0.21.0 Release Notes

Bitcoin Core version 0.21.0 is now available from:

https://bitcoincore.org/bin/bitcoin-core-0.21.0/

This release includes new features, various bug fixes 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 in some cases), then run the installer (on Windows) or just copy over `/Applications/Bitcoin-Qt` (on Mac) or `bitcoind` / `bitcoin-qt` (on Linux).

Upgrading directly from a version of Bitcoin Core that has reached its EOL is possible, but it might take some time if the data directory needs to be migrated. Old wallet versions of Bitcoin Core are generally supported.

Compatibility

Bitcoin Core is supported and extensively tested on operating systems using the Linux kernel, macOS 10.12+, and Windows 7 and newer. Bitcoin Core should also work on most other Unix-like systems but is not as frequently tested on them. It is not recommended to use Bitcoin Core on unsupported systems.

From Bitcoin Core 0.20.0 onwards, macOS versions earlier than 10.12 are no longer supported. Additionally, Bitcoin Core does not yet change appearance when macOS "dark mode" is activated.

The node's known peers are persisted to disk in a file called `peers.dat`. The format of this file has been changed in a backwards-incompatible way in order to accommodate the storage of Tor v3 and other BIP155 addresses. This means that if the file is modified by 0.21.0 or newer then older versions will not be able to read it. Those old versions, in the event of a downgrade, will log an error message "Incorrect keysize in addrman deserialization" and will continue normal operation as if the file was missing, creating a new empty one. (#19954, #20284)

Notable changes

P2P and network changes

- The mempool now tracks whether transactions submitted via the wallet or RPCs have been successfully broadcast. Every 10-15 minutes, the node will try to announce unbroadcast transactions until a peer requests it via a `getdata` message or the transaction is removed from the mempool for other reasons. The node will not track the broadcast status of transactions submitted to the node using P2P relay. This version reduces the initial broadcast guarantees for wallet transactions submitted via P2P to a node running the wallet. (#18038)

- The size of the set of transactions that peers have announced and we consider for requests has been reduced from 100000 to 5000 (per peer), and further announcements will be ignored when that limit is reached. If you need to dump (very) large batches of transactions, exceptions can be made for trusted peers using the "relay" network permission. For localhost for example it can be enabled using the command line option `-whitelist=relay@127.0.0.1` . (#19988)

- This release adds support for Tor version 3 hidden services, and rumoring them over the network to other peers using BIP155. Version 2 hidden services are still fully supported by Bitcoin Core, but the Tor network will start deprecating them in the coming months. (#19954)

- The Tor onion service that is automatically created by setting the `-listenonion` configuration parameter will now be created as a Tor v3 service instead of Tor v2. The private key that was used for Tor v2 (if any) will be left untouched in the `onion_private_key` file in the data directory (see `-datadir`) and can be removed if not needed. Bitcoin Core will no longer attempt to read it. The private key for the Tor v3 service will be saved in a file named `onion_v3_private_key` . To use the deprecated Tor v2 service (not recommended), the `onion_private_key` can be copied over `onion_v3_private_key` , e.g. `cp -f onion_private_key onion_v3_private_key` . (#19954)

- The client writes a file (`anchors.dat`) at shutdown with the network addresses of the node's two outbound block-relay-only peers (so called "anchors"). The next time the node starts, it reads this file and attempts to reconnect to those same two peers. This prevents an attacker from using node restarts to trigger a complete change in peers, which would be something they could use as part of an eclipse attack. (#17428)

- This release adds support for serving BIP157 compact filters to peers on the network when enabled using `-blockfilterindex=1 -peerblockfilters=1` . (#16442)

- This release adds support for signets (BIP325) in addition to the existing mainnet, testnet, and regtest networks. Signets are centrally-controlled test networks, allowing them to be more predictable test environments than the older testnet. One public signet is maintained, and selectable using `-signet` . It is also possible to create personal signets. (#18267).

- This release implements BIP339 wtxid relay. When negotiated, transactions are announced using their wtxid instead of their txid. (#18044).

- This release implements the proposed Taproot consensus rules (BIP341 and BIP342), without activation on mainnet. Experimentation with Taproot can be done on signet, where its rules are already active. (#19553)

Updated RPCs

- The `getpeerinfo` RPC has a new `network` field that provides the type of network ("ipv4", "ipv6", or "onion") that the peer connected through. (#20002)

- The `getpeerinfo` RPC now has additional `last_block` and `last_transaction` fields that return the UNIX epoch time of the last block and the last *valid* transaction received from each peer. (#19731)

- `getnetworkinfo` now returns two new fields, `connections_in` and `connections_out` , that provide the number of inbound and outbound peer connections. These new fields are in addition to the existing `connections` field, which returns the total number of peer connections. (#19405)

- Exposed transaction version numbers are now treated as unsigned 32-bit integers instead of signed 32-bit integers. This matches their treatment in consensus logic. Versions greater than 2 continue to be non-standard (matching previous behavior of smaller than 1 or greater than 2 being non-standard). Note that this includes the `joinpsbt` command, which combines partially-signed transactions by selecting the highest version number. (#16525)

- `getmempoolinfo` now returns an additional `unbroadcastcount` field. The mempool tracks locally submitted transactions until their initial broadcast is acknowledged by a peer. This field returns the count of transactions waiting for acknowledgement.
- Mempool RPCs such as `getmempoolentry` and `getrawmempool` with `verbose=true` now return an additional `unbroadcast` field. This indicates whether initial broadcast of the transaction has been acknowledged by a peer. `getmempoolancestors` and `getmempooldescendants` are also updated.
- The `getpeerinfo` RPC no longer returns the `banscore` field unless the configuration option `-deprecatedrpc=banscore` is used. The `banscore` field will be fully removed in the next major release. (#19469)
- The `testmempoolaccept` RPC returns `vsize` and a `fees` object with the `base` fee if the transaction would pass validation. (#19940)
- The `getpeerinfo` RPC now returns a `connection_type` field. This indicates the type of connection established with the peer. It will return one of six options. For more information, see the `getpeerinfo` help documentation. (#19725)
- The `getpeerinfo` RPC no longer returns the `addnode` field by default. This field will be fully removed in the next major release. It can be accessed with the configuration option `-deprecatedrpc=getpeerinfo_addnode`. However, it is recommended to instead use the `connection_type` field (it will return `manual` when addnode is true). (#19725)
- The `getpeerinfo` RPC no longer returns the `whitelisted` field by default. This field will be fully removed in the next major release. It can be accessed with the configuration option `-deprecatedrpc=getpeerinfo_whitelisted`. However, it is recommended to instead use the `permissions` field to understand if specific privileges have been granted to the peer. (#19770)
- The `walletcreatefundedpsbt` RPC call will now fail with `Insufficient funds` when inputs are manually selected but are not enough to cover the outputs and fee. Additional inputs can automatically be added through the new `add_inputs` option. (#16377)
- The `fundrawtransaction` RPC now supports `add_inputs` option that when `false` prevents adding more inputs if necessary and consequently the RPC fails.

Changes to Wallet or GUI related RPCs can be found in the GUI or Wallet section below.

New RPCs

- The `getindexinfo` RPC returns the actively running indices of the node, including their current sync status and height. It also accepts an `index_name` to specify returning the status of that index only. (#19550)

Build System

Updated settings

- The same ZeroMQ notification (e.g. `-zmqpubhashtx=address`) can now be specified multiple times to publish the same notification to different ZeroMQ sockets. (#18309)
- The `-banscore` configuration option, which modified the default threshold for disconnecting and discouraging misbehaving peers, has been removed as part of changes in 0.20.1 and in this release to the handling of misbehaving peers. Refer to "Changes regarding misbehaving peers" in the 0.20.1 release notes for details. (#19464)
- The `-debug=db` logging category, which was deprecated in 0.20 and replaced by `-debug=walletdb` to distinguish it from `coindb`, has been removed. (#19202)
- A `download` permission has been extracted from the `noban` permission. For compatibility, `noban` implies the `download` permission, but this may change in future releases. Refer to the help of the affected settings `-whitebind` and `-whitelist` for more details. (#19191)
- Netmasks that contain 1-bits after 0-bits (the 1-bits are not contiguous on the left side, e.g. 255.0.255.255) are no longer accepted. They are invalid according to RFC 4632. Netmasks are used in the `-rpcallowip` and `-whitelist` configuration options and in the `setban` RPC. (#19628)
- The `-blocksonly` setting now completely disables fee estimation. (#18766)

Changes to Wallet or GUI related settings can be found in the GUI or Wallet section below.

Tools and Utilities

- A new `bitcoin-cli -netinfo` command provides a network peer connections dashboard that displays data from the `getpeerinfo` and `getnetworkinfo` RPCs in a human-readable format. An optional integer argument from `0` to `4` may be passed to see increasing levels of detail. (#19643)
- A new `bitcoin-cli -generate` command, equivalent to RPC `generatenewaddress` followed by `generatetoaddress`, can generate blocks for command line testing purposes. This is a client-side version of the former `generate` RPC. See the help for details. (#19133)
- The `bitcoin-cli -getinfo` command now displays the wallet name and balance for each of the loaded wallets when more than one is loaded (e.g. in multiwallet mode) and a wallet is not specified with `-rpcwallet`. (#18594)
- The `connections` field of `bitcoin-cli -getinfo` is now expanded to return a JSON object with `in`, `out` and `total` numbers of peer connections. It previously returned a single integer value for the total number of peer connections. (#19405)

New settings

- The `startupnotify` option is used to specify a command to execute when Bitcoin Core has finished with its startup sequence. (#15367)

Wallet

- Backwards compatibility has been dropped for two `getaddressinfo` RPC deprecations, as notified in the 0.20 release notes. The deprecated `label` field has been removed as well as the deprecated `labels` behavior of returning a JSON object containing `name` and `purpose` key-value pairs. Since 0.20, the `labels` field returns a JSON array of label names. (#19200)

- To improve wallet privacy, the frequency of wallet rebroadcast attempts is reduced from approximately once every 15 minutes to once every 12-36 hours. To maintain a similar level of guarantee for initial broadcast of wallet transactions, the mempool tracks these transactions as a part of the newly introduced unbroadcast set. See the "P2P and network changes" section for more information on the unbroadcast set. (#18038)
- The `sendtoaddress` and `sendmany` RPCs accept an optional `verbose=True` argument to also return the fee reason about the sent tx. (#19501)
- The wallet can create a transaction without change even when the keypool is empty. Previously it failed. (#17219)
- The `-salvagewallet` startup option has been removed. A new `salvage` command has been added to the `bitcoin-wallet` tool which performs the salvage operations that `-salvagewallet` did. (#18918)
- A new configuration flag `-maxapsfee` has been added, which sets the max allowed avoid partial spends (APS) fee. It defaults to 0 (i.e. fee is the same with and without APS). Setting it to -1 will disable APS, unless `-avoidpartialspends` is set. (#14582)
- The wallet will now avoid partial spends (APS) by default, if this does not result in a difference in fees compared to the non-APS variant. The allowed fee threshold can be adjusted using the new `-maxapsfee` configuration option. (#14582)
- The `createwallet`, `loadwallet`, and `unloadwallet` RPCs now accept `load_on_startup` options to modify the settings list. Unless these options are explicitly set to true or false, the list is not modified, so the RPC methods remain backwards compatible. (#15937)
- A new `send` RPC with similar syntax to `walletcreatefundedpsbt`, including support for coin selection and a custom fee rate, is added. The `send` RPC is experimental and may change in subsequent releases. (#16378)
- The `estimate_mode` parameter is now case-insensitive in the `bumpfee`, `fundrawtransaction`, `sendmany`, `sendtoaddress`, `send` and `walletcreatefundedpsbt` RPCs. (#11413)
- The `bumpfee` RPC now uses `conf_target` rather than `confTarget` in the options. (#11413)
- `fundrawtransaction` and `walletcreatefundedpsbt` when used with the `lockUnspents` argument now lock manually selected coins, in addition to automatically selected coins. Note that locked coins are never used in automatic coin selection, but can still be manually selected. (#18244)
- The `-zapwallettxes` startup option has been removed and its functionality removed from the wallet. This option was originally intended to allow for rescuing wallets which were affected by a malleability attack. More recently, it has been used in the fee bumping of transactions that did not signal RBF. This functionality has been superseded with the abandon transaction feature. (#19671)
- The error code when no wallet is loaded, but a wallet RPC is called, has been changed from `-32601` (method not found) to `-18` (wallet not found). (#20101)

Automatic wallet creation removed

Bitcoin Core will no longer automatically create new wallets on startup. It will load existing wallets specified by `-wallet` options on the command line or in `bitcoin.conf` or `settings.json` files. And by default it will also load a top-level unnamed ("") wallet. However, if specified wallets don't exist, Bitcoin Core will now just log warnings instead of creating new wallets with new keys and addresses like previous releases did.

New wallets can be created through the GUI (which has a more prominent create wallet option), through the `bitcoin-cli createwallet` or `bitcoin-wallet create` commands, or the `createwallet` RPC. (#15454, #20186)

Experimental Descriptor Wallets

Please note that Descriptor Wallets are still experimental and not all expected functionality is available. Additionally there may be some bugs and current functions may change in the future. Bugs and missing functionality can be reported to the issue tracker.

0.21 introduces a new type of wallet - Descriptor Wallets. Descriptor Wallets store scriptPubKey information using output descriptors. This is in contrast to the Legacy Wallet structure where keys are used to implicitly generate scriptPubKeys and addresses. Because of this shift to being script based instead of key based, many of the confusing things that Legacy Wallets do are not possible with Descriptor Wallets. Descriptor Wallets use a definition of "mine" for scripts which is simpler and more intuitive than that used by Legacy Wallets. Descriptor Wallets also uses different semantics for watch-only things and imports.

As Descriptor Wallets are a new type of wallet, their introduction does not affect existing wallets. Users who already have a Bitcoin Core wallet can continue to use it as they did before without any change in behavior. Newly created Legacy Wallets (which remains the default type of wallet) will behave as they did in previous versions of Bitcoin Core.

The differences between Descriptor Wallets and Legacy Wallets are largely limited to non user facing things. They are intended to behave similarly except for the import/export and watchonly functionality as described below.

Creating Descriptor Wallets

Descriptor wallets are not the default type of wallet.

In the GUI, a checkbox has been added to the Create Wallet Dialog to indicate that a Descriptor Wallet should be created. And a `descriptors` option has been added to `createwallet` RPC. Setting `descriptors` to `true` will create a Descriptor Wallet instead of a Legacy Wallet.

Without those options being set, a Legacy Wallet will be created instead.

`IsMine` Semantics

`IsMine` refers to the function used to determine whether a script belongs to the wallet. This is used to determine whether an output belongs to the wallet. `IsMine` in Legacy Wallets returns true if the wallet would be able to sign an input that spends an output with that script. Since keys can be involved in a variety of different scripts, this definition for `IsMine` can lead to many unexpected scripts being considered part of the wallet.

With Descriptor Wallets, descriptors explicitly specify the set of scripts that are owned by the wallet. Since descriptors are deterministic and easily enumerable, users will know exactly what scripts the wallet will consider to belong to it. Additionally the implementation of `IsMine` in Descriptor Wallets is far simpler than for Legacy Wallets. Notably, in Legacy Wallets, `IsMine` allowed for users to take one type of address (e.g. P2PKH), mutate it into another address type (e.g. P2WPKH), and the wallet would still detect outputs sending to the new address type even without that address being requested from the wallet. Descriptor Wallets do not allow for this and will only watch for the addresses that were explicitly requested from the wallet.

These changes to `IsMine` will make it easier to reason about what scripts the wallet will actually be watching for in outputs. However for the vast majority of users, this change is largely transparent and will not have noticeable effect.

Imports and Exports

In Legacy Wallets, raw scripts and keys could be imported to the wallet. Those imported scripts and keys are treated separately from the keys generated by the wallet. This complicates the `IsMine` logic as it has to distinguish between spendable and watchonly.

Descriptor Wallets handle importing scripts and keys differently. Only complete descriptors can be imported. These descriptors are then added to the wallet as if it were a descriptor generated by the wallet itself. This simplifies the `IsMine` logic so that it no longer has to distinguish between spendable and watchonly. As such, the watchonly model for Descriptor Wallets is also different and described in more detail in the next section.

To import into a Descriptor Wallet, a new `importdescriptors` RPC has been added that uses a syntax similar to that of `importmulti`.

As Legacy Wallets and Descriptor Wallets use different mechanisms for storing and importing scripts and keys the existing import RPCs have been disabled for descriptor wallets. New export RPCs for Descriptor Wallets have not yet been added.

The following RPCs are disabled for Descriptor Wallets:

- `importprivkey`
- `importpubkey`
- `importaddress`
- `importwallet`
- `dumpprivkey`
- `dumpwallet`
- `importmulti`
- `addmultisigaddress`
- `sethdseed`

Watchonly Wallets

A Legacy Wallet contains both private keys and scripts that were being watched. Those watched scripts would not contribute to your normal balance. In order to see the watchonly balance and to use watchonly things in transactions, an `include_watchonly` option was added to many RPCs that would allow users to do that. However it is easy to forget to include this option.

Descriptor Wallets move to a per-wallet watchonly model. Instead an entire wallet is considered to be watchonly depending on whether it was created with private keys disabled. This eliminates the need to distinguish between things that are watchonly and things that are not within a wallet itself.

This change does have a caveat. If a Descriptor Wallet with private keys *enabled* has a multiple key descriptor without all of the private keys (e.g. `multi(...)` with only one private key), then the wallet will fail to sign and broadcast transactions. Such wallets would need to use the PSBT workflow but the typical GUI Send, `sendtoaddress`, etc. workflows would still be available, just non-functional.

This issue is worsened if the wallet contains both single key (e.g. `wpkh(...)`) descriptors and such multiple key descriptors as some transactions could be signed and broadcast and others not. This is due to some transactions containing only single key inputs, while others would contain both single key and multiple key inputs, depending on which are available and how the coin selection algorithm selects inputs. However this is not considered to be a supported use case; multisigs should be in their own wallets which do not already have descriptors. Although users cannot export descriptors with private keys for now as explained earlier.

BIP 44/49/84 Support

The change to using descriptors changes the default derivation paths used by Bitcoin Core to adhere to BIP 44/49/84. Descriptors with different derivation paths can be imported without issue.

SQLite Database Backend

Descriptor wallets use SQLite for the wallet file instead of the Berkeley DB used in legacy wallets. This will break compatibility with any existing tooling that operates on wallets, however compatibility was already being broken by the move to descriptors.

Wallet RPC changes

- The `upgradewallet` RPC replaces the `-upgradewallet` command line option. (#15761)
- The `settxfee` RPC will fail if the fee was set higher than the `-maxtxfee` command line setting. The wallet will already fail to create transactions with fees higher than `-maxtxfee`. (#18467)
- A new `fee_rate` parameter/option denominated in satoshis per vbyte (sat/vB) is introduced to the `sendtoaddress`, `sendmany`, `fundrawtransaction` and `walletcreatefundedpsbt` RPCs as well as to the experimental new `send` RPC. The legacy `feeRate` option in `fundrawtransaction` and `walletcreatefundedpsbt` still exists for setting a fee rate in BTC per 1,000 vbytes (BTC/kvB), but it is expected to be deprecated soon to avoid confusion. For these RPCs, the fee rate error message is updated from BTC/kB to sat/vB and the help documentation in BTC/kB is updated to BTC/kvB. The `send` and `sendtoaddress` RPC examples are updated to aid users in creating transactions with explicit fee rates. (#20305, #11413)
- The `bumpfee` RPC `fee_rate` option is changed from BTC/kvB to sat/vB and the help documentation is updated. Users are warned that this is a breaking API change, but it should be relatively benign: the large (100,000 times) difference between BTC/kvB and sat/vB units means that a transaction with a fee rate mistakenly calculated in BTC/kvB rather than sat/vB should raise an error due to the fee rate being set too low. In the worst case, the transaction may send at 1 sat/vB, but as Replace-by-Fee (BIP125 RBF) is active by default when an explicit fee rate is used, the transaction fee can be bumped. (#20305)

GUI changes

- Wallets created or loaded in the GUI will now be automatically loaded on startup, so they don't need to be manually reloaded next time Bitcoin Core is started. The list of wallets to load on startup is stored in `\<datadir\>/settings.json` and augments any command line or `bitcoin.conf` `-wallet=` settings that specify more wallets to load. Wallets that are unloaded in the GUI get removed from the settings list so they won't load again automatically next startup. (#19754)
- The GUI Peers window no longer displays a "Ban Score" field. This is part of changes in 0.20.1 and in this release to the handling of misbehaving peers. Refer to "Changes regarding misbehaving peers" in the 0.20.1 release notes for details. (#19512)

Low-level changes

RPC

- To make RPC `sendtoaddress` more consistent with `sendmany` the following error `sendtoaddress` codes were changed from `-4` to `-6` :
  - Insufficient funds
  - Fee estimation failed
  - Transaction has too long of a mempool chain
- The `sendrawtransaction` error code for exceeding `maxfeerate` has been changed from `-26` to `-25` . The error string has been changed from "absurdly-high-fee" to "Fee exceeds maximum configured by user (e.g. -maxtxfee, maxfeerate)." The `testmempoolaccept` RPC returns `max-fee-exceeded` rather than `absurdly-high-fee` as the `reject-reason` . (#19339)
- To make wallet and rawtransaction RPCs more consistent, the error message for exceeding maximum feerate has been changed to "Fee exceeds maximum configured by user (e.g. -maxtxfee, maxfeerate)." (#19339)

Tests

- The BIP 325 default signet can be enabled by the `-chain=signet` or `-signet` setting. The settings `-signetchallenge` and `-signetseednode` allow enabling a custom signet.
- The `generateblock` RPC allows testers using regtest mode to generate blocks that consist of a custom set of transactions. (#17693)

0.21.0 change log

Consensus

- #18267 BIP-325: Signet (kallewoof)
- #20016 uint256: 1 is a constant (ajtowns)
- #20006 Fix misleading error message: Clean stack rule (sanket1729)
- #19953 Implement BIP 340-342 validation (Schnorr/taproot/tapscript) (sipa)
- #20169 Taproot follow-up: Make ComputeEntrySchnorr and ComputeEntryECDSA const to clarify contract (practicalswift)

Policy

- #18766 Disable fee estimation in blocksonly mode (darosior)
- #19630 Cleanup fee estimation code (darosior)
- #20165 Only relay Taproot spends if next block has it active (sipa)

Mining

- #17946 Fix GBT: Restore "!segwit" and "csv" to "rules" key (luke-jr)

Privacy

- #16432 Add privacy to the Overview page (hebasto)
- #18861 Do not answer GETDATA for to-be-announced tx (sipa)
- #18038 Mempool tracks locally submitted transactions to improve wallet privacy (amitiuttarwar)
- #19109 Only allow getdata of recently announced invvs (sipa)

Block and transaction handling

- #17737 Add ChainstateManager, remove BlockManager global (jamesob)
- #18960 indexes: Add compact block filter headers cache (jnewbery)
- #13204 Faster sigcache nonce (JeremyRubin)
- #19088 Use std::chrono throughout some validation functions (fanquake)
- #19142 Make VerifyDB level 4 interruptible (MarcoFalke)
- #17994 Flush undo files after last block write (kallewoof)
- #18990 log: Properly log txs rejected from mempool (MarcoFalke)
- #18984 Remove unnecessary input blockfile SetPos (dgenr8)
- #19526 log: Avoid treating remote misbehvior as local system error (MarcoFalke)
- #18044 Use wtxid for transaction relay (sdaftuar)
- #18637 coins: allow cache resize after init (jamesob)
- #19854 Avoid locking CTxMemPool::cs recursively in simple cases (hebasto)
- #19478 Remove CTxMempool::mapLinks data structure member (JeremyRubin)
- #19927 Reduce direct `g_chainman` usage (dongcarl)
- #19898 log: print unexpected version warning in validation log category (n-thumann)
- #20036 signet: Add assumed values for default signet (MarcoFalke)
- #20048 chainparams: do not log signet startup messages for other chains (jonatack)
- #19339 re-delegate absurd fee checking from mempool to clients (glozow)
- #20035 signet: Fix uninitialized read in validation (MarcoFalke)
- #20157 Bugfix: chainparams: Add missing (always enabled) Taproot deployment for Signet (luke-jr)
- #20263 Update assumed chain params (MarcoFalke)
- #20372 Avoid signed integer overflow when loading a mempool.dat file with a malformed time field (practicalswift)
- #18621 script: Disallow silent bool -> cscript conversion (MarcoFalke)
- #18612, #18732 script: Remove undocumented and unused operator+ (MarcoFalke)
- #19317 Add a left-justified width field to `log2_work` component for a uniform debug.log output (jamesgmorgan)

P2P protocol and network code

- #18544 Limit BIP37 filter lifespan (active between `filterload` .. `filterclear`) (theStack)
- #18806 Remove is{Empty,Full} flags from CBloomFilter, clarify CVE fix (theStack)
- #18512 Improve asmap checks and add sanity check (sipa)
- #18877 Serve cfcheckpt requests (jnewbery)
- #18895 Unbroadcast followups: rpcs, nLastResend, mempool sanity check (gzhao408)
- #19010 net processing: Add support for `getcfheaders` (jnewbery)
- #16939 Delay querying DNS seeds (ajtowns)
- #18807 Unbroadcast follow-ups (amitiuttarwar)
- #19044 Add support for getcfilters (jnewbery)
- #19084 improve code documentation for dns seed behaviour (ajtowns)
- #19260 disconnect peers that send filterclear + update existing filter msg disconnect logic (gzhao408)
- #19284 Add seed.bitcoin.wiz.biz to DNS seeds (wiz)
- #19322 split PushInventory() (jnewbery)
- #19204 Reduce inv traffic during IBD (MarcoFalke)
- #19470 banlist: log post-swept banlist size at startup (fanquake)
- #19191 Extract download permission from noban (MarcoFalke)
- #14033 Drop `CADDR_TIME_VERSION` checks now that `MIN_PEER_PROTO_VERSION` is greater (Empact)
- #19464 net, rpc: remove -banscore option, deprecate banscore in getpeerinfo (jonatack)
- #19514 [net/net processing] check banman pointer before dereferencing (jnewbery)

- #19512 banscore updates to gui, tests, release notes (jonatack)
- #19360 improve encapsulation of CNetAddr (vasild)
- #19217 disambiguate block-relay-only variable names from blocksonly variables (glowang)
- #19473 Add -networkactive option (hebasto)
- #19472 [net processing] Reduce `cs_main` scope in MaybeDiscourageAndDisconnect() (jnewbery)
- #19583 clean up Misbehaving() (jnewbery)
- #19534 save the network type explicitly in CNetAddr (vasild)
- #19569 Enable fetching of orphan parents from wtxid peers (sipa)
- #18991 Cache responses to GETADDR to prevent topology leaks (naumenkogs)
- #19596 Deduplicate parent txid loop of requested transactions and missing parents of orphan transactions (sdaftuar)
- #19316 Cleanup logic around connection types (amitiuttarwar)
- #19070 Signal support for compact block filters with `NODE_COMPACT_FILTERS` (jnewbery)
- #19705 Shrink CAddress from 48 to 40 bytes on x64 (vasild)
- #19704 Move ProcessMessage() to PeerLogicValidation (jnewbery)
- #19628 Change CNetAddr::ip to have flexible size (vasild)
- #19797 Remove old check for 3-byte shifted IP addresses from pre-0.2.9 nodes (#19797)
- #19607 Add Peer struct for per-peer data in net processing (jnewbery)
- #19857 improve nLastBlockTime and nLastTXTime documentation (jonatack)
- #19724 Cleanup connection types- followups (amitiuttarwar)
- #19670 Protect localhost and block-relay-only peers from eviction (sdaftuar)
- #19728 Increase the ip address relay branching factor for unreachable networks (sipa)
- #19879 Miscellaneous wtxid followups (amitiuttarwar)
- #19697 Improvements on ADDR caching (naumenkogs)
- #17785 Unify Send and Receive protocol versions (hebasto)
- #19845 CNetAddr: add support to (un)serialize as ADDRv2 (vasild)
- #19107 Move all header verification into the network layer, extend logging (troygiorshev)
- #20003 Exit with error message if -proxy is specified without arguments (instead of continuing without proxy server) (practicalswift)
- #19991 Use alternative port for incoming Tor connections (hebasto)
- #19723 Ignore unknown messages before VERACK (sdaftuar)
- #19954 Complete the BIP155 implementation and upgrade to TORv3 (vasild)
- #20119 BIP155 follow-ups (sipa)
- #19988 Overhaul transaction request logic (sipa)
- #17428 Try to preserve outbound block-relay-only connections during restart (hebasto)
- #19911 Guard `vRecvGetData` with `cs_vRecv` and `orphan_work_set` with `g_cs_orphans` (narula)
- #19753 Don't add AlreadyHave transactions to recentRejects (troygiorshev)
- #20187 Test-before-evict bugfix and improvements for block-relay-only peers (sdaftuar)
- #20237 Hardcoded seeds update for 0.21 (laanwj)
- #20212 Fix output of peer address in version message (vasild)
- #20284 Ensure old versions don't parse peers.dat (vasild)
- #20405 Avoid calculating onion address checksum when version is not 3 (lontivero)
- #20564 Don't send 'sendaddrv2' to pre-70016 software, and send before 'verack' (sipa)
- #20660 Move signet onion seed from v2 to v3 (Sjors)

Wallet

- #18262 Exit selection when `best_waste` is 0 (achow101)
- #17824 Prefer full destination groups in coin selection (fjahr)
- #17219 Allow transaction without change if keypool is empty (Sjors)

- #15761 Replace -upgradewallet startup option with upgradewallet RPC (achow101)
- #18671 Add BlockUntilSyncedToCurrentChain to dumpwallet (MarcoFalke)
- #16528 Native Descriptor Wallets using DescriptorScriptPubKeyMan (achow101)
- #18777 Recommend absolute path for dumpwallet (MarcoFalke)
- #16426 Reverse `cs_main`, `cs_wallet` lock order and reduce `cs_main` locking (ariard)
- #18699 Avoid translating RPC errors (MarcoFalke)
- #18782 Make sure no DescriptorScriptPubKeyMan or WalletDescriptor members are left uninitialized after construction (practicalswift)
- #9381 Remove CWalletTx merging logic from AddToWallet (ryanofsky)
- #16946 Include a checksum of encrypted private keys (achow101)
- #17681 Keep inactive seeds after sethdseed and derive keys from them as needed (achow101)
- #18918 Move salvagewallet into wallettool (achow101)
- #14988 Fix for confirmed column in csv export for payment to self transactions (benthecarman)
- #18275 Error if an explicit fee rate was given but the needed fee rate differed (kallewoof)
- #19054 Skip hdKeypath of 'm' when determining inactive hd seeds (achow101)
- #17938 Disallow automatic conversion between disparate hash types (Empact)
- #19237 Check size after unserializing a pubkey (elichai)
- #11413 sendtoaddress/sendmany: Add explicit feerate option (kallewoof)
- #18850 Fix ZapSelectTx to sync wallet spends (bvbfan)
- #18923 Never schedule MaybeCompactWalletDB when `-flushwallet` is off (MarcoFalke)
- #19441 walletdb: Don't reinitialize desc cache with multiple cache entries (achow101)
- #18907 walletdb: Don't remove database transaction logs and instead error (achow101)
- #19334 Introduce WalletDatabase abstract class (achow101)
- #19335 Cleanup and separate BerkeleyDatabase and BerkeleyBatch (achow101)
- #19102 Introduce and use DummyDatabase instead of dummy BerkeleyDatabase (achow101)
- #19568 Wallet should not override signing errors (fjahr)
- #17204 Do not turn `OP_1NEGATE` in scriptSig into `0x0181` in signing code (sipa) (meshcollider)
- #19457 Cleanup wallettool salvage and walletdb extraneous declarations (achow101)
- #15937 Add loadwallet and createwallet `load_on_startup` options (ryanofsky)
- #16841 Replace GetScriptForWitness with GetScriptForDestination (meshcollider)
- #14582 always do avoid partial spends if fees are within a specified range (kallewoof)
- #19743 -maxapsfee follow-up (kallewoof)
- #19289 GetWalletTx and IsMine require `cs_wallet` lock (promag)
- #19671 Remove -zapwallettxes (achow101)
- #19805 Avoid deserializing unused records when salvaging (achow101)
- #19754 wallet, gui: Reload previously loaded wallets on startup (achow101)
- #19738 Avoid multiple BerkeleyBatch in DelAddressBook (promag)
- #19919 bugfix: make LoadWallet assigns status always (AkioNak)
- #16378 The ultimate send RPC (Sjors)
- #15454 Remove the automatic creation and loading of the default wallet (achow101)
- #19501 `send*` RPCs in the wallet returns the "fee reason" (stackman27)
- #20130 Remove db mode string (S3RK)
- #19077 Add sqlite as an alternative wallet database and use it for new descriptor wallets (achow101)
- #20125 Expose database format in getwalletinfo (promag)
- #20198 Show name, format and if uses descriptors in bitcoin-wallet tool (jonasschnelli)
- #20216 Fix buffer over-read in SQLite file magic check (theStack)
- #20186 Make -wallet setting not create wallets (ryanofsky)
- #20230 Fix bug when just created encrypted wallet cannot get address (hebasto)
- #20282 Change `upgradewallet` return type to be an object (jnewbery)
- #20220 Explicit fee rate follow-ups/fixes for 0.21 (jonatack)

- #20199 Ignore (but warn) on duplicate -wallet parameters (jonasschnelli)
- #20324 Set DatabaseStatus::SUCCESS in MakeSQLiteDatabase (MarcoFalke)
- #20266 Fix change detection of imported internal descriptors (achow101)
- #20153 Do not import a descriptor with hardened derivations into a watch-only wallet (S3RK)
- #20344 Fix scanning progress calculation for single block range (theStack)
- #19502 Bugfix: Wallet: Soft-fail exceptions within ListWalletDir file checks (luke-jr)
- #20378 Fix potential division by 0 in WalletLogPrintf (jonasschnelli)
- #18836 Upgradewallet fixes and additional tests (achow101)
- #20139 Do not return warnings from UpgradeWallet() (stackman27)
- #20305 Introduce `fee_rate` sat/vB param/option (jonatack)
- #20426 Allow zero-fee fundrawtransaction/walletcreatefundedpsbt and other fixes (jonatack)
- #20573 wallet, bugfix: allow send with string `fee_rate` amounts (jonatack)

RPC and other APIs

- #18574 cli: Call getbalances.ismine.trusted instead of getwalletinfo.balance (jonatack)
- #17693 Add `generateblock` to mine a custom set of transactions (andrewtoth)
- #18495 Remove deprecated migration code (vasild)
- #18493 Remove deprecated "size" from mempool txs (vasild)
- #18467 Improve documentation and return value of settxfee (fjahr)
- #18607 Fix named arguments in documentation (MarcoFalke)
- #17831 doc: Fix and extend getblockstats examples (asoltys)
- #18785 Prevent valgrind false positive in `rest_blockhash_by_height` (ryanofsky)
- #18999 log: Remove "No rpcpassword set" from logs (MarcoFalke)
- #19006 Avoid crash when `g_thread_http` was never started (MarcoFalke)
- #18594 cli: Display multiwallet balances in -getinfo (jonatack)
- #19056 Make gettxoutsetinfo/GetUTXOStats interruptible (MarcoFalke)
- #19112 Remove special case for unknown service flags (MarcoFalke)
- #18826 Expose txinwitness for coinbase in JSON form from RPC (rvagg)
- #19282 Rephrase generatetoaddress help, and use `PACKAGE_NAME` (luke-jr)
- #16377 don't automatically append inputs in walletcreatefundedpsbt (Sjors)
- #19200 Remove deprecated getaddressinfo fields (jonatack)
- #19133 rpc, cli, test: add bitcoin-cli -generate command (jonatack)
- #19469 Deprecate banscore field in getpeerinfo (jonatack)
- #16525 Dump transaction version as an unsigned integer in RPC/TxToUniv (TheBlueMatt)
- #19555 Deduplicate WriteHDKeypath() used in decodepsbt (theStack)
- #19589 Avoid useless mempool query in gettxoutproof (MarcoFalke)
- #19585 RPCResult Type of MempoolEntryDescription should be OBJ (stylesuxx)
- #19634 Document getwalletinfo's `unlocked_until` field as optional (justinmoon)
- #19658 Allow RPC to fetch all addrman records and add records to addrman (jnewbery)
- #19696 Fix addnode remove command error (fjahr)
- #18654 Separate bumpfee's psbt creation function into psbtbumpfee (achow101)
- #19655 Catch listsinceblock `target_confirmations` exceeding block count (adaminsky)
- #19644 Document returned error fields as optional if applicable (theStack)
- #19455 rpc generate: print useful help and error message (jonatack)
- #19550 Add listindices RPC (fjahr)
- #19169 Validate provided keys for `query_options` parameter in listunspent (PastaPastaPasta)
- #18244 fundrawtransaction and walletcreatefundedpsbt also lock manually selected coins (Sjors)
- #14687 zmq: Enable TCP keepalive (mruddy)
- #19405 Add network in/out connections to `getnetworkinfo` and `-getinfo` (jonatack)
- #19878 rawtransaction: Fix argument in combinerawtransaction help message (pinheadmz)

- #19940 Return fee and vsize from testmempoolaccept (gzhao408)
- #13686 zmq: Small cleanups in the ZMQ code (domob1812)
- #19386, #19528, #19717, #19849, #19994 Assert that RPCArg names are equal to CRPCCommand ones (MarcoFalke)
- #19725 Add connection type to getpeerinfo, improve logs (amitiuttarwar)
- #19969 Send RPC bug fix and touch-ups (Sjors)
- #18309 zmq: Add support to listen on multiple interfaces (n-thumann)
- #20055 Set HTTP Content-Type in bitcoin-cli (laanwj)
- #19956 Improve invalid vout value rpc error message (n1rna)
- #20101 Change no wallet loaded message to be clearer (achow101)
- #19998 Add `via_tor` to `getpeerinfo` output (hebasto)
- #19770 getpeerinfo: Deprecate "whitelisted" field (replaced by "permissions") (luke-jr)
- #20120 net, rpc, test, bugfix: update GetNetworkName, GetNetworksInfo, regression tests (jonatack)
- #20595 Improve heuristic hex transaction decoding (sipa)
- #20731 Add missing description of vout in getrawtransaction help text (benthecarman)
- #19328 Add gettxoutsetinfo `hash_type` option (fjahr)
- #19731 Expose nLastBlockTime/nLastTXTime as last `block/last_transaction` in getpeerinfo (jonatack)
- #19572 zmq: Create "sequence" notifier, enabling client-side mempool tracking (instagibbs)
- #20002 Expose peer network in getpeerinfo; simplify/improve -netinfo (jonatack)

GUI

- #17905 Avoid redundant tx status updates (ryanofsky)
- #18646 Use `PACKAGE_NAME` in exception message (fanquake)
- #17509 Save and load PSBT (Sjors)
- #18769 Remove bug fix for Qt < 5.5 (10xcryptodev)
- #15768 Add close window shortcut (IPGlider)
- #16224 Bilingual GUI error messages (hebasto)
- #18922 Do not translate InitWarning messages in debug.log (hebasto)
- #18152 Use NotificationStatus enum for signals to GUI (hebasto)
- #18587 Avoid wallet tryGetBalances calls in WalletModel::pollBalanceChanged (ryanofsky)
- #17597 Fix height of QR-less ReceiveRequestDialog (hebasto)
- #17918 Hide non PKHash-Addresses in signing address book (emilengler)
- #17956 Disable unavailable context menu items in transactions tab (kristapsk)
- #17968 Ensure that ModalOverlay is resized properly (hebasto)
- #17993 Balance/TxStatus polling update based on last block hash (furszy)
- #18424 Use parent-child relation to manage lifetime of OptionsModel object (hebasto)
- #18452 Fix shutdown when `waitfor*` cmds are called from RPC console (hebasto)
- #15202 Add Close All Wallets action (promag)
- #19132 lock `cs_main`, `m_cached_tip_mutex` in that order (vasild)
- #18898 Display warnings as rich text (hebasto)
- #19231 add missing translation.h include to fix build (fanquake)
- #18027 "PSBT Operations" dialog (gwillen)
- #19256 Change combiner for signals to `optional_last_value` (fanquake)
- #18896 Reset toolbar after all wallets are closed (hebasto)
- #18993 increase console command max length (10xcryptodev)
- #19323 Fix regression in *txoutset* in GUI console (hebasto)
- #19210 Get rid of cursor in out-of-focus labels (hebasto)
- #19011 Reduce `cs_main` lock accumulation during GUI startup (jonasschnelli)
- #19844 Remove usage of boost::bind (fanquake)

- #20479 Fix QPainter non-determinism on macOS (0.21 backport) (laanwj)
- gui#6 Do not truncate node flag strings in debugwindow peers details tab (Saibato)
- gui#8 Fix regression in TransactionTableModel (hebasto)
- gui#17 doc: Remove outdated comment in TransactionTablePriv (MarcoFalke)
- gui#20 Wrap tooltips in the intro window (hebasto)
- gui#30 Disable the main window toolbar when the modal overlay is shown (hebasto)
- gui#34 Show permissions instead of whitelisted (laanwj)
- gui#35 Parse params directly instead of through node (ryanofsky)
- gui#39 Add visual accenting for the 'Create new receiving address' button (hebasto)
- gui#40 Clarify block height label (hebasto)
- gui#43 bugfix: Call setWalletActionsEnabled(true) only for the first wallet (hebasto)
- gui#97 Relax GUI freezes during IBD (jonasschnelli)
- gui#71 Fix visual quality of text in QR image (hebasto)
- gui#96 Slight improve create wallet dialog (Sjors)
- gui#102 Fix SplashScreen crash when run with -disablewallet (hebasto)
- gui#116 Fix unreasonable default size of the main window without loaded wallets (hebasto)
- gui#120 Fix multiwallet transaction notifications (promag)

Build system

- #18504 Drop bitcoin-tx and bitcoin-wallet dependencies on libevent (ryanofsky)
- #18586 Bump gitian descriptors to 0.21 (laanwj)
- #17595 guix: Enable building for `x86_64-w64-mingw32` target (dongcarl)
- #17929 add linker optimisation flags to gitian & guix (Linux) (fanquake)
- #18556 Drop make dist in gitian builds (hebasto)
- #18088 ensure we aren't using GNU extensions (fanquake)
- #18741 guix: Make source tarball using git-archive (dongcarl)
- #18843 warn on potentially uninitialized reads (vasild)
- #17874 make linker checks more robust (fanquake)
- #18535 remove -Qunused-arguments workaround for clang + ccache (fanquake)
- #18743 Add --sysroot option to mac os native compile flags (ryanofsky)
- #18216 test, build: Enable -Werror=sign-compare (Empact)
- #18928 don't pass -w when building for Windows (fanquake)
- #16710 Enable -Wsuggest-override if available (hebasto)
- #18738 Suppress -Wdeprecated-copy warnings (hebasto)
- #18862 Remove fdelt_chk back-compat code and sanity check (fanquake)
- #18887 enable -Werror=gnu (vasild)
- #18956 enforce minimum required Windows version (7) (fanquake)
- #18958 guix: Make V=1 more powerful for debugging (dongcarl)
- #18677 Multiprocess build support (ryanofsky)
- #19094 Only allow ASCII identifiers (laanwj)
- #18820 Propagate well-known vars into depends (dongcarl)
- #19173 turn on --enable-c++17 by --enable-fuzz (vasild)
- #18297 Use pkg-config in BITCOIN_QT_CONFIGURE for all hosts including Windows (hebasto)
- #19301 don't warn when doxygen isn't found (fanquake)
- #19240 macOS toolchain simplification and bump (dongcarl)
- #19356 Fix search for brew-installed BDB 4 on OS X (gwillen)
- #19394 Remove unused `RES_IMAGES` (Bushstar)
- #19403 improve `__builtin_clz*` detection (fanquake)
- #19375 target Windows 7 when building libevent and fix ipv6 usage (fanquake)
- #19331 Do not include server symbols in wallet (MarcoFalke)

- #19257 remove BIP70 configure option (fanquake)
- #18288 Add MemorySanitizer (MSan) in Travis to detect use of uninitialized memory (practicalswift)
- #18307 Require pkg-config for all of the hosts (hebasto)
- #19445 Update msvc build to use ISO standard C++17 (sipsorcery)
- #18882 fix -Wformat-security check when compiling with GCC (fanquake)
- #17919 Allow building with system clang (dongcarl)
- #19553 pass -fcommon when building genisoimage (fanquake)
- #19565 call `AC_PATH_TOOL` for dsymutil in macOS cross-compile (fanquake)
- #19530 build LTO support into Apple's ld64 (theuni)
- #19525 add -Wl,-z,separate-code to hardening flags (fanquake)
- #19667 set minimum required Boost to 1.58.0 (fanquake)
- #19672 make clean removes .gcda and .gcno files from fuzz directory (Crypt-iQ)
- #19622 Drop ancient hack in gitian-linux descriptor (hebasto)
- #19688 Add support for llvm-cov (hebasto)
- #19718 Add missed gcov files to 'make clean' (hebasto)
- #19719 Add Werror=range-loop-analysis (MarcoFalke)
- #19015 Enable some commonly enabled compiler diagnostics (practicalswift)
- #19689 build, qt: Add Qt version checking (hebasto)
- #17396 modest Android improvements (icota)
- #18405 Drop all of the ZeroMQ patches (hebasto)
- #15704 Move Win32 defines to configure.ac to ensure they are globally defined (luke-jr)
- #19761 improve sed robustness by not using sed (fanquake)
- #19758 Drop deprecated and unused `GUARDED_VAR` and `PT_GUARDED_VAR` annotations (hebasto)
- #18921 add stack-clash and control-flow protection options to hardening flags (fanquake)
- #19803 Bugfix: Define and use `HAVE_FDATASYNC` correctly outside LevelDB (luke-jr)
- #19685 CMake invocation cleanup (dongcarl)
- #19861 add /usr/local/ to `LCOV_FILTER_PATTERN` for macOS builds (Crypt-iQ)
- #19916 allow user to specify `DIR_FUZZ_SEED_CORPUS` for `cov_fuzz` (Crypt-iQ)
- #19944 Update secp256k1 subtree (including BIP340 support) (sipa)
- #19558 Split pthread flags out of ldflags and dont use when building libconsensus (fanquake)
- #19959 patch qt libpng to fix powerpc build (fanquake)
- #19868 Fix target name (hebasto)
- #19960 The vcpkg tool has introduced a proper way to use manifests (sipsorcery)
- #20065 fuzz: Configure check for main function (MarcoFalke)
- #18750 Optionally skip external warnings (vasild)
- #20147 Update libsecp256k1 (endomorphism, test improvements) (sipa)
- #20156 Make sqlite support optional (compile-time) (luke-jr)
- #20318 Ensure source tarball has leading directory name (MarcoFalke)
- #20447 Patch `qt_intersect_spans` to avoid non-deterministic behavior in LLVM 8 (achow101)
- #20505 Avoid secp256k1.h include from system (dergoegge)
- #20527 Do not ignore Homebrew's SQLite on macOS (hebasto)
- #20478 Don't set BDB flags when configuring without (jonasschnelli)
- #20563 Check that Homebrew's berkeley-db4 package is actually installed (hebasto)
- #19493 Fix clang build on Mac (bvbfan)

Tests and QA

- #18593 Complete impl. of `msg_merkleblock` and `wait_for_merkleblock` (theStack)
- #18609 Remove REJECT message code (hebasto)
- #18584 Check that the version message does not leak the local address (MarcoFalke)
- #18597 Extend `wallet_dump` test to cover comments (MarcoFalke)

- #18596 Try once more when RPC connection fails on Windows (MarcoFalke)
- #18451 shift coverage from getunconfirmedbalance to getbalances (jonatack)
- #18631 appveyor: Disable functional tests for now (MarcoFalke)
- #18628 Add various low-level p2p tests (MarcoFalke)
- #18615 Avoid accessing free'd memory in `validation_chainstatemanager_tests` (MarcoFalke)
- #18571 fuzz: Disable debug log file (MarcoFalke)
- #18653 add coverage for bitcoin-cli -rpcwait (jonatack)
- #18660 Verify findCommonAncestor always initializes outputs (ryanofsky)
- #17669 Have coins simulation test also use CCoinsViewDB (jamesob)
- #18662 Replace gArgs with local argsman in bench (MarcoFalke)
- #18641 Create cached blocks not in the future (MarcoFalke)
- #18682 fuzz: `http_request` workaround for libevent < 2.1.1 (theStack)
- #18692 Bump timeout in `wallet_import_rescan` (MarcoFalke)
- #18695 Replace boost::mutex with std::mutex (hebasto)
- #18633 Properly raise FailedToStartError when rpc shutdown before warmup finished (MarcoFalke)
- #18675 Don't initialize PrecomputedTransactionData in txvalidationcache tests (jnewbery)
- #18691 Add `wait_for_cookie_credentials()` to framework for rpcwait tests (jonatack)
- #18672 Add further BIP37 size limit checks to `p2p_filter.py` (theStack)
- #18721 Fix linter issue (hebasto)
- #18384 More specific `feature_segwit` test error messages and fixing incorrect comments (gzhao408)
- #18575 bench: Remove requirement that all benches use same testing setup (MarcoFalke)
- #18690 Check object hashes in `wait_for_getdata` (robot-visions)
- #18712 display command line options passed to `send_cli()` in debug log (jonatack)
- #18745 Check submitblock return values (MarcoFalke)
- #18756 Use `wait_for_getdata()` in `p2p_compactblocks.py` (theStack)
- #18724 Add coverage for -rpcwallet cli option (jonatack)
- #18754 bench: Add caddrman benchmarks (vasild)
- #18585 Use zero-argument super() shortcut (Python 3.0+) (theStack)
- #18688 fuzz: Run in parallel (MarcoFalke)
- #18770 Remove raw-tx byte juggling in `mempool_reorg` (MarcoFalke)
- #18805 Add missing `sync_all` to `wallet_importdescriptors.py` (achow101)
- #18759 bench: Start nodes with -nodebuglogfile (MarcoFalke)
- #18774 Added test for upgradewallet RPC (brakmic)
- #18485 Add `mempool_updatefromblock.py` (hebasto)
- #18727 Add CreateWalletFromFile test (ryanofsky)
- #18726 Check misbehavior more independently in `p2p_filter.py` (robot-visions)
- #18825 Fix message for `ECC_InitSanityCheck` test (fanquake)
- #18576 Use unittest for `test_framework` unit testing (gzhao408)
- #18828 Strip down previous releases boilerplate (MarcoFalke)
- #18617 Add factor option to adjust test timeouts (brakmic)
- #18855 `feature_backwards_compatibility.py` test downgrade after upgrade (achow101)
- #18864 Add v0.16.3 backwards compatibility test, bump v0.19.0.1 to v0.19.1 (Sjors)
- #18917 fuzz: Fix vector size problem in system fuzzer (brakmic)
- #18901 fuzz: use std::optional for `sep_pos_opt` variable (brakmic)
- #18888 Remove RPCOverloadWrapper boilerplate (MarcoFalke)
- #18952 Avoid os-dependent path (fametrano)
- #18938 Fill fuzzing coverage gaps for functions in consensus/validation.h, primitives/block.h and util/translation.h (practicalswift)
- #18986 Add capability to disable RPC timeout in functional tests (rajarshimaitra)

- #18530 Add test for -blocksonly and -whitelistforcerelay param interaction (glowang)
- #19014 Replace `TEST_PREVIOUS_RELEASES` env var with `test_framework` option (MarcoFalke)
- #19052 Don't limit fuzzing inputs to 1 MB for afl-fuzz (now: ∞ ∀ fuzzers) (practicalswift)
- #19060 Remove global `wait_until` from `p2p_getdata` (MarcoFalke)
- #18926 Pass ArgsManager into `getarg_tests` (glowang)
- #19110 Explain that a bug should be filed when the tests fail (MarcoFalke)
- #18965 Implement `base58_decode` (10xcryptodev)
- #16564 Always define the `raii_event_tests` test suite (candrews)
- #19122 Add missing `sync_blocks` to `wallet_hd` (MarcoFalke)
- #18875 fuzz: Stop nodes in `process_message*` fuzzers (MarcoFalke)
- #18974 Check that invalid witness destinations can not be imported (MarcoFalke)
- #18210 Type hints in Python tests (kiminuo)
- #19159 Make valgrind.supp work on aarch64 (MarcoFalke)
- #19082 Moved the CScriptNum asserts into the unit test in script.py (gillichu)
- #19172 Do not swallow flake8 exit code (hebasto)
- #19188 Avoid overwriting the NodeContext member of the testing setup [-Wshadow-field] (MarcoFalke)
- #18890 `disconnect_nodes` should warn if nodes were already disconnected (robot-visions)
- #19227 change blacklist to blocklist (TrentZ)
- #19230 Move base58 to own module to break circular dependency (sipa)
- #19083 `msg_mempool`, `fRelay`, and other bloomfilter tests (gzhao408)
- #16756 Connection eviction logic tests (mzumsande)
- #19177 Fix and clean `p2p_invalid_messages` functional tests (troygiorshev)
- #19264 Don't import asyncio to test magic bytes (jnewbery)
- #19178 Make `mininode_lock` non-reentrant (jnewbery)
- #19153 Mempool compatibility test (S3RK)
- #18434 Add a test-security target and run it in CI (fanquake)
- #19252 Wait for disconnect in `disconnect_p2ps` + bloomfilter test followups (gzhao408)
- #19298 Add missing `sync_blocks` (MarcoFalke)
- #19304 Check that message sends successfully when header is split across two buffers (troygiorshev)
- #19208 move `sync_blocks` and `sync_mempool` functions to `test_framework.py` (ycshao)
- #19198 Check that peers with forcerelay permission are not asked to feefilter (MarcoFalke)
- #19351 add two edge case tests for CSubNet (vasild)
- #19272 net, test: invalid p2p messages and test framework improvements (jonatack)
- #19348 Bump linter versions (duncandean)
- #19366 Provide main(...) function in fuzzer. Allow building uninstrumented harnesses with --enable-fuzz (practicalswift)
- #19412 move `TEST_RUNNER_EXTRA` into native tsan setup (fanquake)
- #19368 Improve functional tests compatibility with BSD/macOS (S3RK)
- #19028 Set -logthreadnames in unit tests (MarcoFalke)
- #18649 Add std::locale::global to list of locale dependent functions (practicalswift)
- #19140 Avoid fuzzer-specific nullptr dereference in libevent when handling PROXY requests (practicalswift)
- #19214 Auto-detect SHA256 implementation in benchmarks (sipa)
- #19353 Fix mistakenly swapped "previous" and "current" lock orders (hebasto)
- #19533 Remove unnecessary `cs_mains` in `denialofservice_tests` (jnewbery)
- #19423 add functional test for txrelay during and after IBD (gzhao408)
- #16878 Fix non-deterministic coverage of test `DoS_mapOrphans` (davereikher)
- #19548 fuzz: add missing overrides to `signature_checker` (jonatack)
- #19562 Fix fuzzer compilation on macOS (freenancial)
- #19370 Static asserts for consistency of fee defaults (domob1812)

- #19599 clean `message_count` and `last_message` (troygiorshev)
- #19597 test decodepsbt fee calculation (count input value only once per UTXO) (theStack)
- #18011 Replace current benchmarking framework with nanobench (martinus)
- #19489 Fail `wait_until` early if connection is lost (MarcoFalke)
- #19340 Preserve the `LockData` initial state if "potential deadlock detected" exception thrown (hebasto)
- #19632 Catch decimal.InvalidOperation from `TestNodeCLI#send_cli` (Empact)
- #19098 Remove duplicate NodeContext hacks (ryanofsky)
- #19649 Restore test case for p2p transaction blinding (instagibbs)
- #19657 Wait until `is_connected` in `add_p2p_connection` (MarcoFalke)
- #19631 Wait for 'cmpctblock' in `p2p_compactblocks` when it is expected (Empact)
- #19674 use throwaway _ variable for unused loop counters (theStack)
- #19709 Fix 'make cov' with clang (hebasto)
- #19564 `p2p_feefilter` improvements (logging, refactoring, speedup) (theStack)
- #19756 add `sync_all` to fix race condition in wallet groups test (kallewoof)
- #19727 Removing unused classes from `p2p_leak.py` (dhruv)
- #19722 Add test for getblockheader verboseness (torhte)
- #19659 Add a seed corpus generation option to the fuzzing `test_runner` (darosior)
- #19775 Activate segwit in TestChain100Setup (MarcoFalke)
- #19760 Remove confusing mininode terminology (jnewbery)
- #19752 Update `wait_until` usage in tests not to use the one from utils (slmtpz)
- #19839 Set appveyor VM version to previous Visual Studio 2019 release (sipsorcery)
- #19830 Add tsan supp for leveldb::DBImpl::DeleteObsoleteFiles (MarcoFalke)
- #19710 bench: Prevent thread oversubscription and decreases the variance of result values (hebasto)
- #19842 Update the vcpkg checkout commit ID in appveyor config (sipsorcery)
- #19507 Expand functional zmq transaction tests (instagibbs)
- #19816 Rename wait until helper to `wait_until_helper` (MarcoFalke)
- #19859 Fixes failing functional test by changing version (n-thumann)
- #19887 Fix flaky `wallet_basic` test (fjahr)
- #19897 Change `FILE_CHAR_BLOCKLIST` to `FILE_CHARS_DISALLOWED` (verretor)
- #19800 Mockwallet (MarcoFalke)
- #19922 Run `rpc_txoutproof.py` even with wallet disabled (MarcoFalke)
- #19936 batch rpc with params (instagibbs)
- #19971 create default wallet in extended tests (Sjors)
- #19781 add parameterized constructor for `msg_sendcmpct()` (theStack)
- #19963 Clarify blocksonly whitelistforcerelay test (t-bast)
- #20022 Use explicit p2p objects where available (guggero)
- #20028 Check that invalid peer traffic is accounted for (MarcoFalke)
- #20004 Add signet witness commitment section parse tests (MarcoFalke)
- #20034 Get rid of default wallet hacks (ryanofsky)
- #20069 Mention commit id in scripted diff error (laanwj)
- #19947 Cover `change_type` option of "walletcreatefundedpsbt" RPC (guggero)
- #20126 `p2p_leak_tx.py` improvements (use MiniWallet, add `p2p_lock` acquires) (theStack)
- #20129 Don't export `in6addr_loopback` (vasild)
- #20131 Remove unused nVersion=1 in p2p tests (MarcoFalke)
- #20161 Minor Taproot follow-ups (sipa)
- #19401 Use GBT to get block versions correct (luke-jr)
- #20159 `mining_getblocktemplate_longpoll.py` improvements (use MiniWallet, add logging) (theStack)
- #20039 Convert amounts from float to decimal (prayank23)

- #20112 Speed up `wallet_resendwallettransactions` with mockscheduler RPC (MarcoFalke)
- #20247 fuzz: Check for addrv1 compatibility before using addrv1 serializer. Fuzz addrv2 serialization (practicalswift)
- #20167 Add test for -blockversion (MarcoFalke)
- #19877 Clarify `rpc_net` & `p2p_disconnect_ban functional` tests (amitiuttarwar)
- #20258 Remove getnettotals/getpeerinfo consistency test (jnewbery)
- #20242 fuzz: Properly initialize PrecomputedTransactionData (MarcoFalke)
- #20262 Skip --descriptor tests if sqlite is not compiled (achow101)
- #18788 Update more tests to work with descriptor wallets (achow101)
- #20289 fuzz: Check for addrv1 compatibility before using addrv1 serializer/deserializer on CService (practicalswift)
- #20290 fuzz: Fix DecodeHexTx fuzzing harness issue (practicalswift)
- #20245 Run `script_assets_test` even if built --with-libs=no (MarcoFalke)
- #20300 fuzz: Add missing `ECC_Start` to `descriptor_parse` test (S3RK)
- #20283 Only try witness deser when checking for witness deser failure (MarcoFalke)
- #20303 fuzz: Assert expected DecodeHexTx behaviour when using legacy decoding (practicalswift)
- #20316 Fix `wallet_multiwallet` test issue on Windows (MarcoFalke)
- #20326 Fix `ecdsa_verify` in test framework (stepansnigirev)
- #20328 cirrus: Skip tasks on the gui repo main branch (MarcoFalke)
- #20355 fuzz: Check for addrv1 compatibility before using addrv1 serializer/deserializer on CSubNet (practicalswift)
- #20332 Mock IBD in `net_processing` fuzzers (MarcoFalke)
- #20218 Suppress `epoll_ctl` data race (MarcoFalke)
- #20375 fuzz: Improve coverage for CPartialMerkleTree fuzzing harness (practicalswift)
- #19669 contrib: Fixup valgrind suppressions file (MarcoFalke)
- #18879 valgrind: remove outdated suppressions (fanquake)
- #19226 Add BerkeleyDatabase tsan suppression (MarcoFalke)
- #20379 Remove no longer needed UBSan suppression (float divide-by-zero in validation.cpp) (practicalswift)
- #18190, #18736, #18744, #18775, #18783, #18867, #18994, #19065, #19067, #19143, #19222, #19247, #19286, #19296, #19379, #19934, #20188, #20395 Add fuzzing harnessses (practicalswift)
- #18638 Use mockable time for ping/pong, add tests (MarcoFalke)
- #19951 CNetAddr scoped ipv6 test coverage, rename scopeId to `m_scope_id` (jonatack)
- #20027 Use mockable time everywhere in `net_processing` (sipa)
- #19105 Add Muhash3072 implementation in Python (fjahr)
- #18704, #18752, #18753, #18765, #18839, #18866, #18873, #19022, #19023, #19429, #19552, #19778, #20176, #20179, #20214, #20292, #20299, #20322 Fix intermittent test issues (MarcoFalke)
- #20390 CI/Cirrus: Skip `merge_base` step for non-PRs (luke-jr)
- #18634 ci: Add fuzzbuzz integration configuration file (practicalswift)
- #18591 Add C++17 build to Travis (sipa)
- #18581, #18667, #18798, #19495, #19519, #19538 CI improvements (hebasto)
- #18683, #18705, #18735, #18778, #18799, #18829, #18912, #18929, #19008, #19041, #19164, #19201, #19267, #19276, #19321, #19371, #19427, #19730, #19746, #19881, #20294, #20339, #20368 CI improvements (MarcoFalke)
- #20489, #20506 MSVC CI improvements (sipsorcery)

Miscellaneous

- #18713 scripts: Add macho stack canary check to security-check.py (fanquake)
- #18629 scripts: Add pe .reloc section check to security-check.py (fanquake)

- #18437 util: `Detect posix_fallocate()` instead of assuming (vasild)
- #18413 script: Prevent ub when computing abs value for num opcode serialize (pierreN)
- #18443 lockedpool: avoid sensitive data in core files (FreeBSD) (vasild)
- #18885 contrib: Move optimize-pngs.py script to the maintainer repo (MarcoFalke)
- #18317 Serialization improvements step 6 (all except wallet/gui) (sipa)
- #16127 More thread safety annotation coverage (ajtowns)
- #19228 Update libsecp256k1 subtree (sipa)
- #19277 util: Add assert identity function (MarcoFalke)
- #19491 util: Make assert work with any value (MarcoFalke)
- #19205 script: `previous_release.sh` rewritten in python (bliotti)
- #15935 Add /settings.json persistent settings storage (ryanofsky)
- #19439 script: Linter to check commit message formatting (Ghorbanian)
- #19654 lint: Improve commit message linter in travis (fjahr)
- #15382 util: Add runcommandparsejson (Sjors)
- #19614 util: Use `have_fdatasync` to determine fdatasync() use (fanquake)
- #19813 util, ci: Hard code previous release tarball checksums (hebasto)
- #19841 Implement Keccak and `SHA3_256` (sipa)
- #19643 Add -netinfo peer connections dashboard (jonatack)
- #15367 feature: Added ability for users to add a startup command (benthecarman)
- #19984 log: Remove static log message "Initializing chainstate Chainstate [ibd] @ height -1 (null)" (practicalswift)
- #20092 util: Do not use gargs global in argsmanager member functions (hebasto)
- #20168 contrib: Fix `gen_key_io_test_vectors.py` imports (MarcoFalke)
- #19624 Warn on unknown `rw_settings` (MarcoFalke)
- #20257 Update secp256k1 subtree to latest master (sipa)
- #20346 script: Modify security-check.py to use "==" instead of "is" for literal comparison (tylerchambers)
- #18881 Prevent UB in DeleteLock() function (hebasto)
- #19180, #19189, #19190, #19220, #19399 Replace RecursiveMutex with Mutex (hebasto)
- #19347 Make `cs_inventory` nonrecursive (jnewbery)
- #19773 Avoid recursive lock in IsTrusted (promag)
- #18790 Improve thread naming (hebasto)
- #20140 Restore compatibility with old CSubNet serialization (sipa)
- #17775 DecodeHexTx: Try case where txn has inputs first (instagibbs)

Documentation

- #18502 Update docs for getbalance (default minconf should be 0) (uzyn)
- #18632 Fix macos comments in release-notes (MarcoFalke)
- #18645 Update thread information in developer docs (jnewbery)
- #18709 Note why we can't use `thread_local` with glibc back compat (fanquake)
- #18410 Improve commenting for coins.cpp|h (jnewbery)
- #18157 fixing init.md documentation to not require rpcpassword (jkcd)
- #18739 Document how to fuzz Bitcoin Core using Honggfuzz (practicalswift)
- #18779 Better explain GNU ld's dislike of ld64's options (fanquake)
- #18663 Mention build docs in README.md (saahilshangle)
- #18810 Update rest info on block size and json (chrisabrams)
- #18939 Add c++17-enable flag to fuzzing instructions (mzumsande)
- #18957 Add a link from ZMQ doc to ZMQ example in contrib/ (meeDamian)
- #19058 Drop protobuf stuff (hebasto)
- #19061 Add link to Visual Studio build readme (maitrebitcoin)
- #19072 Expand section on Getting Started (MarcoFalke)

- #18968 noban precludes maxuploadtarget disconnects (MarcoFalke)
- #19005 Add documentation for 'checklevel' argument in 'verifychain' RPC… (kcalvinalvin)
- #19192 Extract net permissions doc (MarcoFalke)
- #19071 Separate repository for the gui (MarcoFalke)
- #19018 fixing description of the field sequence in walletcreatefundedpsbt RPC method (limpbrains)
- #19367 Span pitfalls (sipa)
- #19408 Windows WSL build recommendation to temporarily disable Win32 PE support (sipsorcery)
- #19407 explain why passing -mlinker-version is required when cross-compiling (fanquake)
- #19452 afl fuzzing comment about afl-gcc and afl-g++ (Crypt-iQ)
- #19258 improve subtree check instructions (Sjors)
- #19474 Use precise permission flags where possible (MarcoFalke)
- #19494 CONTRIBUTING.md improvements (jonatack)
- #19268 Add non-thread-safe note to FeeFilterRounder::round() (hebasto)
- #19547 Update macOS cross compilation dependencies for Focal (hebasto)
- #19617 Clang 8 or later is required with `FORCE_USE_SYSTEM_CLANG` (fanquake)
- #19639 Remove Reference Links #19582 (RobertHosking)
- #19605 Set `CC_FOR_BUILD` when building on OpenBSD (fanquake)
- #19765 Fix getmempoolancestors RPC result doc (MarcoFalke)
- #19786 Remove label from good first issue template (MarcoFalke)
- #19646 Updated outdated help command for getblocktemplate (jakeleventhal)
- #18817 Document differences in bitcoind and bitcoin-qt locale handling (practicalswift)
- #19870 update PyZMQ install instructions, fix `zmq_sub.py` file permissions (jonatack)
- #19903 Update build-openbsd.md with GUI support (grubles)
- #19241 help: Generate checkpoint height from chainparams (luke-jr)
- #18949 Add CODEOWNERS file to automatically nominate PR reviewers (adamjonas)
- #20014 Mention signet in -help output (hebasto)
- #20015 Added default signet config for linearize script (gr0kchain)
- #19958 Better document features of feelers (naumenkogs)
- #19871 Clarify scope of eviction protection of outbound block-relay peers (ariard)
- #20076 Update and improve files.md (hebasto)
- #20107 Collect release-notes snippets (MarcoFalke)
- #20109 Release notes and followups from 19339 (glozow)
- #20090 Tiny followups to new getpeerinfo connection type field (amitiuttarwar)
- #20152 Update wallet files in files.md (hebasto)
- #19124 Document `ALLOW_HOST_PACKAGES` dependency option (skmcontrib)
- #20271 Document that wallet salvage is experimental (MarcoFalke)
- #20281 Correct getblockstats documentation for `(sw)total_weight` (shesek)
- #20279 release process updates/fixups (jonatack)
- #20238 Missing comments for signet parameters (decryp2kanon)
- #20756 Add missing field (permissions) to the getpeerinfo help (amitiuttarwar)
- #20668 warn that incoming conns are unlikely when not using default ports (adamjonas)
- #19961 tor.md updates (jonatack)
- #19050 Add warning for rest interface limitation (fjahr)
- #19390 doc/REST-interface: Remove stale info (luke-jr)
- #19344 docs: update testgen usage example (Bushstar)

Credits

Thanks to everyone who directly contributed to this release:

- 10xcryptodev
- Aaron Clauson
- Aaron Hook
- Adam Jonas
- Adam Soltys
- Adam Stein
- Akio Nakamura
- Alex Willmer
- Amir Ghorbanian
- Amiti Uttarwar
- Andrew Chow
- Andrew Toth
- Anthony Fieroni
- Anthony Towns
- Antoine Poinsot
- Antoine Riard
- Ben Carman
- Ben Woosley
- Benoit Verret
- Brian Liotti
- Bushstar
- Calvin Kim
- Carl Dong
- Chris Abrams
- Chris L
- Christopher Coverdale
- codeShark149
- Cory Fields
- Craig Andrews
- Damian Mee
- Daniel Kraft
- Danny Lee
- David Reikher
- DesWurstes
- Dhruv Mehta
- Duncan Dean
- Elichai Turkel
- Elliott Jin
- Emil Engler
- Ethan Heilman
- eugene
- Fabian Jahr
- fanquake
- Ferdinando M. Ametrano
- freenancial
- furszy
- Gillian Chu
- Gleb Naumenko
- Glenn Willen
- Gloria Zhao

- glowang
- gr0kchain
- Gregory Sanders
- grubles
- gzhao408
- Harris
- Hennadii Stepanov
- Hugo Nguyen
- Igor Cota
- Ivan Metlushko
- Ivan Vershigora
- Jake Leventhal
- James O'Beirne
- Jeremy Rubin
- jgmorgan
- Jim Posen
- ”jkcd”
- jmorgan
- John Newbery
- Johnson Lau
- Jon Atack
- Jonas Schnelli
- Jonathan Schoeller
- João Barbosa
- Justin Moon
- kanon
- Karl-Johan Alm
- Kiminuo
- Kristaps Kaupe
- lontivero
- Luke Dashjr
- Marcin Jachymiak
- MarcoFalke
- Martin Ankerl
- Martin Zumsande
- maskoficarus
- Matt Corallo
- Matthew Zipkin
- MeshCollider
- Miguel Herranz
- MIZUTA Takeshi
- mruddy
- Nadav Ivgi
- Neha Narula
- Nicolas Thumann
- Niklas Gögge
- Nima Yazdanmehr
- nsa
- nthumann
- Oliver Gugger

- pad
- pasta
- Peter Bushnell
- pierrenn
- Pieter Wuille
- practicalswift
- Prayank
- Raúl Martínez (RME)
- RandyMcMillan
- Rene Pickhardt
- Riccardo Masutti
- Robert
- Rod Vagg
- Roy Shao
- Russell Yanofsky
- Saahil Shangle
- sachinkm77
- saibato
- Samuel Dobson
- sanket1729
- Sebastian Falbesoner
- Seleme Topuz
- Sishir Giri
- Sjors Provoost
- skmcontrib
- Stepan Snigirev
- Stephan Oeste
- Suhas Daftuar
- t-bast
- Tom Harding
- Torhte Butler
- TrentZ
- Troy Giorshev
- tryphe
- Tyler Chambers
- U-Zyn Chua
- Vasil Dimov
- wiz
- Wladimir J. van der Laan

As well as to everyone that helped with translations on Transifex.