array of raw transactions and combines them into a single raw transaction (See [PR 10571](#)).
- `estimatorawfee` returns raw fee data so that customized logic can be implemented to analyze the data and calculate estimates. See [Fee Estimation Improvements](#) for full details on changes to the fee estimation logic and interface.
- `getchaintxstats` returns statistics about the total number and rate of transactions in the chain (See [PR 9733](#)).
- `listwallets` lists wallets which are currently loaded. See the *Multi-wallet* section of these release notes for full details (See [Multi-wallet support](#)).
- `uptime` returns the total runtime of the `bitcoind` server since its last start (See [PR 10400](#)).

## Low-level RPC changes

- When using Bitcoin Core in multi-wallet mode, RPC requests for wallet methods must specify the wallet that they're intended for. See [Multi-wallet support](#) for full details.
- The new database model no longer stores information about transaction versions of unspent outputs (See [Performance improvements](#)). This means that:
  - The `gettxout` RPC no longer has a `version` field in the response.
  - The `gettxoutsetinfo` RPC reports `hash_serialized_2` instead of `hash_serialized`, which does not commit to the transaction versions of unspent outputs, but does commit to the height and coinbase information.
  - The `getutxos` REST path no longer reports the `txvers` field in JSON format, and always reports 0 for transaction versions in the binary format

- The `estimatefee` RPC is deprecated. Clients should switch to using the `estimatesmartfee` RPC, which returns better fee estimates. See [Fee Estimation Improvements](#) for full details on changes to the fee estimation logic and interface.
- The `gettxoutsetinfo` response now contains `disk_size` and `bogosize` instead of `bytes_serialized`. The first is a more accurate estimate of actual disk usage, but is not deterministic. The second is unrelated to disk usage, but is a database-independent metric of UTXO set size: it counts every UTXO entry as 50 + the length of its scriptPubKey (See [PR 10426](#)).
- `signrawtransaction` can no longer be used to combine multiple transactions into a single transaction. Instead, use the new `combinerawtransaction` RPC (See [PR 10571](#)).
- `fundrawtransaction` no longer accepts a `reserveChangeKey` option. This option used to allow RPC users to fund a raw transaction using a key from the keypool for the change address without removing it from the available keys in the keypool. The key could then be re-used for a `getnewaddress` call, which could potentially result in confusing or dangerous behaviour (See [PR 10784](#)).
- `estimatepriority` and `estimatesmartpriority` have been removed. See [Removal of Coin Age Priority](#).
- The `listunspent` RPC now takes a `query_options` argument (see [PR 8952](#)), which is a JSON object containing one or more of the following members:
  - `minimumAmount` - a number specifying the minimum value of each UTXO
  - `maximumAmount` - a number specifying the maximum value of each UTXO
  - `maximumCount` - a number specifying the maximum number of UTXOs
  - `minimumSumAmount` - a number specifying the minimum sum value of all UTXOs
- The `getmempoolancestors`, `getmempooldescendants`, `getmempoolentry` and `getrawmempool` RPCs no longer return `startingpriority` and `currentpriority`. See [Removal of Coin Age Priority](#).
- The `dumpwallet` RPC now returns the full absolute path to the dumped wallet. It used to return no value, even if successful (See [PR 9740](#)).
- In the `getpeerinfo` RPC, the return object for each peer now returns an `addrbind` member, which contains the ip address and port of the connection to the peer. This is in addition to the `addrlocal` member which contains the ip address and port of the local node as reported by the peer (See [PR 10478](#)).
- The `disconnectnode` RPC can now disconnect a node specified by node ID (as well as by IP address/port). To disconnect a node based on node ID, call the RPC with the new `nodeid` argument (See [PR 10143](#)).
- The second argument in `prioritisetransaction` has been renamed from `priority_delta` to `dummy` since Bitcoin Core no longer has a concept of coin age priority. The `dummy` argument has no functional effect, but is retained for positional argument compatibility. See [Removal of Coin Age Priority](#).
- The `resendwallettransactions` RPC throws an error if the `-walletbroadcast` option is set to false (See [PR 10995](#)).
- The second argument in the `submitblock` RPC argument has been renamed from `parameters` to `dummy`. This argument never had any effect, and the renaming is simply to communicate this fact to the

user (See [PR 10191](#)) (Clients should, however, use positional arguments for `submitblock` in order to be compatible with BIP 22.)

- The `verbose` argument of `getblock` has been renamed to `verbosity` and now takes an integer from 0 to 2. Verbose level 0 is equivalent to `verbose=false`. Verbose level 1 is equivalent to `verbose=true`. Verbose level 2 will give the full transaction details of each transaction in the output as given by `getrawtransaction`. The old behavior of using the `verbose` named argument and a boolean value is still maintained for compatibility.
- Error codes have been updated to be more accurate for the following error cases (See [PR 9853](#)):
  - `getblock` now returns `RPC_MISC_ERROR` if the block can't be found on disk (for example if the block has been pruned). Previously returned `RPC_INTERNAL_ERROR`.
  - `pruneblockchain` now returns `RPC_MISC_ERROR` if the blocks cannot be pruned because the node is not in pruned mode. Previously returned `RPC_METHOD_NOT_FOUND`.
  - `pruneblockchain` now returns `RPC_INVALID_PARAMETER` if the blocks cannot be pruned because the supplied timestamp is too late. Previously returned `RPC_INTERNAL_ERROR`.
  - `pruneblockchain` now returns `RPC_MISC_ERROR` if the blocks cannot be pruned because the blockchain is too short. Previously returned `RPC_INTERNAL_ERROR`.
  - `setban` now returns `RPC_CLIENT_INVALID_IP_OR_SUBNET` if the supplied IP address or subnet is invalid. Previously returned `RPC_CLIENT_NODE_ALREADY_ADDED`.
  - `setban` now returns `RPC_CLIENT_INVALID_IP_OR_SUBNET` if the user tries to unban a node that has not previously been banned. Previously returned `RPC_MISC_ERROR`.
  - `removeprunedfunds` now returns `RPC_WALLET_ERROR` if `bitcoind` is unable to remove the transaction. Previously returned `RPC_INTERNAL_ERROR`.
  - `removeprunedfunds` now returns `RPC_INVALID_PARAMETER` if the transaction does not exist in the wallet. Previously returned `RPC_INTERNAL_ERROR`.
  - `fundrawtransaction` now returns `RPC_INVALID_ADDRESS_OR_KEY` if an invalid change address is provided. Previously returned `RPC_INVALID_PARAMETER`.
  - `fundrawtransaction` now returns `RPC_WALLET_ERROR` if `bitcoind` is unable to create the transaction. The error message provides further details. Previously returned `RPC_INTERNAL_ERROR`.
  - `bumpfee` now returns `RPC_INVALID_PARAMETER` if the provided transaction has descendants in the wallet. Previously returned `RPC_MISC_ERROR`.
  - `bumpfee` now returns `RPC_INVALID_PARAMETER` if the provided transaction has descendants in the mempool. Previously returned `RPC_MISC_ERROR`.
  - `bumpfee` now returns `RPC_WALLET_ERROR` if the provided transaction has been mined or conflicts with a mined transaction. Previously returned `RPC_INVALID_ADDRESS_OR_KEY`.
  - `bumpfee` now returns `RPC_WALLET_ERROR` if the provided transaction is not BIP 125 replaceable. Previously returned `RPC_INVALID_ADDRESS_OR_KEY`.
  - `bumpfee` now returns `RPC_WALLET_ERROR` if the provided transaction has already been bumped by a different transaction. Previously returned `RPC_INVALID_REQUEST`.
  - `bumpfee` now returns `RPC_WALLET_ERROR` if the provided transaction contains inputs which don't belong to this wallet. Previously returned `RPC_INVALID_ADDRESS_OR_KEY`.
  - `bumpfee` now returns `RPC_WALLET_ERROR` if the provided transaction has multiple change outputs. Previously returned `RPC_MISC_ERROR`.
  - `bumpfee` now returns `RPC_WALLET_ERROR` if the provided transaction has no change output. Previously returned `RPC_MISC_ERROR`.

- `bumpfee` now returns `RPC_WALLET_ERROR` if the fee is too high. Previously returned `RPC_MISC_ERROR`.
- `bumpfee` now returns `RPC_WALLET_ERROR` if the fee is too low. Previously returned `RPC_MISC_ERROR`.
- `bumpfee` now returns `RPC_WALLET_ERROR` if the change output is too small to bump the fee. Previously returned `RPC_MISC_ERROR`.

## 0.15.0 Change log

### RPC and other APIs

- #9485 61a640e ZMQ example using python3 and asyncio (mcelrath)
- #9894 0496e15 remove 'label' filter for rpc command help (instagibbs)
- #9853 02bd6e9 Fix error codes from various RPCs (jnewbery)
- #9842 598ef9c Fix RPC failure testing (continuation of #9707) (jnewbery)
- #10038 d34995a Add mallocinfo mode to `getmemoryinfo` RPC (laanwj)
- #9500 3568b30 [Qt][RPC] Autocomplete commands for 'help' command in debug console (achow101)
- #10056 e6156a0 [zmq] Call `va_end()` on `va_start()`ed args (kallewoof)
- #10086 7438cea Trivial: move `rpcserialversion` into RPC option group (jlopp)
- #10150 350b224 [rpc] Add logging rpc (jnewbery)
- #10208 393160c [wallet] Rescan abortability (kallewoof)
- #10143 a987def [net] Allow `disconnectnode` RPC to be called with node id (jnewbery)
- #10281 0e8499c doc: Add RPC interface guidelines (laanwj)
- #9733 d4732f3 Add `getchaintxstats` RPC (sipa)
- #10310 f4b15e2 [doc] Add hint about `getmempoolentry` to `getrawmempool` help (kallewoof)
- #8704 96c850c [RPC] Transaction details in `getblock` (achow101)
- #8952 9390845 Add query options to `listunspent` RPC call (pedrobranco)
- #10413 08ac35a Fix docs (there's no rpc command `setpaytxfee`) (RHavar)
- #8384 e317c0d Add witness data output to `TxnError` messages (instagibbs)
- #9571 4677151 RPC: `getblockchaininfo` returns BIP signaling statistics (pinheadmz)
- #10450 ef2d062 Fix `bumpfee` rpc "errors" return value (ryanofsky)
- #10475 39039b1 [RPC] `getmempoolinfo` `mempoolminfee` is a BTC/KB feerate (instagibbs)
- #10478 296928e rpc: Add listen address to incoming connections in `getpeerinfo` (laanwj)
- #10403 08d0390 Fix `importmulti` failure to return rescan errors (ryanofsky)
- #9740 9fec4da Add friendly output to `dumpwallet` (aideca)
- #10426 16f6c98 Replace `bytes_serialized` with `bogosity` (sipa)
- #10252 980deaf RPC/Mining: Restore API compatibility for `prioritizetransaction` (luke-jr)
- #9672 46311e7 Opt-into-RBF for RPC & bitcoin-tx (luke-jr)
- #10481 9c248e3 Decodehex scripts sanity check (achow101)
- #10488 fa1f106 Note that the `prioritizetransaction` dummy value is deprecated, and has no meaning (TheBlueMatt)
- #9738 c94b89e `gettxoutproof()` should return consistent result (jnewbery)
- #10191 00350bd [trivial] Rename unused RPC arguments 'dummy' (jnewbery)
- #10627 b62b4c8 fixed `listunspent` rpc convert parameter (tnakagawa)
- #10412 bef02fb Improve wallet rescan API (ryanofsky)
- #10400 1680ee0 [RPC] Add an uptime command that displays the amount of time (in seconds) bitcoind has been running (rvelhote)



- #10683 d81bec7 rpc: Move the `generate` RPC call to rpcwallet (laanwj)
- #10710 30bc0f6 REST/RPC example update (Mirobit)
- #10747 9edda0c [rpc] fix verbose argument for getblock in bitcoin-cli (jnewbery)
- #10589 104f5f2 More economical fee estimates for RBF and RPC options to control (morcos)
- #10543 b27b004 Change API to estimatorawfee (morcos)
- #10807 afd2fca getbalance example covers at least 6 confirms (instagibbs)
- #10707 75b5643 Better API for estimatesmartfee RPC (morcos)
- #10784 9e8d6a3 Do not allow users to get keys from keypool without reserving them (TheBlueMatt)
- #10857 d445a2c [RPC] Add a deprecation warning to getinfo's output (achow101)
- #10571 adf170d [RPC] Move transaction combining from signrawtransaction to new RPC (achow101)
- #10783 041dad9 [RPC] Various rpc argument fixes (instagibbs)
- #9622 6ef3c7e [rpc] listsinceblock should include lost transactions when parameter is a reorg'd block (kallewoof)
- #10799 8537187 Prevent user from specifying conflicting parameters to fundrawtx (TheBlueMatt)
- #10931 0b11a07 Fix misleading "Method not found" multiwallet errors (ryanofsky)
- #10788 f66c596 [RPC] Fix addwitnessaddress by replacing ismine with producesignature (achow101)
- #10999 627c3c0 Fix amounts formatting in `decoderawtransaction` (laanwj)
- #11002 4268426 [wallet] return correct error code from resendwallettransaction (jnewbery)
- #11029 96a63a3 [RPC] trivial: gettxout no longer shows version of tx (FelixWeis)
- #11083 6c2b008 Fix combinerawtransaction RPC help result section (jonasnick)
- #11027 07164bb [RPC] Only return hex field once in getrawtransaction (achow101)
- #10698 5af6572 Be consistent in calling transactions "replaceable" for Opt-In RBF (TheBlueMatt)

## Block and transaction handling

- #9801 a8c5751 Removed redundant parameter from mempool.PrioritiseTransaction (gubatron)
- #9819 1efc99c Remove harmless read of unused priority estimates (morcos)
- #9822 b7547fa Remove block file location upgrade code (benma)
- #9602 30ff3a2 Remove coin age priority and free transactions - implementation (morcos)
- #9548 47510ad Remove min reasonable fee (morcos)
- #10249 c73af54 Switch CCoinsMap from boost to std unordered\_map (sipa)
- #9966 2a183de Control mempool persistence using a command line parameter (jnewbery)
- #10199 318ea50 Better fee estimates (morcos)
- #10196 bee3529 Bugfix: PrioritiseTransaction updates the mempool tx counter (sdaftuar)
- #10195 1088b02 Switch chainstate db and cache to per-txout model (sipa)
- #10284 c2ab38b Always log debug information for fee calculation in CreateTransaction (morcos)
- #10503 efbcf2b Use REJECT\_DUPLICATE for already known and conflicted txn (sipa)
- #10537 b3eb0d6 Few Minor per-utxo assert-semantics re-adds and tweak (TheBlueMatt)
- #10626 8c841a3 doc: Remove outdated minrelaytxfee comment (MarcoFalke)
- #10559 234ffc6 Change semantics of HaveCoinInCache to match HaveCoin (morcos)
- #10581 7878353 Simplify return values of GetCoin/HaveCoin(InCache) (sipa)
- #10684 a381f6a Remove no longer used mempool.exists(outpoint) (morcos)
- #10148 d4e551a Use non-atomic flushing with block replay (sipa)
- #10685 30c2130 Clarify CCoinsViewMemPool documentation (TheBlueMatt)
- #10558 90a002e Address nits from per-utxo change (morcos)
- #10706 6859ad2 Improve wallet fee logic and fix GUI bugs (morcos)
- #10526 754aa02 Force on-the-fly compaction during perxtout upgrade (sipa)
- #10985 d896d5c Add undocumented -forcecompactdb to force LevelDB compactions (sipa)

- #10292 e4bbd3d Improved efficiency in COutPoint constructors (mm-s)
- #10290 8d6d43e Add -stopatheight for benchmarking (sipa)

## P2P protocol and network code

- #9726 7639d38 netbase: Do not print an error on connection timeouts through proxy (laanwj)
- #9805 5b583ef Add seed.btc.petertodd.org to mainnet DNS seeds (petertodd)
- #9861 22f609f Trivial: Debug log ambiguity fix for peer addrs (keystrike)
- #9774 90cb2a2 Enable host lookups for -proxy and -onion parameters (jmcorgan)
- #9558 7b585cf Clarify assumptions made about when BlockCheck is called (TheBlueMatt)
- #10135 e19586a [p2p] Send the correct error code in reject messages (jnewbery)
- #9665 eab00d9 Use cached [compact] blocks to respond to getdata messages (TheBlueMatt)
- #10215 a077a90 Check interruptNet during dnsseed lookups (TheBlueMatt)
- #10234 faf2dea [net] listbanned RPC and QT should show correct banned subnets (jnewbery)
- #10134 314ebdf [qa] Fixes segwit block relay test after inv-direct-fetch was disabled (sdaftuar)
- #10351 3f57c55 removed unused code in INV message (Greg-Griffith)
- #10061 ae78609 [net] Added SetSocketNoDelay() utility function (tjps)
- #10408 28c6e8d Net: Improvements to Tor control port parser (str4d)
- #10460 5c63d66 Broadcast address every day, not 9 hours (sipa)
- #10471 400fdd0 Denote functions CNode::GetRecvVersion() and CNode::GetRefCount() as const (pavlosantoniou)
- #10345 67700b3 [P2P] Timeout for headers sync (sdaftuar)
- #10564 8d9f45e Return early in IsBanned (gmaxwell)
- #10587 de8db47 Net: Fix resource leak in ReadBinaryFile(...) (practicalswift)
- #9549 b33ca14 [net] Avoid possibility of NULL pointer dereference in MarkBlockAsInFlight(...) (practicalswift)
- #10446 2772dc9 net: avoid extra dns query per seed (theuni)
- #10824 9dd6a2b Avoid unnecessary work in SetNetworkActive (promag)
- #10948 df3a6f4 p2p: Hardcoded seeds update pre-0.15 branch (laanwj)
- #10977 02f4c4a [net] Fix use of uninitialized value in getnetworkinfo(const JSONRPCRequest&) (practicalswift)
- #10982 c8b62c7 Disconnect network service bits 6 and 8 until Aug 1, 2018 (TheBlueMatt)
- #11012 0e5cff6 Make sure to clean up mapBlockSource if we've already seen the block (theuni)

## Validation

- #9725 67023e9 CValidationInterface Cleanups (TheBlueMatt)
- #10178 2584925 Remove CValidationInterface::UpdatedTransaction (TheBlueMatt)
- #10201 a6548a4 pass Consensus::Params& to functions in validation.cpp and make them static (mariodian)
- #10297 431a548 Simplify DisconnectBlock arguments/return value (sipa)
- #10464 f94b7d5 Introduce static DoWarning (simplify UpdateTip) (jtimon)
- #10569 2e7d8f8 Fix stopatheight (achow101)
- #10192 2935b46 Cache full script execution results in addition to signatures (TheBlueMatt)
- #10179 21ed30a Give CValidationInterface Support for calling notifications on the CScheduler Thread (TheBlueMatt)
- #10557 66270a4 Make check to distinguish between orphan txs and old txs more efficient (morcos)
- #10775 7c2400c nCheckDepth chain height fix (romanornr)
- #10821 16240f4 Add SSE4 optimized SHA256 (sipa)

- #10854 04d395e Avoid using sizes on non-fixed-width types to derive protocol constants (gmaxwell)
- #10945 2a50b11 Update defaultAssumeValid according to release-process.md (gmaxwell)
- #10986 2361208 Update chain transaction statistics (sipa)
- #11028 6bdf4b3 Avoid masking of difficulty adjustment errors by checkpoints (sipa)
- #9533 cb598cf Allow non-power-of-2 signature cache sizes (sipa)
- #9208 acd9957 Improve DisconnectTip performance (sdaftuar)
- #10618 f90603a Remove confusing MAX\_BLOCK\_BASE\_SIZE (gmaxwell)
- #10758 bd92424 Fix some chainstate-init-order bugs (TheBlueMatt)
- #10550 b7296bc Don't return stale data from CCoinsViewCache::Cursor() (ryanofsky)
- #10998 2507fd5 Fix upgrade cancel warnings (TheBlueMatt)
- #9868 cbbdb473 Abstract out the command line options for block assembly (sipa)

## Build system

- #9727 5f0556d Remove fallbacks for boost\_filesystem < v3 (laanwj)
- #9788 50a2265 gitian: bump descriptors for master (theuni)
- #9794 7ca2f54 Minor update to qrencode package builder (mitchellcash)
- #9514 2cc0df1 release: Windows signing script (theuni)
- #9921 8b789d8 build: Probe MSG\_DONTWAIT in the same way as MSG\_NOSIGNAL (laanwj)
- #10011 32d1b34 build: Fix typo s/HAVE\_DONTWAIT/HAVE\_MSG\_DONTWAIT (laanwj)
- #9946 90dd9e6 Fix build errors if spaces in path or parent directory (pinheadmz)
- #10136 81da4c7 build: Disable Wshadow warning (laanwj)
- #10166 64962ae Ignore Doxyfile generated from Doxyfile.in template (paveljanik)
- #10239 0416ea9 Make Boost use std::atomic internally (sipa)
- #10228 27faa6c build: regenerate bitcoin-config.h as necessary (theuni)
- #10273 8979f45 [scripts] Minor improvements to macdeployqtplus script (chrisgavin)
- #10325 a26280b 0.15.0 Depends Updates (fanquake)
- #10328 79aef6f Update contrib/debian to latest Ubuntu PPA upload (TheBlueMatt)
- #7522 d25449f Bugfix: Only use git for build info if the repository is actually the right one (luke-jr)
- #10489 e654d61 build: silence gcc7's implicit fallthrough warning (theuni)
- #10549 ad1a13e Avoid printing generic and duplicated "checking for QT" during ./configure (drizzt)
- #10628 8465b68 [depends] expat 2.2.1 (fanquake)
- #10806 db825d2 build: verify that the assembler can handle crc32 functions (theuni)
- #10766 b4d03be Building Environment: Set ARFLAGS to cr (ReneNyffenegger)
- #10803 91edda8 Explicitly search for bdb5.3 (pstratem)
- #10855 81560b0 random: only use getentropy on openbsd (theuni)
- #10508 1caafa6 Run Qt wallet tests on travis (ryanofsky)
- #10851 e222618 depends: fix fontconfig with newer glibc (theuni)
- #10971 88b1e4b build: fix missing sse42 in depends builds (theuni)
- #11097 129b03f gitian: quick hack to fix version string in releases (theuni)
- #10039 919aaf6 Fix compile errors with Qt 5.3.2 and Boost 1.55.0 (ryanofsky)
- #10168 7032021 Fix build warning from #error text (jnewbery)
- #10301 318392c Check if sys/random.h is required for getentropy (jameshilliard)

## GUI

- #9724 1a9fd5c Qt/Intro: Add explanation of IBD process (luke-jr)
- #9834 b00ba62 qt: clean up initialize/shutdown signals (benma)
- #9481 ce01e62 [Qt] Show more significant warning if we fall back to the default fee (jonasschnelli)

- #9974 b9f930b Add basic Qt wallet test (ryanofsky)
- #9690 a387d3a Change 'Clear' button string to 'Reset' (da2x)
- #9592 9c7b7cf [Qt] Add checkbox in the GUI to opt-in to RBF when creating a transaction (ryanofsky)
- #10098 2b477e6 Make qt wallet test compatible with qt4 (ryanofsky)
- #9890 1fa4ae6 Add a button to open the config file in a text editor (ericshawlinux)
- #10156 51833a1 Fix for issues with startup and multiple monitors on windows (AllanDoensen)
- #10177 de01da7 Changed "Send" button default status from true to false (KibbledJiveElkZoo)
- #10221 e96486c Stop treating coinbase outputs differently in GUI: show them at 1conf (TheBlueMatt)
- #10231 987a6c0 [Qt] Reduce a significant cs\_main lock freeze (jonasschnelli)
- #10242 f6f3b58 [qt] Don't call method on null WalletModel object (ryanofsky)
- #10093 a3e756b [Qt] Don't add arguments of sensitive command to console window (jonasschnelli)
- #10362 95546c8 [GUI] Add OSX keystroke to RPCConsole info (spencerlievens)
- #9697 962cd3f [Qt] simple fee bumper with user verification (jonasschnelli)
- #10390 e477516 [wallet] remove minimum total fee option (instagibbs)
- #10420 4314544 Add Qt tests for wallet spends & bumpfee (ryanofsky)
- #10454 c1c9a95 Fix broken q4 test build (ryanofsky)
- #10449 64beb13 Overhaul Qt fee bumper (jonasschnelli)
- #10582 7c72fb9 Pass in smart fee slider value to coin control dialog (morcos)
- #10673 4c72cc3 [qt] Avoid potential null pointer dereference in TransactionView::exportClicked() (practicalswift)
- #10769 8fdd23a [Qt] replace fee slider with a Dropdown, extend conf. targets (jonasschnelli)
- #10870 412b466 [Qt] Use wallet 0 in rpc console if running with multiple wallets (jonasschnelli)
- #10988 a9dd111 qt: Increase BLOCK\_CHAIN\_SIZE constants (laanwj)
- #10644 e292140 Slightly overhaul NSI pixmaps (jonasschnelli)
- #10660 0c3542e Allow to cancel the txdb upgrade via splashscreen keypress 'q' (jonasschnelli)

## Wallet

- #9359 f7ec7cf Add test for CWalletTx::GetImmatureCredit() returning stale values (ryanofsky)
- #9576 56ab672 [wallet] Remove redundant initialization (practicalswift)
- #9333 fa625b0 Document CWalletTx::mapValue entries and remove erase of nonexistent "version" entry (ryanofsky)
- #9906 72fb515 Disallow copy constructor CReserveKeys (instagibbs)
- #9369 3178b2c Factor out CWallet::nTimeSmart computation into a method (ryanofsky)
- #9830 afcd7c0 Add safe flag to listunspent result (NicolasDorier)
- #9993 c49355c Initialize nRelockTime (pstratem)
- #9818 3d857f3 Save watch only key timestamps when reimporting keys (ryanofsky)
- #9294 f34cdcb Use internal HD chain for change outputs (hd split) (jonasschnelli)
- #10164 e183ea2 Wallet: reduce excess logic InMempool() (kewde)
- #10186 c9ff4f8 Remove SYNC\_TRANSACTION\_NOT\_IN\_BLOCK magic number (jnewbery)
- #10226 64c45aa wallet: Use boost to more portably ensure -wallet specifies only a filename (luke-jr)
- #9827 c91ca0a Improve ScanForWalletTransactions return value (ryanofsky)
- #9951 fa1ac28 Wallet database handling abstractions/simplifications (laanwj)
- #10265 c29a0d4 [wallet] [moveonly] Check non-null pindex before potentially referencing (kallewoof)
- #10283 a550f6e Cleanup: reduce to one GetMinimumFee call signature (morcos)
- #10294 e2b99b1 [Wallet] unset change position when there is no change (instagibbs)
- #10115 d3dce0e Avoid reading the old hd master key during wallet encryption (TheBlueMatt)

- #10341 18c9deb rpc/wallet: Workaround older UniValue which returns a std::string temporary for get\_str (luke-jr)
- #10308 94e5227 [wallet] Securely erase potentially sensitive keys/values (tjps)
- #10257 ea1fd43 [test] Add test for getmemoryinfo (jimmysong)
- #10295 ce8176d [qt] Move some WalletModel functions into CWallet (ryanofsky)
- #10506 7cc2c67 Fix bumpfee test after #10449 (ryanofsky)
- #10500 098b01d Avoid CWalletTx copies in GetAddressBalances and GetAddressGroupings (ryanofsky)
- #10455 0747d33 Simplify feebumper minimum fee code slightly (ryanofsky)
- #10522 2805d60 [wallet] Remove unused variables (practicalswift)
- #8694 177433a Basic multiwallet support (luke-jr)
- #10598 7a74f88 Suppress struct/class mismatch warnings introduced in #10284 (paveljanik)
- #9343 209eef6 Don't create change at dust limit (morcos)
- #10744 ed88e31 Use method name via **func** macro (darksh1ne)
- #10712 e8b9523 Add change output if necessary to reduce excess fee (morcos)
- #10816 1c011ff Properly forbid -salvagewallet and -zapwallettxes for multi wallet (morcos)
- #10235 5cfdda2 Track keypool entries as internal vs external in memory (TheBlueMatt)
- #10330 bf0a08b [wallet] fix zapwallettxes interaction with persistent mempool (jnewbery)
- #10831 0b01935 Batch flushing operations to the walletdb during top up and increase keypool size (gmaxwell)
- #10795 7b6e8bc No longer ever reuse keypool indexes (TheBlueMatt)
- #10849 bde4f93 Multiwallet: simplest endpoint support (jonasschnelli)
- #10817 9022aa3 Redefine Dust and add a discard\_rate (morcos)
- #10883 bf3b742 Rename -usewallet to -rpcwallet (morcos)
- #10604 420238d [wallet] [tests] Add listwallets RPC, include wallet name in getwalletinfo and add multiwallet test (jnewbery)
- #10885 70888a3 Reject invalid wallets (promag)
- #10949 af56397 Clarify help message for -discardfee (morcos)
- #10942 2e857bb Eliminate fee overpaying edge case when subtracting fee from recipients (morcos)
- #10995 fa64636 Fix resendwallettransactions assert failure if -walletbroadcast=0 (TheBlueMatt)
- #11022 653a46d Basic keypool topup (jnewbery)
- #11081 9fe1f6b Add length check for CExtKey deserialization (jonasschnelli, guidovranken)
- #11044 4ef8374 [wallet] Keypool topup cleanups (jnewbery)
- #11145 e51bb71 Fix rounding bug in calculation of minimum change (morcos)
- #9605 779f2f9 Use CScheduler for wallet flushing, remove ThreadFlushWalletDB (TheBlueMatt)
- #10108 4e3efd4 ApproximateBestSubset should take inputs by reference, not value (RHavar)

## Tests and QA

- #9744 8efd1c8 Remove unused module from rpc-tests (34ro)
- #9657 7ff4a53 Improve rpc-tests.py (jnewbery)
- #9766 7146d96 Add --exclude option to rpc-tests.py (jnewbery)
- #9577 d6064a8 Fix docstrings in qa tests (jnewbery)
- #9823 a13a417 qa: Set correct path for binaries in rpc tests (MarcoFalke)
- #9847 6206252 Extra test vector for BIP32 (sipa)
- #9350 88c2ae3 [Trivial] Adding label for amount inside of tx\_valid/tx\_invalid.json (Christewart)
- #9888 36afd4d travis: Verify commits only for one target (MarcoFalke)
- #9904 58861ad test: Fail if InitBlockIndex fails (laanwj)
- #9828 67c5cc1 Avoid -Wshadow warnings in wallet\_tests (ryanofsky)

- #9832 48c3429 [qa] assert\_start\_raises\_init\_error (NicolasDorier)
- #9739 9d5fcbf Fix BIP68 activation test (jnewbery)
- #9547 d32581c bench: Assert that division by zero is unreachable (practicalswift)
- #9843 c78adbef Fix segwit getblocktemplate test (jnewbery)
- #9929 d5ce14e tests: Delete unused function \_rpchost\_to\_args (laanwj)
- #9555 19be26a [test] Avoid reading a potentially uninitialized variable in tx\_invalid-test (transaction\_tests.cpp) (practicalswift)
- #9945 ac23a7c Improve logging in bctest.py if there is a formatting mismatch (jnewbery)
- #9768 8910b47 [qa] Add logging to test\_framework.py (jnewbery)
- #9972 21833f9 Fix extended rpc tests broken by #9768 (jnewbery)
- #9977 857d1e1 QA: getblocktemplate\_longpoll.py should always use >0 fee tx (sdaftuar)
- #9970 3cc13ea Improve readability of segwit.py, smartfees.py (sdaftuar)
- #9497 2c781fb CCheckQueue Unit Tests (JeremyRubin)
- #10024 9225de2 [trivial] Use log.info() instead of print() in remaining functional test cases (jnewbery)
- #9956 3192e52 Reorganise qa directory (jnewbery)
- #10017 02d64bd combine\_logs.py - aggregates log files from multiple bitcoinds during functional tests (jnewbery)
- #10047 dfef6b6 [tests] Remove unused variables and imports (practicalswift)
- #9701 a230b05 Make bumpfee tests less fragile (ryanofsky)
- #10053 ca20923 [test] Allow functional test cases to be skipped (jnewbery)
- #10052 a0b1e57 [test] Run extended tests once daily in Travis (jnewbery)
- #10069 1118493 [QA] Fix typo in fundrawtransaction test (NicolasDorier)
- #10083 c044f03 [QA] Renaming rawTx into rawtx (NicolasDorier)
- #10073 b1a4f27 Actually run assumevalid.py (jnewbery)
- #9780 c412fd8 Suppress noisy output from qa tests in Travis (jnewbery)
- #10096 79af9fb Check that all test scripts in test/functional are being run (jnewbery)
- #10076 5b029aa [qa] combine\_logs: Use ordered list for logfiles (MarcoFalke)
- #10107 f2734c2 Remove unused variable. Remove accidental trailing semicolons in Python code (practicalswift)
- #10109 8ac8041 Remove SingleNodeConnCB (jnewbery)
- #10114 edc62c9 [tests] sync\_with\_ping should assert that ping hasn't timed out (jnewbery)
- #10128 427d2fd Speed Up CuckooCache tests (JeremyRubin)
- #10072 12af74b Remove sources of unreliability in extended functional tests (jnewbery)
- #10077 ebfd653 [qa] Add setnetworkactive smoke test (MarcoFalke)
- #10152 080d7c7 [trivial] remove unused line in Travis config (jnewbery)
- #10159 df1ca9e [tests] color test results and sort alphabetically (jnewbery)
- #10124 88799ea [test] Suppress test logging spam (jnewbery)
- #10142 ed09dd3 Run bitcoin\_test-qt under minimal QPA platform (ryanofsky)
- #9949 a27dbc5 [bench] Avoid function call arguments which are pointers to uninitialized values (practicalswift)
- #10187 b44adf9 tests: Fix test\_runner return value in case of skipped test (laanwj)
- #10197 d86bb07 [tests] Functional test warnings (jnewbery)
- #10219 9111df9 Tests: Order Python Tests Differently (jimmysong)
- #10229 f3db4c6 Tests: Add test for getdifficulty (jimmysong)
- #10224 2723bcd [test] Add test for getaddednodeinfo (jimmysong)
- #10023 c530c15 [tests] remove maxblocksinfight.py (functionality covered by other test) (jnewbery)
- #10097 1b25b6d Move zmq test skipping logic into individual test case (jnewbery)

- #10272 54e2d87 [Tests] Prevent warning: variable 'x' is uninitialized (paveljanik)
- #10225 e0a7e19 [test] Add aborttrescan tests (kallewoof)
- #10278 8254a8a [test] Add Unit Test for GetListenPort (jimmysong)
- #10280 47535d7 [test] Unit test amount.h/amount.cpp (jimmysong)
- #10256 80c3a73 [test] Add test for gettxout to wallet.py (jimmysong)
- #10264 492d22f [test] Add tests for getconnectioncount, getnettotals and ping (jimmysong)
- #10169 8f3e384 [tests] Remove func test code duplication (jnewbery)
- #10198 dc8fc0c [tests] Remove is\_network\_split from functional test framework (jnewbery)
- #10255 3c5e6c9 [test] Add test for listaddressgroupings (jimmysong)
- #10137 75171f0 Remove unused import. Remove accidental trailing semicolons (practicalswift)
- #10307 83073de [tests] allow zmq test to be run in out-of-tree builds (jnewbery)
- #10344 e927483 [tests] Fix abandonconflict.py intermittency (jnewbery)
- #10318 170bc2c [tests] fix wait\_for\_inv() (jnewbery)
- #10171 ff72de [tests] Add node methods to test framework (jnewbery)
- #10352 23d78c4 test: Add elapsed time to RPC tracing (laanwj)
- #10342 6a796b2 [tests] Improve mempool\_persist test (jnewbery)
- #10287 776ba23 [tests] Update Unit Test for addrman.h/addrman.cpp (jimmysong)
- #10365 7ee5236 [tests] increase timeouts in sendheaders test (jnewbery)
- #10361 f6241b3 qa: disablewallet: Check that wallet is really disabled (MarcoFalke)
- #10371 4b766fc [tests] Clean up addrman\_tests.cpp (jimmysong)
- #10253 87abe20 [test] Add test for getnetworkhashps (jimmysong)
- #10376 8bd16ee [tests] fix disconnect\_ban intermittency (jnewbery)
- #10374 5411997 qa: Warn when specified test is not found (MarcoFalke)
- #10405 0542978 tests: Correct testcase in script\_tests.json for large number OP\_EQUAL (laanwj)
- #10429 6b99daf tests: fix spurious addrman test failure (theuni)
- #10433 8e57256 [tests] improve tmpdir structure (jnewbery)
- #10415 217b416 [tests] Speed up fuzzing by ~200x when using afl-fuzz (practicalswift)
- #10445 b4b057a Add test for empty chain and reorg consistency for gettxoutsetinfo (gmaxwell)
- #10423 1aefc94 [tests] skipped tests should clean up after themselves (jnewbery)
- #10359 329fc1d [tests] functional tests should call BitcoinTestFramework start/stop node methods (jnewbery)
- #10514 e103b3f Bugfix: missing == 0 after randrange (sipa)
- #10515 c871f32 [test] Add test for getchaintxstats (jimmysong)
- #10509 bea5b00 Remove xvfb configuration from travis (ryanofsky)
- #10535 30853e1 [qa] fundrawtx: Fix shutdown race (MarcoFalke)
- #9909 300f8e7 tests: Add FindEarliestAtLeast test for edge cases (ryanofsky)
- #10331 75e898c Share config between util and functional tests (jnewbery)
- #10321 e801084 Use FastRandomContext for all tests (sipa)
- #10524 6c2d81f [tests] Remove printf(...) (practicalswift)
- #10547 71ab6e5 [tests] Use FastRandomContext instead of boost::random:: {mt19937,uniform\_int\_distribution} (practicalswift)
- #10551 6702617 [Tests] Wallet encryption functional tests (achow101)
- #10555 643fa0b [tests] various improvements to zmq\_test.py (jnewbery)
- #10533 d083bd9 [tests] Use cookie auth instead of rpcuser and rpcpassword (achow101)
- #10632 c68a9a6 qa: Add stopatheight test (MarcoFalke)
- #10636 4bc853b [qa] util: Check return code after closing bitcoind proc (MarcoFalke)

- #10662 e0a7801 Initialize randomness in benchmarks (achow101)
- #10612 7c87a9c The young person's guide to the test\_framework (jnewbery)
- #10659 acb1153 [qa] blockchain: Pass on closed connection during generate call (MarcoFalke)
- #10690 416af3e [qa] Bugfix: allow overriding extra\_args in ComparisonTestFramework (sdaftuar)
- #10556 65cc7aa Move stop/start functions from utils.py into BitcoinTestFramework (jnewbery)
- #10704 dd07f47 [tests] nits in dbcrash.py (jnewbery)
- #10743 be82498 [test] don't run dbcrash.py on Travis (jnewbery)
- #10761 d3b5870 [tests] fix replace\_by\_fee.py (jnewbery)
- #10759 1d4805c Fix multi\_rpc test for hosts that dont default to utf8 (TheBlueMatt)
- #10190 e4f226a [tests] mining functional tests (including regression test for submitblock) (jnewbery)
- #10739 1fc783f test: Move variable `state` down where it is used (paveljanik)
- #9980 fee0d80 Fix mem access violation merkleblock (Christewart)
- #10893 0c173a1 [QA] Avoid running multiwallet.py twice (jonasschnelli)
- #10927 9d5e8f9 test: Make sure wallet.backup is created in temp path (laanwj)
- #10899 f29d5db [test] Qt: Use `_putenv_s` instead of `setenv` on Windows builds (brianmcmichael)
- #10912 5c8eb79 [tests] Fix incorrect memory\_cleanse(...) call in crypto\_tests.cpp (practicalswift)
- #11001 fa8a063 [tests] Test disconnecting unsupported service bits logic (jnewbery)
- #10695 929fd72 [qa] Rewrite BIP65/BIP66 functional tests (sdaftuar)
- #10963 ecd2135 [bench] Restore format state of cout after printing with `std::fixed/setprecision` (practicalswift)
- #11025 e5d26e4 qa: Fix inv race in example\_test (MarcoFalke)
- #10765 2c811e0 Tests: address placement should be deterministic by default (ReneNyffenegger)
- #11000 ac016e1 test: Add resendwallettransactions functional tests (promag)
- #11032 aeb3175 [qa] Fix block message processing error in sendheaders.py (sdaftuar)
- #10105 0b9fb68 [tests] fixup - make all Travis test runs quiet, non just cron job runs (jnewbery)
- #10222 6ce7337 [tests] test\_runner - check unicode (jnewbery)
- #10327 35da2ae [tests] remove import-abort-rescan.py (jnewbery)
- #11023 bf74d37 [tests] Add option to attach a python debugger if functional test fails (jnewbery)
- #10565 8c2098a [coverage] Remove subtrees and benchmarks from coverage report (achow101)

## Miscellaneous

- #9871 be8ba2c Add a tree sha512 hash to merge commits (sipa)
- #9821 d19d45a util: Specific GetOSRandom for Linux/FreeBSD/OpenBSD (laanwj)
- #9903 ba80a68 Docs: add details to -rpcclienttimeout doc (ian-kelling)
- #9910 53c300f Docs: correct and elaborate -rpcbind doc (ian-kelling)
- #9905 01b7cda [contrib] gh-merge: Move second sha512 check to the end (MarcoFalke)
- #9880 4df8213 Verify Tree-SHA512s in merge commits, enforce sigs are not SHA1 (TheBlueMatt)
- #9932 00c13ea Fix verify-commits on travis and always check top commit's tree (TheBlueMatt)
- #9952 6996e06 Add historical release notes for 0.14.0 (laanwj)
- #9940 fa99663 Fix verify-commits on OSX, update for new bad Tree-SHA512, point travis to different keyserver (TheBlueMatt)
- #9963 8040ae6 util: Properly handle errors during log message formatting (laanwj)
- #9984 cce056d devtools: Make github-merge compute SHA512 from git, instead of worktree (laanwj)
- #9995 8bcf934 [doc] clarify blockchain size and pruning (askmike)
- #9734 0c17afc Add updating of chainTxData to release process (sipa)
- #10063 530fcdbd add missing spaces so that markdown recognizes headline (flack)
- #10085 db1ae54 Docs: remove 'noconnect' option (jllopp)



- #10090 8e4f7e7 Update bitcoin.conf with example for pruning (coinables)
- #9424 1a5aaab Change LogAcceptCategory to use uint32\_t rather than sets of strings (gmaxwell)
- #10036 fbff36ca Fix init README format to render correctly on github (jlopp)
- #10058 a2cd0b0 No need to use OpenSSL malloc/free (tjps)
- #10123 471ed00 Allow debug logs to be excluded from specified component (jnewbery)
- #10104 fadf078 linearize script: Option to use RPC cookie (achow101)
- #10162 a3a2160 [trivial] Log calls to getblocktemplate (jnewbery)
- #10155 928695b build: Deduplicate version numbers (laanwj)
- #10211 a86255b [doc] Contributor fixes & new "finding reviewers" section (kallewoof)
- #10250 1428f30 Fix some empty vector references (sipa)
- #10270 95f5e44 Remove Clang workaround for Boost 1.46 (fanquake)
- #10263 cb007e4 Trivial: fix fee estimate write error log message (CryptAxe)
- #9670 bd9ec0e contrib: github-merge improvements (laanwj)
- #10260 1d75597 [doc] Minor corrections to osx dependencies (fanquake)
- #10189 750c5a5 devtools/net: add a verifier for scriptable changes. Use it to make CNode::id private (theuni)
- #10322 bc64b5a Use hardware timestamps in RNG seeding (sipa)
- #10381 7f2b9e0 Shadowing warnings are not enabled by default, update doc accordingly (paveljanik)
- #10380 b6ee855 [doc] Removing comments about dirty entries on txmempool (madeo)
- #10383 d0c37ee [logging] log system time and mock time (jnewbery)
- #10404 b45a52a doc: Add logging to FinalizeNode() (sdaftuar)
- #10388 526e839 Output line to debug.log when IsInitialBlockDownload latches to false (morcos)
- #10372 15254e9 Add perf counter data to GetStrongRandBytes state in scheduler (TheBlueMatt)
- #10461 55b72f3 Update style guide (sipa)
- #10486 10e8c0a devtools: Retry after signing fails in github-merge (laanwj)
- #10447 f259263 Make bitcoind invalid argument error message specific (laanwj)
- #10495 6a38b79 contrib: Update location of seeds.txt (laanwj)
- #10469 b6b150b Fixing typo in rpcdump.cpp help message (keystrike)
- #10451 27b9931 contrib/init/bitcoind.openrcconf: Don't disable wallet by default (luke-jr)
- #10323 00d3692 Update to latest libsecp256k1 master (sipa)
- #10422 cec9e1e Fix timestamp in fee estimate debug message (morcos)
- #10566 5d034ee [docs] Use the "domain name setup" image (previously unused) in the gitian docs (practicalswift)
- #10534 a514ac3 Clarify prevector::erase and avoid swap-to-clear (sipa)
- #10575 22ec768 Header include guideline (sipa)
- #10480 fbff5d3b Improve commit-check-script.sh (sipa)
- #10502 1ad3d4e scripted-diff: Remove BOOST\_FOREACH, Q\_FOREACH and PAIRTYPE (jtimon)
- #10377 b63be2c Use rand as entropy source on supported platforms (sipa)
- #9895 228c319 Turn TryCreateDirectory() into TryCreateDirectories() (benma)
- #10602 d76e84a Make clang-format use C++11 features (e.g. A<A> instead of A<A >) (practicalswift)
- #10623 c38f540 doc: Add 0.14.2 release notes (MarcoFalke)
- #10276 b750b33 contrib/verifybinaries: allow filtering by platform (knocte)
- #10248 01c4b14 Rewrite addrdb with less duplication using CHashVerifier (sipa)
- #10577 232508f Add an explanation of quickly hashing onto a non-power of two range (gmaxwell)
- #10608 eee398f Add a comment explaining the use of MAX\_BLOCK\_BASE\_SIZE (gmaxwell)
- #10728 7397af9 fix typo in help text for removeprunedfunds (AkioNak)

- #10193 6dbcc74 scripted-diff: Remove #include <boost/foreach.hpp> (jtimon)
- #10676 379aed0 document script-based return fields for validateaddress (instagibbs)
- #10651 cef4b5c Verify binaries from bitcoincore.org and bitcoin.org (TheBlueMatt)
- #10786 ca4c545 Add PR description to merge commit in github-merge.py (sipa)
- #10812 c5904e8 [utils] Allow bitcoin-cli's -rpccconnect option to be used with square brackets (jnewbery)
- #10842 3895e25 Fix incorrect Doxygen tag (@ince → @since). Doxygen parameter name matching (practicalswift)
- #10681 df0793f add gdb attach process to test README (instagibbs)
- #10789 1124328 Punctuation/grammar fixes in rpcwallet.cpp (stevendlander)
- #10655 78f307b Properly document target\_confirmations in listsinceblock (RHavar)
- #10917 5c003cb developer-notes: add reference to snake\_case and PascalCase (benma)
- #11003 4b5a7ce Docs: Capitalize bullet points in CONTRIBUTING guide (eklitzke)
- #10968 98aa3f6 Add instructions for parallel gitian builds (coblee)
- #11076 1c4b9b3 0.15 release-notes nits: fix redundancy, remove accidental parenthesis & fix range style (practicalswift)
- #11090 8f0121c Update contributor names in release-notes.md (Derek701)
- #11056 cbdd338 disable jni in builds (instagibbs)
- #11080 2b59cfb doc: Update build-openbsd for 6.1 (laanwj)
- #11119 0a6af47 [doc] build-windows: Mention that only trusty works (MarcoFalke)
- #11108 e8ad101 Changing -txindex requires -reindex, not -reindex-chainstate (TheBlueMatt)
- #9792 342b9bc FastRandomContext improvements and switch to ChaCha20 (sipa)
- #9505 67ed40e Prevector Quick Destruct (JeremyRubin)
- #10820 ef37f20 Use cpuid intrinsics instead of asm code (sipa)
- #9999 a328904 [LevelDB] Plug leveldb logs to bitcoin logs (NicolasDorier)
- #9693 c5e9e42 Prevent integer overflow in ReadVarInt (gmaxwell)
- #10129 351d0ad scheduler: fix sub-second precision with boost < 1.50 (theuni)
- #10153 fade788 logging: Fix off-by-one for shrinkdebugfile default (MarcoFalke)
- #10305 c45da32 Fix potential NPD introduced in b297426c (TheBlueMatt)
- #10338 daf3e7d Maintain state across GetStrongRandBytes calls (sipa)
- #10544 a4fe077 Update to LevelDB 1.20 (sipa)
- #10614 cafe24f random: fix crash on some 64bit platforms (theuni)
- #10714 2a09a38 Avoid printing incorrect block indexing time due to uninitialized variable (practicalswift)
- #10837 8bc6d1f Fix resource leak on error in GetDevURandom (corebob)
- #10832 89bb036 init: Factor out AppInitLockDataDirectory and fix startup core dump issue (laanwj)
- #10914 b995a37 Add missing lock in CScheduler::AreThreadsServicingQueue() (TheBlueMatt)
- #10958 659c096 Update to latest Bitcoin patches for LevelDB (sipa)
- #10919 c1c671f Fix more init bugs (TheBlueMatt)

## Credits

Thanks to everyone who directly contributed to this release:

- ロハン ダル
- Ahmad Kazi
- aideca
- Akio Nakamura
- Alex Morcos

- Allan Doensen
- Andres G. Aragoneses
- Andrew Chow
- Angel Leon
- Awemany
- Bob McElrath
- Brian McMichael
- BtcDrak
- Charlie Lee
- Chris Gavin
- Chris Stewart
- Cory Fields
- CryptAxe
- Dag Robole
- Daniel Aleksandersen
- Daniel Cousens
- darksh1ne
- Dimitris Tsapakidis
- Eric Shaw
- Evan Klitzke
- fanquake
- Felix Weis
- flack
- Guido Vranken
- Greg Griffith
- Gregory Maxwell
- Gregory Sanders
- Ian Kelling
- Jack Grigg
- James Evans
- James Hilliard
- Jameson Lopp
- Jeremy Rubin
- Jimmy Song
- João Barbosa
- Johnathan Corgan
- John Newbery
- Jonas Schnelli
- Jorge Timón
- Karl-Johan Alm
- kewde
- KibbledJiveElkZoo
- Kirit Thadaka
- kobake
- Kyle Honeycutt
- Lawrence Nahum
- Luke Dashjr
- Marco Falke
- Marcos Mayorga
- Marijn Stollenga

- Mario Dian
- Mark Friedenbach
- Marko Bencun
- Masahiko Hyuga
- Matt Corallo
- Matthew Zipkin
- Matthias Grundmann
- Michael Goldstein
- Michael Rotarius
- Mikerah
- Mike van Rossum
- Mitchell Cash
- Nicolas Dorier
- Patrick Strateman
- Pavel Janík
- Pavlos Antoniou
- Pavol Rusnak
- Pedro Branco
- Peter Todd
- Pieter Wuille
- practicalswift
- René Nyffenegger
- Ricardo Velhote
- romanornr
- Russell Yanofsky
- Rusty Russell
- Ryan Havar
- shaolinfry
- Shigeya Suzuki
- Simone Madeo
- Spencer Lievens
- Steven D. Lander
- Suhas Daftuar
- Takashi Mitsuta
- Thomas Snider
- Timothy Redaelli
- tintinweb
- tnaka
- Warren Togami
- Wladimir J. van der Laan

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