Bitcoin Core version 0.12.0 is now available from:

https://bitcoin.org/bin/bitcoin-core-0.12.0/

This is a new major version release, bringing new features and other improvements.

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

Downgrade to a version < 0.10.0

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.

Downgrade to a version < 0.12.0

Because release 0.12.0 and later will obfuscate the chainstate on every fresh sync or reindex, the chainstate is not backwards-compatible with pre-0.12 versions of Bitcoin Core or other software.

If you want to downgrade after you have done a reindex with 0.12.0 or later, you will need to reindex when you first start Bitcoin Core version 0.11 or earlier.

Notable changes

Signature validation using libsecp256k1

ECDSA signatures inside Bitcoin transactions now use validation using libsecp256k1 instead of OpenSSL.

Depending on the platform, this means a significant speedup for raw signature validation speed. The advantage is largest on x86_64, where validation is over five times faster. In practice, this translates to a raw reindexing and new block validation times that are less than half of what it was before.

Libsecp256k1 has undergone very extensive testing and validation.

A side effect of this change is that libconsensus no longer depends on OpenSSL.

Reduce upload traffic

A major part of the outbound traffic is caused by serving historic blocks to other nodes in initial block download state.

It is now possible to reduce the total upload traffic via the `-maxuploadtarget` parameter. This is *not* a hard limit but a threshold to minimize the outbound traffic. When the limit is about to be reached, the uploaded data is cut by not serving historic blocks (blocks older than one week). Moreover, any SPV peer is disconnected when they request a filtered block.

This option can be specified in MiB per day and is turned off by default (`-maxuploadtarget=0`). The recommended minimum is 144 * MAX_BLOCK_SIZE (currently 144MB) per day.

Whitelisted peers will never be disconnected, although their traffic counts for calculating the target.

A more detailed documentation about keeping traffic low can be found in /doc/reduce-traffic.md.

Direct headers announcement (BIP 130)

Between compatible peers, [BIP 130] (https://github.com/bitcoin/bips/blob/master/bip-0130.mediawiki) direct headers announcement is used. This means that blocks are advertised by announcing their headers directly, instead of just announcing the hash. In a reorganization, all new headers are sent, instead of just the new tip. This can often prevent an extra roundtrip before the actual block is downloaded.

Memory pool limiting

Previous versions of Bitcoin Core had their mempool limited by checking a transaction's fees against the node's minimum relay fee. There was no upper bound on the size of the mempool and attackers could send a large number of transactions paying just slighly more than the default minimum relay fee to crash nodes with relatively low RAM. A temporary workaround for previous versions of Bitcoin Core was to raise the default minimum relay fee.

Bitcoin Core 0.12 will have a strict maximum size on the mempool. The default value is 300 MB and can be configured with the `-maxmempool` parameter. Whenever a transaction would cause the mempool to exceed its maximum size, the transaction that (along with in-mempool descendants) has the lowest total feerate (as a package) will be evicted and the node's effective minimum relay feerate will be increased to match this feerate plus the initial minimum relay feerate. The initial minimum relay feerate is set to 1000 satoshis per kB.

Bitcoin Core 0.12 also introduces new default policy limits on the length and size of unconfirmed transaction chains that are allowed in the mempool (generally limiting the length of unconfirmed chains to 25 transactions, with a total size of 101 KB). These limits can be overridden using command line arguments; see the extended help (`--help -help-debug`) for more information.

Opt-in Replace-by-fee transactions

It is now possible to replace transactions in the transaction memory pool of Bitcoin Core 0.12 nodes. Bitcoin Core will only allow replacement of transactions which have any of their inputs' `nSequence` number set to less than `0xffffffff - 1`. Moreover, a replacement transaction may only be accepted when it pays sufficient fee, as described in [BIP 125] (https://github.com/bitcoin/bips/blob/master/bip-0125.mediawiki).

Transaction replacement can be disabled with a new command line option, `-mempoolreplacement=0`. Transactions signaling replacement under BIP125 will still be allowed into the mempool in this configuration, but replacements will be rejected. This option is intended for miners who want to continue the transaction selection behavior of previous releases.

The `-mempoolreplacement` option is *not recommended* for wallet users seeking to avoid receipt of unconfirmed opt-in transactions, because this option does not prevent transactions which are replaceable under BIP 125 from being accepted (only subsequent replacements, which other nodes on the network that implement BIP 125 are likely to relay and mine). Wallet users wishing to detect whether a transaction is subject to replacement under BIP 125 should instead use the updated RPC calls `gettransaction` and `listtransactions`, which now have an additional field in the output indicating if a transaction is replaceable under BIP125 ("bip125-replaceable").

Note that the wallet in Bitcoin Core 0.12 does not yet have support for creating transactions that would be replaceable under BIP 125.

RPC: Random-cookie RPC authentication

When no `-rpcpassword` is specified, the daemon now uses a special 'cookie' file for authentication. This file is generated with random content when the daemon starts, and deleted when it exits. Its contents are used as authentication token. Read access to this file controls who can access through RPC. By default it is stored in the data directory but its location can be overridden with the option `-rpccookiefile`.

This is similar to Tor's CookieAuthentication: see https://www.torproject.org/docs/tor-manual.html.en

This allows running bitcoind without having to do any manual configuration.

Relay: Any sequence of pushdatas in OP_RETURN outputs now allowed

Previously OP_RETURN outputs with a payload were only relayed and mined if they had a single pushdata. This restriction has been lifted to allow any combination of data pushes and numeric constant opcodes (OP_1 to OP_16) after the OP_RETURN. The limit on OP_RETURN output size is now applied to the entire serialized scriptPubKey, 83 bytes by default. (the previous 80 byte default plus three bytes overhead)

Relay: New and only new blocks relayed when pruning

When running in pruned mode, the client will now relay new blocks. When responding to the `getblocks` message, only hashes of blocks that are on disk and are likely to remain there for some reasonable time window (1 hour) will be returned (previously all relevant hashes were returned).

Relay and Mining: Priority transactions

Bitcoin Core has a heuristic 'priority' based on coin value and age. This calculation is used for relaying of transactions which do not pay the minimum relay fee, and can be used as an alternative way of sorting transactions for mined blocks. Bitcoin Core will relay transactions with insufficient fees depending on the setting of `-limitfreerelay=<r>` (default: `r=15` kB per minute) and `-blockprioritysize=<s>`.

In Bitcoin Core 0.12, when mempool limit has been reached a higher minimum relay fee takes effect to limit memory usage. Transactions which do not meet this higher effective minimum relay fee will not be relayed or mined even if they rank highly according to the priority heuristic.

The mining of transactions based on their priority is also now disabled by default. To re-enable it, simply set `-blockprioritysize=<n>` where is the size in bytes of your blocks to reserve for these transactions. The old default was 50k, so to retain approximately the same policy, you would set `-blockprioritysize=50000`.

Additionally, as a result of computational simplifications, the priority value used for transactions received with unconfirmed inputs is lower than in prior versions due to avoiding recomputing the amounts as input transactions confirm.

External miner policy set via the `prioritisetransaction` RPC to rank transactions already in the mempool continues to work as it has previously. Note, however, that if mining priority transactions is left disabled, the priority delta will be ignored and only the fee metric will be effective.

This internal automatic prioritization handling is being considered for removal entirely in Bitcoin Core 0.13, and it is at this time undecided whether the more accurate priority calculation for chained unconfirmed transactions will be restored. Community direction on this topic is particularly requested to help set project priorities.

Automatically use Tor hidden services

Starting with Tor version 0.2.7.1 it is possible, through Tor's control socket API, to create and destroy 'ephemeral' hidden services programmatically. Bitcoin Core has been updated to make use of this.

This means that if Tor is running (and proper authorization is available), Bitcoin Core automatically creates a hidden service to listen on, without manual configuration. Bitcoin Core will also use Tor automatically to connect to other .onion nodes if the control socket can be successfully opened. This will positively affect the number of available .onion nodes and their usage.

This new feature is enabled by default if Bitcoin Core is listening, and a connection to Tor can be made. It can be configured with the `-listenonion`, `-torcontrol` and `-torpassword` settings. To show verbose debugging information, pass `-debug=tor`.

Notifications through ZMQ

Bitcoind can now (optionally) asynchronously notify clients through a ZMQ-based PUB socket of the arrival of new transactions and blocks. This feature requires installation of the ZMQ C API library 4.x and configuring its use through the command line or configuration file. Please see docs/zmq.md for details of operation.

Wallet: Transaction fees

Various improvements have been made to how the wallet calculates transaction fees.

Users can decide to pay a predefined fee rate by setting `-paytxfee=<n>` (or `settxfee <n>` rpc during runtime). A value of `n=0` signals Bitcoin Core to use floating fees. By default, Bitcoin Core will use floating fees.

Based on past transaction data, floating fees approximate the fees required to get into the `m`th block from now. This is configurable with `-txconfirmtarget=<m>` (default: `2`).

Sometimes, it is not possible to give good estimates, or an estimate at all. Therefore, a fallback value can be set with `-fallbackfee=<f>` (default: `0.0002` BTC/kB).

At all times, Bitcoin Core will cap fees at `-maxtxfee=<x>` (default: 0.10) BTC. Furthermore, Bitcoin Core will never create transactions paying less than the current minimum relay fee. Finally, a user can set the minimum fee rate for all transactions with `-mintxfee=<i>`, which defaults to 1000 satoshis per kB.

Wallet: Negative confirmations and conflict detection

The wallet will now report a negative number for confirmations that indicates how deep in the block chain the conflict is found. For example, if a transaction A has 5 confirmations and spends the same input as a wallet transaction B, B will be reported as having -5 confirmations. If another wallet transaction C spends an output from B, it will also be reported as having -5 confirmations. To detect conflicts with historical transactions in the chain a one-time `-rescan` may be needed.

Unlike earlier versions, unconfirmed but non-conflicting transactions will never get a negative confirmation count. They are not treated as spendable unless they're coming from ourself (change) and accepted into our local mempool, however. The new "trusted" field in the `listtransactions` RPC output indicates whether outputs of an unconfirmed transaction are considered spendable.

Wallet: Merkle branches removed

Previously, every wallet transaction stored a Merkle branch to prove its presence in blocks. This wasn't being used for more than an expensive sanity check. Since 0.12, these are no longer stored. When loading a 0.12 wallet into an older version, it will automatically rescan to avoid failed checks.

Wallet: Pruning

With 0.12 it is possible to use wallet functionality in pruned mode. This can reduce the disk usage from currently around 60 GB to around 2 GB.

However, rescans as well as the RPCs `importwallet`, `importaddress`, `importprivkey` are disabled.

To enable block pruning set `prune=<N>` on the command line or in `bitcoin.conf`, where `N` is the number of MiB to allot for raw block & undo data.

A value of 0 disables pruning. The minimal value above 0 is 550. Your wallet is as secure with high values as it is with low ones. Higher values merely ensure that your node will not shut down upon blockchain reorganizations of more than 2 days - which are unlikely to happen in practice. In future releases, a higher value may also help the network as a whole: stored blocks could be served to other nodes.

For further information about pruning, you may also consult the release notes of v0.11.0.

`NODE_BLOOM` service bit

Support for the `NODE_BLOOM` service bit, as described in BIP 111, has been added to the P2P protocol code.

BIP 111 defines a service bit to allow peers to advertise that they support bloom filters (such as used by SPV clients) explicitly. It also bumps the protocol version to allow peers to identify old nodes which allow bloom filtering of the connection despite lacking the new service bit.

In this version, it is only enforced for peers that send protocol versions `>=70011`. For the next major version it is planned that this restriction will be removed. It is recommended to update SPV clients to check for the `NODE_BLOOM` service bit for nodes that report versions newer than 70011.

Option parsing behavior

Command line options are now parsed strictly in the order in which they are specified. It used to be the case that `-X -noX` ends up, unintuitively, with X set, as `-X` had precedence over `-noX`. This is no longer the case. Like for other software, the last specified value for an option will hold.

RPC: Low-level API changes

- Monetary amounts can be provided as strings. This means that for example the argument to sendtoaddress can be "0.0001" instead of 0.0001. This can be an advantage if a JSON library insists on using a lossy floating point type for numbers, which would be dangerous for monetary amounts.
- The `asm` property of each scriptSig now contains the decoded signature hash type for each signature that provides a valid defined hash type.
- OP_NOP2 has been renamed to OP_CHECKLOCKTIMEVERIFY by BIP 65

The following items contain assembly representations of scriptSig signatures and are affected by this change:

- RPC `getrawtransaction`
- RPC `decoderawtransaction`
- RPC `decodescript`
- REST `/rest/tx/` (JSON format)
- REST `/rest/block/` (JSON format when including extended tx details)
- `bitcoin-tx -json`

For example, the `scriptSig.asm` property of a transaction input that previously showed an assembly representation of:

```
304502207fa7a6d1e0ee81132a269ad84e68d695483745cde8b541e3bf630749894e342a022100c1f7ab20e13e22
```

now shows as:

```
304502207fa7a6d1e0ee81132a269ad84e68d695483745cde8b541e3bf630749894e342a022100c1f7ab20e13e22
```

Note that the output of the RPC `decodescript` did not change because it is configured specifically to process scriptPubKey and not scriptSig scripts.

RPC: SSL support dropped

SSL support for RPC, previously enabled by the option `rpcssl` has been dropped from both the client and the server. This was done in preparation for removing the dependency on OpenSSL for the daemon completely.

Trying to use `rpcssl` will result in an error:

```
Error: SSL mode for RPC (-rpcssl) is no longer supported.
```

If you are one of the few people that relies on this feature, a flexible migration path is to use `stunnel`. This is an utility that can tunnel arbitrary TCP connections inside SSL. On e.g. Ubuntu it can be installed with:

```
sudo apt-get install stunnel4
```

Then, to tunnel a SSL connection on 28332 to a RPC server bound on localhost on port 18332 do:

```
stunnel -d 28332 -r 127.0.0.1:18332 -p stunnel.pem -P ''
```

It can also be set up system-wide in inetd style.

Another way to re-attain SSL would be to setup a httpd reverse proxy. This solution would allow the use of different authentication, loadbalancing, on-the-fly compression and caching. A sample config for apache2 could look like:

```
Listen 443

NameVirtualHost *:443
<VirtualHost *:443>

SSLEngine On
SSLCertificateFile /etc/apache2/ssl/server.crt
```

```
SSLCertificateKeyFile /etc/apache2/ssl/server.key

<Location /bitcoinrpc>
    ProxyPass http://127.0.0.1:8332/
    ProxyPassReverse http://127.0.0.1:8332/
    # optional enable digest auth
    # AuthType Digest
    # ...

    # optional bypass bitcoind rpc basic auth
    # RequestHeader set Authorization "Basic <hash>"
    # get the <hash> from the shell with: base64 <<< bitcoinrpc:<password>
</Location>

# Or, balance the load:
# ProxyPass / balancer://balancer_cluster_name

</VirtualHost>
```

Mining Code Changes

The mining code in 0.12 has been optimized to be significantly faster and use less memory. As part of these changes, consensus critical calculations are cached on a transaction's acceptance into the mempool and the mining code now relies on the consistency of the mempool to assemble blocks. However all blocks are still tested for validity after assembly.

Other P2P Changes

The list of banned peers is now stored on disk rather than in memory. Restarting bitcoind will no longer clear out the list of banned peers; instead a new RPC call (`clearbanned`) can be used to manually clear the list. The new `setban` RPC call can also be used to manually ban or unban a peer.

0.12.0 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

- #6121 `466f0ea` Convert entire source tree from json_spirit to UniValue (Jonas Schnelli)

- #6234 `d38cd47` fix rpcmining/getblocktemplate univalue transition logic error (Jonas Schnelli)
- #6239 `643114f` Don't go through double in AmountFromValue and ValueFromAmount (Wladimir J. van der Laan)
- #6266 `ebab5d3` Fix univalue handling of ŭ0000 characters. (Daniel Kraft)
- #6276 `f3d4dbb` Fix getbalance * 0 (Tom Harding)
- #6257 `5ebe7db` Add `paytxfee` and `errors` JSON fields where appropriate (Stephen)
- #6271 `754aae5` New RPC command disconnectnode (Alex van der Peet)
- #6158 `0abfa8a` Add setban/listbanned RPC commands (Jonas Schnelli)
- #6307 `7ecdcd9` rpcban fixes (Jonas Schnelli)
- #6290 `5753988` rpc: make `gettxoutsettinfo` run lock-free (Wladimir J. van der Laan)
- #6262 `247b914` Return all available information via RPC call "validateaddress" (dexX7)
- #6339 `c3f0490` UniValue: don't escape solidus, keep espacing of reverse solidus (Jonas Schnelli)
- #6353 `6bcb0a2` Show softfork status in getblockchaininfo (Wladimir J. van der Laan)
- #6247 `726e286` Add getblockheader RPC call (Peter Todd)
- #6362 `d6db115` Fix null id in RPC response during startup (Forrest Voight)
- #5486 `943b322` [REST] JSON support for /rest/headers (Jonas Schnelli)
- #6379 `c52e8b3` rpc: Accept scientific notation for monetary amounts in JSON (Wladimir J. van der Laan)
- #6388 `fd5dfda` rpc: Implement random-cookie based authentication (Wladimir J. van der Laan)
- #6457 `3c923e8` Include pruned state in chaininfo.json (Simon Males)
- #6456 `bfd807f` rpc: Avoid unnecessary parsing roundtrip in number formatting, fix locale issue (Wladimir J. van der Laan)
- #6380 `240b30e` rpc: Accept strings in AmountFromValue (Wladimir J. van der Laan)
- #6346 `6bb2805` Add OP_RETURN support in createrawtransaction RPC call, add tests. (paveljanik)
- #6013 `6feeec1` [REST] Add memory pool API (paveljanik)
- #6576 `da9beb2` Stop parsing JSON after first finished construct. (Daniel Kraft)
- #5677 `9aa9099` libevent-based http server (Wladimir J. van der Laan)
- #6633 `bbc2b39` Report minimum ping time in getpeerinfo (Matt Corallo)
- #6648 `cd381d7` Simplify logic of REST request suffix parsing. (Daniel Kraft)
- #6695 `5e21388` libevent http fixes (Wladimir J. van der Laan)
- #5264 `48efbdb` show scriptSig signature hash types in transaction decodes. fixes #3166 (mruddy)
- #6719 `1a9f19a` Make HTTP server shutdown more graceful (Wladimir J. van der Laan)
- #6859 `0fbfc51` http: Restrict maximum size of http + headers (Wladimir

J. van der Laan)
- #5936 `bf7c195` [RPC] Add optional locktime to createrawtransaction (Tom Harding)
- #6877 `26f5b34` rpc: Add maxmempool and effective min fee to getmempoolinfo (Wladimir J. van der Laan)
- #6970 `92701b3` Fix crash in validateaddress with -disablewallet (Wladimir J. van der Laan)
- #5574 `755b4ba` Expose GUI labels in RPC as comments (Luke-Jr)
- #6990 `dbd2c13` http: speed up shutdown (Wladimir J. van der Laan)
- #7013 `36baa9f` Remove LOCK(cs_main) from decodescript (Peter Todd)
- #6999 `972bf9c` add (max)uploadtarget infos to getnettotals RPC help (Jonas Schnelli)
- #7011 `31de241` Add mediantime to getblockchaininfo (Peter Todd)
- #7065 `f91e29f` http: add Boost 1.49 compatibility (Wladimir J. van der Laan)
- #7087 `be281d8` [Net]Add -enforcenodebloom option (Patrick Strateman)
- #7044 `438ee59` RPC: Added additional config option for multiple RPC users. (Gregory Sanders)
- #7072 `c143c49` [RPC] Add transaction size to JSON output (Nikita Zhavoronkov)
- #7022 `9afbd96` Change default block priority size to 0 (Alex Morcos)
- #7141 `c0c08c7` rpc: Don't translate warning messages (Wladimir J. van der Laan)
- #7312 `fd4bd50` Add RPC call abandontransaction (Alex Morcos)
- #7222 `e25b158` RPC: indicate which transactions are replaceable (Suhas Daftuar)
- #7472 `b2f2b85` rpc: Add WWW-Authenticate header to 401 response (Wladimir J. van der Laan)
- #7469 `9cb31e6` net.h fix spelling: misbeha{b,v}ing (Matt)

Configuration and command-line options

- #6164 `8d05ec7` Allow user to use -debug=1 to enable all debugging (lpescher)
- #5288 `4452205` Added `-whiteconnections=<n>` option (Josh Lehan)
- #6284 `10ac38e` Fix argument parsing oddity with -noX (Wladimir J. van der Laan)
- #6489 `c9c017a` Give a better error message if system clock is bad (Casey Rodarmor)
- #6462 `c384800` implement uacomment config parameter which can add comments to user agent as per BIP-0014 (Pavol Rusnak)
- #6647 `a3babc8` Sanitize uacomment (MarcoFalke)
- #6742 `3b2d37c` Changed logging to make -logtimestamps to work also for -printtoconsole (arnuschky)
- #6846 `2cd020d` alias -h for -help (Daniel Cousens)
- #6622 `7939164` Introduce -maxuploadtarget (Jonas Schnelli)

- #6881 `2b62551` Debug: Add option for microsecond precision in debug.log (Suhas Daftuar)
- #6776 `e06c14f` Support -checkmempool=N, which runs checks once every N transactions (Pieter Wuille)
- #6896 `d482c0a` Make -checkmempool=1 not fail through int32 overflow (Pieter Wuille)
- #6993 `b632145` Add -blocksonly option (Patrick Strateman)
- #7323 `a344880` 0.12: Backport -bytespersigop option (Luke-Jr)
- #7386 `da83ecd` Add option `-permitrbf` to set transaction replacement policy (Wladimir J. van der Laan)
- #7290 `b16b5bc` Add missing options help (MarcoFalke)
- #7440 `c76bfff` Rename permitrbf to mempoolreplacement and provide minimal string-list forward compatibility (Luke-Jr)

Block and transaction handling

- #6203 `f00b623` Remove P2SH coinbase flag, no longer interesting (Luke-Jr)
- #6222 `9c93ee5` Explicitly set tx.nVersion for the genesis block and mining tests (Mark Friedenbach)
- #5985 `3a1d3e8` Fix removing of orphan transactions (Alex Morcos)
- #6221 `dd8fe82` Prune: Support noncontiguous block files (Adam Weiss)
- #6124 `41076aa` Mempool only CHECKLOCKTIMEVERIFY (BIP65) verification, unparameterized version (Peter Todd)
- #6329 `d0a10c1` acceptnonstdtxn option to skip (most) "non-standard transaction" checks, for testnet/regtest only (Luke-Jr)
- #6410 `7cdefb9` Implement accurate memory accounting for mempool (Pieter Wuille)
- #6444 `24ce77d` Exempt unspendable transaction outputs from dust checks (dexX7)
- #5913 `a0625b8` Add absurdly high fee message to validation state (Shaul Kfir)
- #6177 `2f746c6` Prevent block.nTime from decreasing (Mark Friedenbach)
- #6377 `e545371` Handle no chain tip available in InvalidChainFound() (Ross Nicoll)
- #6551 `39ddaeb` Handle leveldb::DestroyDB() errors on wipe failure (Adam Weiss)
- #6654 `b0ce450` Mempool package tracking (Suhas Daftuar)
- #6715 `82d2aef` Fix mempool packages (Suhas Daftuar)
- #6680 `4f44530` use CBlockIndex instead of uint256 for UpdatedBlockTip signal (Jonas Schnelli)
- #6650 `4fac576` Obfuscate chainstate (James O'Beirne)
- #6777 `9caaf6e` Unobfuscate chainstate data in CCoinsViewDB::GetStats (James O'Beirne)
- #6722 `3b20e23` Limit mempool by throwing away the cheapest txn and setting min relay fee to it (Matt Corallo)
- #6889 `38369dd` fix locking issue with new mempool limiting (Jonas

Schnelli)
- #6464 `8f3b3cd` Always clean up manual transaction prioritization (Casey Rodarmor)
- #6865 `d0badb9` Fix chainstate serialized_size computation (Pieter Wuille)
- #6566 `ff057f4` BIP-113: Mempool-only median time-past as endpoint for lock-time calculations (Mark Friedenbach)
- #6934 `3038eb6` Restores mempool only BIP113 enforcement (Gregory Maxwell)
- #6965 `de7d459` Benchmark sanity checks and fork checks in ConnectBlock (Matt Corallo)
- #6918 `eb6172a` Make sigcache faster, more efficient, larger (Pieter Wuille)
- #6771 `38ed190` Policy: Lower default limits for tx chains (Alex Morcos)
- #6932 `73fa5e6` ModifyNewCoins saves database lookups (Alex Morcos)
- #5967 `05d5918` Alter assumptions in CCoinsViewCache::BatchWrite (Alex Morcos)
- #6871 `0e93586` nSequence-based Full-RBF opt-in (Peter Todd)
- #7008 `eb77416` Lower bound priority (Alex Morcos)
- #6915 `2ef5ffa` [Mempool] Improve removal of invalid transactions after reorgs (Suhas Daftuar)
- #6898 `4077ad2` Rewrite CreateNewBlock (Alex Morcos)
- #6872 `bdda4d5` Remove UTXO cache entries when the tx they were added for is removed/does not enter mempool (Matt Corallo)
- #7062 `12c469b` [Mempool] Fix mempool limiting and replace-by-fee for PrioritiseTransaction (Suhas Daftuar)
- #7276 `76de36f` Report non-mandatory script failures correctly (Pieter Wuille)
- #7217 `e08b7cb` Mark blocks with too many sigops as failed (Suhas Daftuar)
- #7387 `f4b2ce8` Get rid of inaccurate ScriptSigArgsExpected (Pieter Wuille)

P2P protocol and network code

- #6172 `88a7ead` Ignore getheaders requests when not synced (Suhas Daftuar)
- #5875 `9d60602` Be stricter in processing unrequested blocks (Suhas Daftuar)
- #6256 `8ccc07c` Use best header chain timestamps to detect partitioning (Gavin Andresen)
- #6283 `a903ad7` make CAddrMan::size() return the correct type of size_t (Diapolo)
- #6272 `40400d5` Improve proxy initialization (continues #4871) (Wladimir J. van der Laan, Diapolo)
- #6310 `66e5465` banlist.dat: store banlist on disk (Jonas Schnelli)
- #6412 `1a2de32` Test whether created sockets are select()able (Pieter Wuille)
- #6498 `219b916` Keep track of recently rejected transactions with a rolling

bloom filter (cont'd) (Peter Todd)
- #6556 `70ec975` Fix masking of irrelevant bits in address groups. (Alex Morcos)
- #6530 `ea19c2b` Improve addrman Select() performance when buckets are nearly empty (Pieter Wuille)
- #6583 `af9305a` add support for miniupnpc api version 14 (Pavel Vasin)
- #6374 `69dc5b5` Connection slot exhaustion DoS mitigation (Patrick Strateman)
- #6636 `536207f` net: correctly initialize nMinPingUsecTime (Wladimir J. van der Laan)
- #6579 `0c27795` Add NODE_BLOOM service bit and bump protocol version (Matt Corallo)
- #6148 `999c8be` Relay blocks when pruning (Suhas Daftuar)
- #6588 `cf9bb11` In (strCommand == "tx"), return if AlreadyHave() (Tom Harding)
- #6974 `2f71b07` Always allow getheaders from whitelisted peers (Wladimir J. van der Laan)
- #6639 `bd629d7` net: Automatically create hidden service, listen on Tor (Wladimir J. van der Laan)
- #6984 `9ffc687` don't enforce maxuploadtarget's disconnect for whitelisted peers (Jonas Schnelli)
- #7046 `c322652` Net: Improve blocks only mode. (Patrick Strateman)
- #7090 `d6454f6` Connect to Tor hidden services by default (when listening on Tor) (Peter Todd)
- #7106 `c894fbb` Fix and improve relay from whitelisted peers (Pieter Wuille)
- #7129 `5d5ef3a` Direct headers announcement (rebase of #6494) (Pieter Wuille)
- #7079 `1b5118b` Prevent peer flooding inv request queue (redux) (redux) (Gregory Maxwell)
- #7166 `6ba25d2` Disconnect on mempool requests from peers when over the upload limit. (Gregory Maxwell)
- #7133 `f31955d` Replace setInventoryKnown with a rolling bloom filter (rebase of #7100) (Pieter Wuille)
- #7174 `82aff88` Don't do mempool lookups for "mempool" command without a filter (Matt Corallo)
- #7179 `44fef99` net: Fix sent reject messages for blocks and transactions (Wladimir J. van der Laan)
- #7181 `8fc174a` net: Add and document network messages in protocol.h (Wladimir J. van der Laan)
- #7125 `10b88be` Replace global trickle node with random delays (Pieter Wuille)
- #7415 `cb83beb` net: Hardcoded seeds update January 2016 (Wladimir J. van der Laan)
- #7438 `e2d9a58` Do not absolutely protect local peers; decide group ties based on time (Gregory Maxwell)

- #7439 `86755bc` Add whitelistforcerelay to control forced relaying. [#7099 redux] (Gregory Maxwell)
- #7482 `e16f5b4` Ensure headers count is correct (Suhas Daftuar)

Validation

- #5927 `8d9f0a6` Reduce checkpoints' effect on consensus. (Pieter Wuille)
- #6299 `24f2489` Bugfix: Don't check the genesis block header before accepting it (Jorge Timón)
- #6361 `d7ada03` Use real number of cores for default -par, ignore virtual cores (Wladimir J. van der Laan)
- #6519 `87f37e2` Make logging for validation optional (Wladimir J. van der Laan)
- #6351 `2a1090d` CHECKLOCKTIMEVERIFY (BIP65) IsSuperMajority() soft-fork (Peter Todd)
- #6931 `54e8bfe` Skip BIP 30 verification where not necessary (Alex Morcos)
- #6954 `e54ebbf` Switch to libsecp256k1-based ECDSA validation (Pieter Wuille)
- #6508 `61457c2` Switch to a constant-space Merkle root/branch algorithm. (Pieter Wuille)
- #6914 `327291a` Add pre-allocated vector type and use it for CScript (Pieter Wuille)
- #7500 `889e5b3` Correctly report high-S violations (Pieter Wuille)

Build system

- #6210 `0e4f2a0` build: disable optional use of gmp in internal secp256k1 build (Wladimir J. van der Laan)
- #6214 `87406aa` [OSX] revert renaming of Bitcoin-Qt.app and use CFBundleDisplayName (partial revert of #6116) (Jonas Schnelli)
- #6218 `9d67b10` build/gitian misc updates (Cory Fields)
- #6269 `d4565b6` gitian: Use the new bitcoin-detached-sigs git repo for OSX signatures (Cory Fields)
- #6418 `d4a910c` Add autogen.sh to source tarball. (randy-waterhouse)
- #6373 `1ae3196` depends: non-qt bumps for 0.12 (Cory Fields)
- #6434 `059b352` Preserve user-passed CXXFLAGS with –enable-debug (Gavin Andresen)
- #6501 `fee6554` Misc build fixes (Cory Fields)
- #6600 `ef4945f` Include bitcoin-tx binary on Debian/Ubuntu (Zak Wilcox)
- #6619 `4862708` depends: bump miniupnpc and ccache (Michael Ford)
- #6801 `ae69a75` [depends] Latest config.guess and config.sub (Michael Ford)
- #6938 `193f7b5` build: If both Qt4 and Qt5 are installed, use Qt5 (Wladimir J. van der Laan)
- #7092 `348b281` build: Set osx permissions in the dmg to make Gatekeeper happy (Cory Fields)
- #6980 `eccd671` [Depends] Bump Boost, miniupnpc, ccache & zeromq

(Michael Ford)
- #7424 `aa26ee0` Add security/export checks to gitian and fix current failures (Cory Fields)

Wallet

- #6183 `87550ee` Fix off-by-one error w/ nLockTime in the wallet (Peter Todd)
- #6057 `ac5476e` re-enable wallet in autoprune (Jonas Schnelli)
- #6356 `9e6c33b` Delay initial pruning until after wallet init (Adam Weiss)
- #6088 `91389e5` fundrawtransaction (Matt Corallo)
- #6415 `ddd8d80` Implement watchonly support in fundrawtransaction (Matt Corallo)
- #6567 `0f0f323` Fix crash when mining with empty keypool. (Daniel Kraft)
- #6688 `4939eab` Fix locking in GetTransaction. (Alex Morcos)
- #6645 `4dbd43e` Enable wallet key imports without rescan in pruned mode. (Gregory Maxwell)
- #6550 `5b77244` Do not store Merkle branches in the wallet. (Pieter Wuille)
- #5924 `12a7712` Clean up change computation in CreateTransaction. (Daniel Kraft)
- #6906 `48b5b84` Reject invalid pubkeys when reading ckey items from the wallet. (Gregory Maxwell)
- #7010 `e0a5ef8` Fix fundrawtransaction handling of includeWatching (Peter Todd)
- #6851 `616d61b` Optimisation: Store transaction list order in memory rather than compute it every need (Luke-Jr)
- #6134 `e92377f` Improve usage of fee estimation code (Alex Morcos)
- #7103 `a775182` [wallet, rpc tests] Fix settxfee, paytxfee (MarcoFalke)
- #7105 `30c2d8c` Keep track of explicit wallet conflicts instead of using mempool (Pieter Wuille)
- #7096 `9490bd7` Wallet Improve minimum absolute fee GUI options (Jonas Schnelli)
- #6216 `83f06ca` Take the training wheels off anti-fee-sniping (Peter Todd)
- #4906 `96e8d12` Issue#1643: Coinselection prunes extraneous inputs from ApproximateBestSubset (Murch)
- #7200 `06c6a58` Checks for null data transaction before issuing error to debug.log (Andy Craze)
- #7296 `a36d79b` Add sane fallback for fee estimation (Alex Morcos)
- #7293 `ff9b610` Add regression test for vValue sort order (MarcoFalke)
- #7306 `4707797` Make sure conflicted wallet tx's update balances (Alex Morcos)
- #7381 `621bbd8` [walletdb] Fix syntax error in key parser (MarcoFalke)
- #7491 `00ec73e` wallet: Ignore MarkConflict if block hash is not known (Wladimir J. van der Laan)
- #7502 `1329963` Update the wallet best block marker before pruning (Pieter Wuille)

GUI

- #6217 `c57e12a` disconnect peers from peers tab via context menu (Diapolo)
- #6209 `ab0ec67` extend rpc console peers tab (Diapolo)
- #6484 `1369d69` use CHashWriter also in SignVerifyMessageDialog (Pavel Vasin)
- #6487 `9848d42` Introduce PlatformStyle (Wladimir J. van der Laan)
- #6505 `100c9d3` cleanup icons (MarcoFalke)
- #4587 `0c465f5` allow users to set -onion via GUI (Diapolo)
- #6529 `c0f66ce` show client user agent in debug window (Diapolo)
- #6594 `878ea69` Disallow duplicate windows. (Casey Rodarmor)
- #5665 `6f55cdd` add verifySize() function to PaymentServer (Diapolo)
- #6317 `ca5e2a1` minor optimisations in peertablemodel (Diapolo)
- #6315 `e59d2a8` allow banning and unbanning over UI->peers table (Jonas Schnelli)
- #6653 `e04b2fa` Pop debug window in foreground when opened twice (MarcoFalke)
- #6864 `c702521` Use monospace font (MarcoFalke)
- #6887 `3694b74` Update coin control and smartfee labels (MarcoFalke)
- #7000 `814697c` add shortcurts for debug-/console-window (Jonas Schnelli)
- #6951 `03403d8` Use maxTxFee instead of 10000000 (MarcoFalke)
- #7051 `a190777` ui: Add "Copy raw transaction data" to transaction list context menu (Wladimir J. van der Laan)
- #6979 `776848a` simple mempool info in debug window (Jonas Schnelli)
- #7006 `26af1ac` add startup option to reset Qt settings (Jonas Schnelli)
- #6780 `2a94cd6` Call init's parameter interaction before we create the UI options model (Jonas Schnelli)
- #7112 `96b8025` reduce cs_main locks during tip update, more fluently update UI (Jonas Schnelli)
- #7206 `f43c2f9` Add "NODE_BLOOM" to guiutil so that peers don't get UNKNOWN[4] (Matt Corallo)
- #7282 `5cadf3e` fix coincontrol update issue when deleting a send coins entry (Jonas Schnelli)
- #7319 `1320300` Intro: Display required space (MarcoFalke)
- #7318 `9265e89` quickfix for RPC timer interface problem (Jonas Schnelli)
- #7327 `b16b5bc` Wallet Transaction View: LastMonth calculation fixed (crowning-)
- #7364 `7726c48` [qt] Windows: Make rpcconsole monospace font larger (MarcoFalke)
- #7384 `294f432` [qt] Peertable: Increase SUBVERSION_COLUMN_WIDTH (MarcoFalke)

Tests and QA

- #6305 `9005c91` build: comparison tool swap (Cory Fields)
- #6318 `e307e13` build: comparison tool NPE fix (Cory Fields)

- #6337 `0564c5b` Testing infrastructure: mocktime fixes (Gavin Andresen)
- #6350 `60abba1` add unit tests for the decodescript rpc (mruddy)
- #5881 `3203a08` Fix and improve txn_doublespend.py test (Tom Harding)
- #6390 `6a73d66` tests: Fix bitcoin-tx signing test case (Wladimir J. van der Laan)
- #6368 `7fc25c2` CLTV: Add more tests to improve coverage (Esteban Ordano)
- #6414 `5121c68` Fix intermittent test failure, reduce test time (Tom Harding)
- #6417 `44fa82d` [QA] fix possible reorg issue in (fund)rawtransaction(s).py RPC test (Jonas Schnelli)
- #6398 `3d9362d` rpc: Remove chain-specific RequireRPCPassword (Wladimir J. van der Laan)
- #6428 `bb59e78` tests: Remove old sh-based test framework (Wladimir J. van der Laan)
- #5515 `d946e9a` RFC: Assert on probable deadlocks if the second lock isnt try_lock (Matt Corallo)
- #6287 `d2464df` Clang lock debug (Cory Fields)
- #6465 `410fd74` Don't share objects between TestInstances (Casey Rodarmor)
- #6534 `6c1c7fd` Fix test locking issues and un-revert the probable-deadlines assertions commit (Cory Fields)
- #6509 `bb4faee` Fix race condition on test node shutdown (Casey Rodarmor)
- #6523 `561f8af` Add p2p-fullblocktest.py (Casey Rodarmor)
- #6590 `981fd92` Fix stale socket rebinding and re-enable python tests for Windows (Cory Fields)
- #6730 `cb4d6d0` build: Remove dependency of bitcoin-cli on secp256k1 (Wladimir J. van der Laan)
- #6616 `5ab5dca` Regression Tests: Migrated rpc-tests.sh to all Python rpc-tests.py (Peter Tschipper)
- #6720 `d479311` Creates unittests for addrman, makes addrman more testable. (Ethan Heilman)
- #6853 `c834f56` Added fPowNoRetargeting field to Consensus::Params (Eric Lombrozo)
- #6827 `87e5539` [rpc-tests] Check return code (MarcoFalke)
- #6848 `f2c869a` Add DERSIG transaction test cases (Ross Nicoll)
- #6813 `5242bb3` Support gathering code coverage data for RPC tests with lcov (dexX7)
- #6888 `c8322ff` Clear strMiscWarning before running PartitionAlert (Eric Lombrozo)
- #6894 `2675276` [Tests] Fix BIP65 p2p test (Suhas Daftuar)
- #6863 `725539e` [Test Suite] Fix test for null tx input (Daniel Kraft)
- #6926 `a6d0d62` tests: Initialize networking on windows (Wladimir J. van der Laan)
- #6822 `9fa54a1` [tests] Be more strict checking dust (MarcoFalke)

- #6804 `5fcc14e` [tests] Add basic coverage reporting for RPC tests (James O'Beirne)
- #7045 `72dccfc` Bugfix: Use unique autostart filenames on Linux for testnet/regtest (Luke-Jr)
- #7095 `d8368a0` Replace scriptnum_test's normative ScriptNum implementation (Wladimir J. van der Laan)
- #7063 `6abf6eb` [Tests] Add prioritisetransaction RPC test (Suhas Daftuar)
- #7137 `16f4a6e` Tests: Explicitly set chain limits in replace-by-fee test (Suhas Daftuar)
- #7216 `9572e49` Removed offline testnet DNSSeed 'alexykot.me'. (tnull)
- #7209 `f3ad812` test: don't override BITCOIND and BITCOINCLI if they're set (Wladimir J. van der Laan)
- #7226 `301f16a` Tests: Add more tests to p2p-fullblocktest (Suhas Daftuar)
- #7153 `9ef7c54` [Tests] Add mempool_limit.py test (Jonas Schnelli)
- #7170 `453c567` tests: Disable Tor interaction (Wladimir J. van der Laan)
- #7229 `1ed938b` [qa] wallet: Check if maintenance changes the balance (MarcoFalke)
- #7308 `d513405` [Tests] Eliminate intermittent failures in sendheaders.py (Suhas Daftuar)
- #7468 `947c4ff` [rpc-tests] Change solve() to use rehash (Brad Andrews)

Miscellaneous

- #6213 `e54ff2f` [init] add -blockversion help and extend -upnp help (Diapolo)
- #5975 `1fea667` Consensus: Decouple ContextualCheckBlockHeader from checkpoints (Jorge Timón)
- #6061 `eba2f06` Separate Consensus::CheckTxInputs and GetSpendHeight in CheckInputs (Jorge Timón)
- #5994 `786ed11` detach wallet from miner (Jonas Schnelli)
- #6387 `11576a5` [bitcoin-cli] improve error output (Jonas Schnelli)
- #6401 `6db53b4` Add BITCOIND_SIGTERM_TIMEOUT to OpenRC init scripts (Florian Schmaus)
- #6430 `b01981e` doc: add documentation for shared library libbitcoinconsensus (Braydon Fuller)
- #6372 `dcc495e` Update Linearize tool to support Windows paths; fix variable scope; update README and example configuration (Paul Georgiou)
- #6453 `8fe5cce` Separate core memory usage computation in core_memusage.h (Pieter Wuille)
- #6149 `633fe10` Buffer log messages and explicitly open logs (Adam Weiss)
- #6488 `7cbed7f` Avoid leaking file descriptors in RegisterLoad (Casey Rodarmor)
- #6497 `a2bf40d` Make sure LogPrintf strings are line-terminated (Wladimir J. van der Laan)
- #6504 `b6fee6b` Rationalize currency unit to "BTC" (Ross Nicoll)
- #6507 `9bb4dd8` Removed contrib/bitrpc (Casey Rodarmor)

- #6527 `41d650f` Use unique name for AlertNotify tempfile (Casey Rodarmor)
- #6561 `e08a7d9` limitedmap fixes and tests (Casey Rodarmor)
- #6565 `a6f2aff` Make sure we re-acquire lock if a task throws (Casey Rodarmor)
- #6599 `f4d88c4` Make sure LogPrint strings are line-terminated (Ross Nicoll)
- #6630 `195942d` Replace boost::reverse_lock with our own (Casey Rodarmor)
- #6103 `13b8282` Add ZeroMQ notifications (João Barbosa)
- #6692 `d5d1d2e` devtools: don't push if signing fails in github-merge (Wladimir J. van der Laan)
- #6728 `2b0567b` timedata: Prevent warning overkill (Wladimir J. van der Laan)
- #6713 `f6ce59c` SanitizeString: Allow hypen char (MarcoFalke)
- #5987 `4899a04` Bugfix: Fix testnet-in-a-box use case (Luke-Jr)
- #6733 `b7d78fd` Simple benchmarking framework (Gavin Andresen)
- #6854 `a092970` devtools: Add security-check.py (Wladimir J. van der Laan)
- #6790 `fa1d252` devtools: add clang-format.py (MarcoFalke)
- #7114 `f3d0fdd` util: Don't set strMiscWarning on every exception (Wladimir J. van der Laan)
- #7078 `93e0514` uint256::GetCheapHash bigendian compatibility (arowser)
- #7094 `34e02e0` Assert now > 0 in GetTime GetTimeMillis GetTimeMicros (Patrick Strateman)

Credits

Thanks to everyone who directly contributed to this release:

- accraze
- Adam Weiss
- Alex Morcos
- Alex van der Peet
- AlSzacrel
- Altoidnerd
- Andriy Voskoboinyk
- antonio-fr
- Arne Brutschy
- Ashley Holman
- Bob McElrath
- Braydon Fuller
- BtcDrak
- Casey Rodarmor
- centaur1

- Chris Kleeschulte
- Christian Decker
- Cory Fields
- crowning-
- daniel
- Daniel Cousens
- Daniel Kraft
- David Hill
- dexX7
- Diego Viola
- Elias Rohrer
- Eric Lombrozo
- Erik Mossberg
- Esteban Ordano
- EthanHeilman
- Florian Schmaus
- Forrest Voight
- Gavin Andresen
- Gregory Maxwell
- Gregory Sanders / instagibbs
- Ian T
- Irving Ruan
- Jacob Welsh
- James O'Beirne
- Jeff Garzik
- Johnathan Corgan
- Jonas Schnelli
- Jonathan Cross
- João Barbosa
- Jorge Timón
- Josh Lehan
- J Ross Nicoll
- kazcw
- Kevin Cooper
- lpescher
- Luke Dashjr
- MarcoFalke
- Mark Friedenbach
- Matt
- Matt Bogosian
- Matt Corallo
- Matt Quinn
- Micha
- Michael
- Michael Ford / fanquake
- Midnight Magic

- Mitchell Cash
- mrbandrews
- mruddy
- Nick
- Patrick Strateman
- Paul Georgiou
- Paul Rabahy
- Pavel Janík / paveljanik
- Pavel Vasin
- Pavol Rusnak
- Peter Josling
- Peter Todd
- Philip Kaufmann
- Pieter Wuille
- ptschip
- randy-waterhouse
- rion
- Ross Nicoll
- Ryan Havar
- Shaul Kfir
- Simon Males
- Stephen
- Suhas Daftuar
- tailsjoin
- Thomas Kerin
- Tom Harding
- tulip
- unsystemizer
- Veres Lajos
- Wladimir J. van der Laan
- xor-freenet
- Zak Wilcox
- zathras-crypto

As well as everyone that helped translating on Transifex.