

Bitcoin Core version 0.17.0 is now available from:

<https://bitcoincore.org/bin/bitcoin-core-0.17.0/>

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

Please report bugs using the issue tracker at GitHub:

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

To receive security and update notifications, please subscribe to:

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

How to Upgrade

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

If your node has a txindex, the txindex db will be migrated the first time you run 0.17.0 or newer, which may take up to a few hours. Your node will not be functional until this migration completes.

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

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

Downgrading warning

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

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

Compatibility

Bitcoin Core is extensively tested on multiple operating systems using the Linux kernel, macOS 10.10+, and Windows 7 and newer (Windows XP is not supported).

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

From 0.17.0 onwards macOS <10.10 is no longer supported. 0.17.0 is built using Qt 5.9.x, which doesn't support versions of macOS older than 10.10.

Known issues

- Upgrading from 0.13.0 or older currently results in memory blow-up during the roll-back of blocks to the SegWit activation point. In these cases, a full `-reindex` is necessary.
- The GUI suffers from visual glitches in the new MacOS dark mode. This has to do with our Qt theme handling and is not a new problem in 0.17.0, but is expected to be resolved in 0.17.1.

Notable changes

Changed configuration options

- `-includeconf=<file>` can be used to include additional configuration files. Only works inside the `bitcoin.conf` file, not inside included files or from command-line. Multiple files may be included. Can be disabled from command-line via `-noincludeconf`. Note that multi-argument commands like `-includeconf` will override preceding `-noincludeconf`, i.e.

```
noincludeconf=1
includeconf=relative.conf
```

as `bitcoin.conf` will still include `relative.conf`.

GUI changes

- Block storage can be limited under Preferences, in the Main tab. Undoing this setting requires downloading the full blockchain again. This mode is incompatible with `-txindex` and `-rescan`.

External wallet files

The `-wallet=<path>` option now accepts full paths instead of requiring wallets to be located in the `-walletdir` directory.

Newly created wallet format

If `-wallet=<path>` is specified with a path that does not exist, it will now create a wallet directory at the specified location (containing a `wallet.dat` data file, a `db.log` file, and `database/log.??????????` files) instead of just creating a data

file at the path and storing log files in the parent directory. This should make backing up wallets more straightforward than before because the specified wallet path can just be directly archived without having to look in the parent directory for transaction log files.

For backwards compatibility, wallet paths that are names of existing data files in the `-walletdir` directory will continue to be accepted and interpreted the same as before.

Dynamic loading and creation of wallets

Previously, wallets could only be loaded or created at startup, by specifying `-wallet` parameters on the command line or in the `bitcoin.conf` file. It is now possible to load, create and unload wallets dynamically at runtime:

- Existing wallets can be loaded by calling the `loadwallet` RPC. The wallet can be specified as file/directory basename (which must be located in the `walletdir` directory), or as an absolute path to a file/directory.
- New wallets can be created (and loaded) by calling the `createwallet` RPC. The provided name must not match a wallet file in the `walletdir` directory or the name of a wallet that is currently loaded.
- Loaded wallets can be unloaded by calling the `unloadwallet` RPC.

This feature is currently only available through the RPC interface.

Coin selection

Partial spend avoidance

When an address is paid multiple times the coins from those separate payments can be spent separately which hurts privacy due to linking otherwise separate addresses. A new `-avoidpartialspends` flag has been added (default=false). If enabled, the wallet will always spend existing UTXO to the same address together even if it results in higher fees. If someone were to send coins to an address after it was used, those coins will still be included in future coin selections.

Configuration sections for testnet and regtest

It is now possible for a single configuration file to set different options for different networks. This is done by using sections or by prefixing the option with the network, such as:

```
main.uacomment=bitcoin
test.uacomment=bitcoin-testnet
regtest.uacomment=regtest
[main]
mempoolsize=300
```

```
[test]
mempoolsize=100
[regtest]
mempoolsize=20
```

If the following options are not in a section, they will only apply to mainnet: `addnode=`, `connect=`, `port=`, `bind=`, `rpcport=`, `rpcbind=` and `wallet=`. The options to choose a network (`regtest=` and `testnet=`) must be specified outside of sections.

‘label’ and ‘account’ APIs for wallet

A new ‘label’ API has been introduced for the wallet. This is intended as a replacement for the deprecated ‘account’ API. The ‘account’ can continue to be used in V0.17 by starting bitcoind with the ‘-deprecatedrpc=accounts’ argument, and will be fully removed in V0.18.

The label RPC methods mirror the account functionality, with the following functional differences:

- Labels can be set on any address, not just receiving addresses. This functionality was previously only available through the GUI.
- Labels can be deleted by reassigning all addresses using the `setlabel` RPC method.
- There isn’t support for sending transactions *from* a label, or for determining which label a transaction was sent from.
- Labels do not have a balance.

Here are the changes to RPC methods:

Deprecated Method	New Method	Notes
<code>getaccount</code>	<code>getaddressinfo</code>	<code>getaddressinfo</code> returns a json object with address information instead of just the name of the account as a string.
<code>getaccountaddress</code>	n/a	There is no replacement for <code>getaccountaddress</code> since labels do not have an associated receive address.

Deprecated Method	New Method	Notes
<code>getaddressesbyaccount</code>	<code>getaddressesbylabel</code>	<code>getaddressesbylabel</code> returns a json object with the addresses as keys, instead of a list of strings.
<code>getreceivedbyaccount</code>	<code>getreceivedbylabel</code>	<i>no change in behavior</i>
<code>listaccounts</code>	<code>listlabels</code>	<code>listlabels</code> does not return a balance or accept <code>minconf</code> and <code>watchonly</code> arguments.
<code>listreceivedbyaccount</code>	<code>listreceivedbylabel</code>	Both methods return new <code>label</code> fields, along with <code>account</code> fields for backward compatibility.
<code>move</code>	n/a	<i>no replacement</i>
<code>sendfrom</code>	n/a	<i>no replacement</i>
<code>setaccount</code>	<code>setlabel</code>	Both methods now:

Changed Method	Notes
<code>addmultisigaddress</code>	Renamed <code>account</code> named parameter to <code>label</code> . Still accepts <code>account</code> for backward compatibility if running with ‘-deprecatedrpc=accounts’.

Changed Method	Notes
<code>getnewaddress</code>	Renamed <code>account</code> named parameter to <code>label</code> . Still accepts <code>account</code> for backward compatibility. if running with ‘-deprecatedrpc=accounts’
<code>listunspent</code>	Returns new <code>label</code> fields. <code>account</code> field will be returned for backward compatibility if running with ‘-deprecatedrpc=accounts’
<code>sendmany</code>	The <code>account</code> named parameter has been renamed to <code>dummy</code> . If provided, the <code>dummy</code> parameter must be set to the empty string, unless running with the <code>-deprecatedrpc=accounts</code> argument (in which case functionality is unchanged).

Changed Method	Notes
<code>listtransactions</code>	The <code>account</code> named parameter has been renamed to <code>dummy</code> . If provided, the <code>dummy</code> parameter must be set to the string <code>*</code> , unless running with the <code>-deprecatedrpc=accounts</code> argument (in which case functionality is unchanged).
<code>getbalance</code>	<code>account</code> , <code>minconf</code> and <code>include_watchonly</code> parameters are deprecated, and can only be used if running with <code>'-deprecatedrpc=accounts'</code>

BIP 174 Partially Signed Bitcoin Transactions support

BIP 174 PSBT is an interchange format for Bitcoin transactions that are not fully signed yet, together with relevant metadata to help entities work towards signing it. It is intended to simplify workflows where multiple parties need to cooperate to produce a transaction. Examples include hardware wallets, multisig setups, and CoinJoin transactions.

Overall workflow

Overall, the construction of a fully signed Bitcoin transaction goes through the following steps:

- A **Creator** proposes a particular transaction to be created. He constructs a PSBT that contains certain inputs and outputs, but no additional metadata.
- For each input, an **Updater** adds information about the UTXOs being spent by the transaction to the PSBT.
- A potentially other Updater adds information about the scripts and public keys involved in each of the inputs (and possibly outputs) of the PSBT.
- **Signers** inspect the transaction and its metadata to decide whether they

agree with the transaction. They can use amount information from the UTXOs to assess the values and fees involved. If they agree, they produce a partial signature for the inputs for which they have relevant key(s).

- A **Finalizer** is run for each input to convert the partial signatures and possibly script information into a final `scriptSig` and/or `scriptWitness`.
- An **Extractor** produces a valid Bitcoin transaction (in network format) from a PSBT for which all inputs are finalized.

Generally, each of the above (excluding Creator and Extractor) will simply add more and more data to a particular PSBT. In a naive workflow, they all have to operate sequentially, passing the PSBT from one to the next, until the Extractor can convert it to a real transaction. In order to permit parallel operation, **Combiners** can be employed which merge metadata from different PSBTs for the same unsigned transaction.

The names above in bold are the names of the roles defined in BIP174. They're useful in understanding the underlying steps, but in practice, software and hardware implementations will typically implement multiple roles simultaneously.

RPCs

- `converttopsbt` (**Creator**) is a utility RPC that converts an unsigned raw transaction to PSBT format. It ignores existing signatures.
- `createpsbt` (**Creator**) is a utility RPC that takes a list of inputs and outputs and converts them to a PSBT with no additional information. It is equivalent to calling `createrawtransaction` followed by `converttopsbt`.
- `walletcreatefundedpsbt` (**Creator, Updater**) is a wallet RPC that creates a PSBT with the specified inputs and outputs, adds additional inputs and change to it to balance it out, and adds relevant metadata. In particular, for inputs that the wallet knows about (counting towards its normal or watch-only balance), UTXO information will be added. For outputs and inputs with UTXO information present, key and script information will be added which the wallet knows about. It is equivalent to running `createrawtransaction`, followed by `fundrawtransaction`, and `converttopsbt`.
- `walletprocesspsbt` (**Updater, Signer, Finalizer**) is a wallet RPC that takes as input a PSBT, adds UTXO, key, and script data to inputs and outputs that miss it, and optionally signs inputs. Where possible it also finalizes the partial signatures.
- `finalizepsbt` (**Finalizer, Extractor**) is a utility RPC that finalizes any partial signatures, and if all inputs are finalized, converts the result to a fully signed transaction which can be broadcast with `sendrawtransaction`.
- `combinepsbt` (**Combiner**) is a utility RPC that implements a Combiner. It can be used at any point in the workflow to merge information added to different versions of the same PSBT. In particular it is useful to combine the output of multiple Updaters or Signers.
- `decodepsbt` is a diagnostic utility RPC which will show all information

in a PSBT in human-readable form, as well as compute its eventual fee if known.

Upgrading non-HD wallets to HD wallets

Since Bitcoin Core 0.13.0, creating new BIP 32 Hierarchical Deterministic wallets has been supported by Bitcoin Core but old non-HD wallets could not be upgraded to HD. Now non-HD wallets can be upgraded to HD using the `-upgradewallet` command line option. This upgrade will result in the all keys in the keypool being marked as used and a new keypool generated. **A new backup must be made when this upgrade is performed.**

Additionally, `-upgradewallet` can be used to upgrade from a non-split HD chain (all keys generated with `m/0'/0'/i'`) to a split HD chain (receiving keys generated with `'m/0'/0'/i'` and change keys generated with `m'/0'/1'/i'`). When this upgrade occurs, all keys already in the keypool will remain in the keypool to be used until all keys from before the upgrade are exhausted. This is to avoid issues with backups and downgrades when some keys may come from the change key keypool. Users can begin using the new split HD chain keypools by using the `newkeypool` RPC to mark all keys in the keypool as used and begin using a new keypool generated from the split HD chain.

HD Master key rotation

A new RPC, `sethdseed`, has been introduced which allows users to set a new HD seed or set their own HD seed. This allows for a new HD seed to be used. **A new backup must be made when a new HD seed is set.**

Low-level RPC changes

- The new RPC `scantxoutset` can be used to scan the UTXO set for entries that match certain output descriptors. Refer to the output descriptors reference documentation for more details. This call is similar to `listunspent` but does not use a wallet, meaning that the wallet can be disabled at compile or run time. This call is experimental, as such, is subject to changes or removal in future releases.
- The `createrawtransaction` RPC will now accept an array or dictionary (kept for compatibility) for the `outputs` parameter. This means the order of transaction outputs can be specified by the client.
- The `fundrawtransaction` RPC will reject the previously deprecated `reserveChangeKey` option.
- `sendmany` now shuffles outputs to improve privacy, so any previously expected behavior with regards to output ordering can no longer be relied upon.

- The new RPC `testmempoolaccept` can be used to test acceptance of a transaction to the mempool without adding it.
- JSON transaction decomposition now includes a `weight` field which provides the transaction's exact weight. This is included in REST `/rest/tx/` and `/rest/block/` endpoints when in json mode. This is also included in `getblock` (with `verbosity=2`), `listsinceblock`, `listtransactions`, and `getrawtransaction` RPC commands.
- New `fees` field introduced in `getrawmempool`, `getmempoolancestors`, `getmempooldescendants` and `getmempoolentry` when `verbosity` is set to `true` with sub-fields `ancestor`, `base`, `modified` and `descendant` denominated in BTC. This new field deprecates previous fee fields, such as `fee`, `modifiedfee`, `ancestorfee` and `descendantfee`.
- The new RPC `getzmqnotifications` returns information about active ZMQ notifications.
- When bitcoin is not started with any `-wallet=<path>` options, the name of the default wallet returned by `getwalletinfo` and `listwallets` RPCs is now the empty string `"` instead of `"wallet.dat"`. If bitcoin is started with any `-wallet=<path>` options, there is no change in behavior, and the name of any wallet is just its `<path>` string.
- Passing an empty string `("")` as the `address_type` parameter to `getnewaddress`, `getrawchangeaddress`, `addmultisigaddress`, `fundrawtransaction` RPCs is now an error. Previously, this would fall back to using the default address type. It is still possible to pass null or leave the parameter unset to use the default address type.
- Bare multisig outputs to our keys are no longer automatically treated as incoming payments. As this feature was only available for multisig outputs for which you had all private keys in your wallet, there was generally no use for them compared to single-key schemes. Furthermore, no address format for such outputs is defined, and wallet software can't easily send to it. These outputs will no longer show up in `listtransactions`, `listunspent`, or contribute to your balance, unless they are explicitly watched (using `importaddress` or `importmulti` with hex script argument). `signrawtransaction*` also still works for them.
- The `getwalletinfo` RPC method now returns an `hdseedid` value, which is always the same as the incorrectly-named `hdmasterkeyid` value. `hdmasterkeyid` will be removed in V0.18.
- The `getaddressinfo` RPC method now returns an `hdseedid` value, which is always the same as the incorrectly-named `hdmasterkeyid` value. `hdmasterkeyid` will be removed in V0.18.
- Parts of the `validateaddress` RPC method have been deprecated and moved to `getaddressinfo`. Clients must transition to using

`getaddressinfo` to access this information before upgrading to v0.18. The following deprecated fields have moved to `getaddressinfo` and will only be shown with `-deprecatedrpc=validateaddress`: `ismine`, `iswatchonly`, `script`, `hex`, `pubkeys`, `sigsrequired`, `pubkey`, `addresses`, `embedded`, `iscompressed`, `account`, `timestamp`, `hdkeypath`, `hdmasterkeyid`.

- `signrawtransaction` is deprecated and will be fully removed in v0.18. To use `signrawtransaction` in v0.17, restart bitcoind with `-deprecatedrpc=signrawtransaction`. Projects should transition to using `signrawtransactionwithkey` and `signrawtransactionwithwallet` before upgrading to v0.18.

Other API changes

- The `inactivehdmaster` property in the `dumpwallet` output has been corrected to `inactivehdseed`

Logging

- The log timestamp format is now ISO 8601 (e.g. “2018-02-28T12:34:56Z”).
- When running bitcoind with `-debug` but without `-daemon`, logging to stdout is now the default behavior. Setting `-printtoconsole=1` no longer implicitly disables logging to `debug.log`. Instead, logging to file can be explicitly disabled by setting `-debuglogfile=0`.

Transaction index changes

The transaction index is now built separately from the main node procedure, meaning the `-txindex` flag can be toggled without a full reindex. If bitcoind is run with `-txindex` on a node that is already partially or fully synced without one, the transaction index will be built in the background and become available once caught up. When switching from running `-txindex` to running without the flag, the transaction index database will *not* be deleted automatically, meaning it could be turned back on at a later time without a full resync.

Miner block size removed

The `-blockmaxsize` option for miners to limit their blocks’ sizes was deprecated in V0.15.1, and has now been removed. Miners should use the `-blockmaxweight` option if they want to limit the weight of their blocks.

Python Support

Support for Python 2 has been discontinued for all test files and tools.

0.17.0 change log

Consensus

- #12204 3fa24bb Fix overly eager BIP30 bypass (morcos)

Policy

- #12568 ed6ae80 Allow dustrelayfee to be set to zero (luke-jr)
- #13120 ca2a233 Treat segwit as always active (MarcoFalke)
- #13096 062738c Fix MAX_STANDARD_TX_WEIGHT check (jl2012)

Mining

- #12693 df529dc Remove unused variable in SortForBlock (drewx2)
- #12448 84efa9a Interrupt block generation on shutdown request (promag)

Block and transaction handling

- #12225 67447ba Mempool cleanups (sdaftuar)
- #12356 fd65937 Fix ‘mempool min fee not met’ debug output (Empact)
- #12287 bf3353d Optimise lock behaviour for GuessVerificationProgress() (jonasschnelli)
- #11889 47a7666 Drop extra script variable in ProduceSignature (ryanofsky)
- #11880 d59b8d6 Stop special-casing phashBlock handling in validation for TBV (TheBlueMatt)
- #12431 947c25e Only call NotifyBlockTip when chainActive changes (jamesob)
- #12653 534b8fa Allow to optional specify the directory for the blocks storage (jonasschnelli)
- #12172 3b62a91 Bugfix: RPC: savemempool: Don’t save until LoadMempool() is finished (jtimon)
- #12167 88430cb Make segwit failure due to CLEANSTACK violation return a SCRIPT_ERR_CLEANSTACK error code (maaku)
- #12561 24133b1 Check for block corruption in ConnectBlock() (sdaftuar)
- #11617 1b5723e Avoid lock: Call FlushStateToDisk(...) regardless of fCheckForPruning (practicalswift)
- #11739 0a8b7b4 Enforce SCRIPT_VERIFY_P2SH and SCRIPT_VERIFY_WITNESS from genesis (sdaftuar)
- #12885 a49381d Reduce implementation code inside CScript (sipa)
- #13032 34dd1a6 Output values for “min relay fee not met” error (kristapsk)
- #13033 a07e8ca Build txindex in parallel with validation (jimpo)
- #13080 66cc47b Add compile time checking for ::mempool.cs runtime locking assertions (practicalswift)
- #13185 08c1caf Bugfix: the end of a reorged chain is invalid when connect fails (sipa)

- #11689 0264836 Fix missing locking in CTxMemPool::check(...) and CTxMemPool::setSanityCheck(...) (practicalswift)
- #13011 3c2a41a Cache witness hash in CTransaction (MarcoFalke)
- #13191 0de7cc8 Specialized double-SHA256 with 64 byte inputs with SSE4.1 and AVX2 (sipa)
- #13243 ea263e1 Make reusable base class for auxiliary indices (jimpo)
- #13393 a607d23 Enable double-SHA256-for-64-byte code on 32-bit x86 (sipa)
- #13428 caabdea validation: check the specified number of blocks (off-by-one) (kallewoof)
- #13438 450055b Improve coverage of SHA256 SelfTest code (sipa)
- #13431 954f4a9 validation: count blocks correctly for check level < 3 (kallewoof)
- #13386 3a3eabe SHA256 implementations based on Intel SHA Extensions (sipa)
- #11658 9a1ad2c During IBD, when doing pruning, prune 10% extra to avoid pruning again soon after (luke-jr)
- #13794 8ce55df chainparams: Update with data from assumed valid chain (MarcoFalke)
- #13527 e7ea858 Remove promiscuousmempoolflags (MarcoFalke)

P2P protocol and network code

- #12342 eaeaa2d Extend #11583 (“Do not make it trivial for inbound peers to generate log entries”) to include “version handshake timeout” message (clemtaylor)
- #12218 9a32114 Move misbehaving logging to net logging category (laanwj)
- #10387 5c2aff8 Eventually connect to NODE_NETWORK_LIMITED peers (jonasschnelli)
- #9037 a36834f Add test-before-evict discipline to addrman (EthanHeilman)
- #12622 e1d6e2a Correct addrman logging (laanwj)
- #11962 0a01843 add seed.bitcoin.sprovoost.nl to DNS seeds (Sjors)
- #12569 23e7fe8 Increase signal-to-noise ratio in debug.log by adjusting log level when logging failed non-manual connect():s (practicalswift)
- #12855 c199869 Minor accumulated cleanups (tjps)
- #13153 ef46c99 Add missing newlines to debug logging (laanwj)
- #13162 a174702 Don’t incorrectly log that REJECT messages are unknown (jnewbery)
- #13151 7f4db9a Serve blocks directly from disk when possible (laanwj)
- #13134 70d3541 Add option `-enablebip61` to configure sending of BIP61 notifications (laanwj)
- #13532 7209fec Log warning when deprecated network name ‘tor’ is used (wodry)
- #13615 172f984 Remove unused interrupt from SendMessages (fanquake)

- #13417 1e90862 Tighten scope in `net_processing` (skeees)
- #13298 f8d470e Bucketing INV delays (1 bucket) for incoming connections to hide tx time (naumenkogs)
- #13672 0d8d6be Modified `in_addr6` cast in `CConman` class to work with msvc (sipsorcery)
- #11637 c575260 Remove dead service bits code (MarcoFalke)
- #13212 a6f00ce Fixed a race condition when disabling the network (lmaners)
- #13656 1211b15 Remove the boost/algorithm/string/predicate.hpp dependency (251Labs)
- #13423 f58674a Thread safety annotations in `net_processing` (skeees)
- #13776 7d36237 Add missing verification of IPv6 address in `CNetAddr::GetIn6Addr(...)` (practicalswift)
- #13907 48bf8ff Introduce a maximum size for locators (gmaxwell)
- #13951 8a9ffec Hardcoded seeds update pre-0.17 branch (laanwj)

Wallet

- #12330 2a30e67 Reduce scope of `cs_main` and `cs_wallet` locks in list-transactions (promag)
- #12298 a1ffddb Refactor HaveKeys to early return on false result (promag)
- #12282 663911e Disallow abandon of conflicted txes (MarcoFalke)
- #12333 d405bee Make `CWallet::ListCoins` atomic (promag)
- #12296 8e6f9f4 Only fee-bump non-conflicted/non-confirmed txes (MarcoFalke)
- #11866 6bb9c13 Do not un-mark `fInMempool` on wallet txn if ATMP fails (TheBlueMatt)
- #11882 987a809 Disable default fallbackfee on mainnet (jonasschnelli)
- #9991 4ca7c1e listreceivedbyaddress Filter Address (NicolasDorier)
- #11687 98bc27f External wallet files (ryanofsky)
- #12658 af88094 Sanitize some wallet serialization (sipa)
- #9680 6acd870 Unify `CWalletTx` construction (ryanofsky)
- #10637 e057589 Coin Selection with Murch's algorithm (achow101, Xekyo)
- #12408 c39dd2e Change output type globals to members (MarcoFalke)
- #12694 9552dfb Actually disable BnB when there are preset inputs (achow101)
- #11536 cead84b Rename account to label where appropriate (ryanofsky)
- #12709 02b7e83 shuffle sendmany recipients ordering (instagibbs)
- #12699 c948dc8 Shuffle transaction inputs before signing (instagibbs)
- #10762 6d53663 Remove Wallet dependencies from init.cpp (jnewbery)
- #12857 821980c Avoid travis lint-include-guards error (ken2812221)
- #12702 dab0d68 importprivkey: hint about importmulti (kallewoof)
- #12836 9abdb7c Make `WalletInitInterface` and `DummyWalletInit` private, fix nullptr deref (promag)
- #12785 215158a Initialize `m_last_block_processed` to nullptr (practicalswift)

- #12932 8d651ae Remove redundant lambda function arg in handleTransactionChanged (laanwj)
- #12749 a84b056 feebumper: discard change outputs below discard rate (instagibbs)
- #12892 9b3370d introduce 'label' API for wallet (jnewbery)
- #12925 6d3de17 Logprint the start of a rescan (jonasschnelli)
- #12888 39439e5 debug log number of unknown wallet records on load (instagibbs)
- #12977 434150a Refactor `g_wallet_init_interface` to const reference (promag)
- #13017 65d7083 Add wallets management functions (promag)
- #12953 d1d54ae Deprecate accounts (jnewbery)
- #12909 476cb35 Make fee settings to be non-static members (MarcoFalke)
- #13002 487dcbe Do not treat bare multisig outputs as IsMine unless watched (sipa)
- #13028 783bb64 Make vpwallets usage thread safe (promag)
- #12507 2afdc29 Interrupt rescan on shutdown request (promag)
- #12729 979150b Get rid of ambiguous OutputType::NONE value (ryanofsky)
- #13079 5778d44 Fix rescanblockchain rpc to properly report progress (Empact)
- #12560 e03c0db Upgrade path for non-HD wallets to HD (achow101)
- #13161 7cc1bd3 Reset BerkeleyDB handle after connection fails (real-or-random)
- #13081 0dec5b5 Add compile time checking for `cs_wallet` runtime locking assertions (practicalswift)
- #13127 19a3a9e Add Clang thread safety annotations for variables guarded by `cs_db` (practicalswift)
- #10740 4cfe17c `loadwallet` RPC - load wallet at runtime (jnewbery)
- #12924 6738813 Fix hdmaster-key / seed-key confusion (scripted diff) (jnewbery)
- #13297 d82c5d1 Fix incorrect comment for `DeriveNewSeed` (jnewbery)
- #13063 6378eef Use shared pointer to retain wallet instance (promag)
- #13142 56fe3dc Separate IsMine from solvability (sipa)
- #13194 fd96d54 Remove template matching and pseudo opcodes (sipa)
- #13252 c4cc8d9 Refactor `ReserveKeyFromKeyPool` for safety (Empact)
- #13058 343d4e4 `createwallet` RPC - create new wallet at runtime (jnewbery)
- #13351 2140f6c Prevent segfault when sending to unspendable witness (MarcoFalke)
- #13060 3f0f394 Remove `getlabeladdress` RPC (jnewbery)
- #13111 000abbb Add `unloadwallet` RPC (promag)
- #13160 868cf43 Unlock spent outputs (promag)
- #13498 f54f373 Fixups from account API deprecation (jnewbery)
- #13491 61a044a Improve handling of INVALID in IsMine (sipa)
- #13425 028b0d9 Moving final scriptSig construction from `CombineSigna-`

- tures to ProduceSignature (PSBT signer logic) (achow101)
- #13564 88a15eb loadwallet shouldn't create new wallets (jnewbery)
- #12944 619cd29 ScanforWalletTransactions should mark input txns as dirty (instagibbs)
- #13630 d6b2235 Drop unused pindexRet arg to CMerkleTx::GetDepthInMainChain (Empact)
- #13566 ad552a5 Fix get balance (jnewbery)
- #13500 4a3e8c5 Decouple wallet version from client version (achow101)
- #13712 aba2e66 Fix non-determinism in ParseHDKeyPath(...). Avoid using an uninitialized variable in path calculation (practicalswift)
- #9662 6b6e854 Add createwallet "disableprivatekeys" option: a sane mode for watchonly-wallets (jonasschnelli)
- #13683 e8c7434 Introduce assertion to document the assumption that cache and cache_used are always set in tandem (practicalswift)
- #12257 5f7575e Use destination groups instead of coins in coin select (kallewoof)
- #13773 89a116d Fix accidental use of the comma operator (practicalswift)
- #13805 c88529a Correctly limit output group size (sdaftuar)
- #12992 26f59f5 Add wallet name to log messages (PierreRochard)
- #13667 b81a8a5 Fix backupwallet for multiwallets (domob1812)
- #13657 51c693d assert to ensure accuracy of CMerkleTx::GetBlocksToMaturity (Empact)
- #13812 9d86aad sum ancestors rather than taking max in output groups (kallewoof)
- #13876 8eb9870 Catch filesystem_error and raise InitError (MarcoFalke)
- #13808 13d51a2 shuffle coins before grouping, where warranted (kallewoof)
- #13666 2115cba Always create signatures with Low R values (achow101)
- #13917 0333914 Additional safety checks in PSBT signer (sipa)
- #13968 65e7a8b couple of walletcreatefundedpsbt fixes (instagibbs)
- #14055 2307a6e fix walletcreatefundedpsbt deriv paths, add test (instagibbs)

RPC and other APIs

- #12336 3843780 Remove deprecated rpc options (jnewbery)
- #12193 5dc00f6 Consistently use UniValue.pushKV instead of push_back(Pair()) (karel-3d) (MarcoFalke)
- #12409 0cc45ed Reject deprecated reserveChangeKey in fundrawtransaction (MarcoFalke)
- #10583 8a98dfe Split part of validateaddress into getaddressinfo (achow101)
- #10579 ffc6e48 Split signrawtransaction into wallet and non-wallet RPC command (achow101)
- #12494 e4ffcac Declare CMutableTransaction a struct in rawtransaction.h (Empact)

- #12503 0e26591 createmultisig no longer takes addresses (instagibbs)
- #12083 228b086 Improve getchaintxstats test coverage (promag)
- #12479 cd5e438 Add child transactions to getrawmempool verbose output (conscott)
- #11872 702e8b7 createrawtransaction: Accept sorted outputs (MarcoFalke)
- #12700 ebd84c Document RPC method aliasing (ryanofsky)
- #12727 8ee5c7b Remove unreachable help conditions in rpcwallet.cpp (lutangar)
- #12778 b648974 Add username and ip logging for RPC method requests (GabrielDav)
- #12717 ac898b6 rest: Handle utxo retrieval when ignoring the mempool (romanz)
- #12787 cd99e5b Adjust ifdef to avoid unreachable code (practicalswift)
- #11742 18815b4 Add testmempoolaccept (MarcoFalke)
- #12942 fefb817 Drop redundant testing of signrawtransaction prevtxs args (Empact)
- #11200 5f2a399 Allow for aborting rescans in the GUI (achow101)
- #12791 3a8a4dc Expose a transaction's weight via RPC (TheBlueMatt)
- #12436 6e67754 Adds a functional test to validate the transaction version number in the RPC output (251Labs)
- #12240 6f8b345 Introduced a new **fees** structure that aggregates all sub-field fee types denominated in BTC (mryandao)
- #12321 eac067a p2wsh and p2sh-p2wsh address in decodescript (fivepiece)
- #13090 17266a1 Remove Safe mode (achow101, laanwj)
- #12639 7eb7076 Reduce **cs_main** lock in listunspent (promag)
- #10267 7b966d9 New -includeconf argument for including external configuration files (kallewoof)
- #10757 b9551d3 Introduce getblockstats to plot things (jtimon)
- #13288 a589f53 Remove the need to include rpc/blockchain.cpp in order to put **GetDifficulty** under test (Empact)
- #13394 e1f8dce cli: Ignore libevent warnings (theuni)
- #13439 3f398d7 Avoid “duplicate” return value for invalid submitblock (TheBlueMatt)
- #13570 a247594 Add new “getmqnotifications” method (domob1812)
- #13072 b25a4c2 Update createmultisig RPC to support segwit (ajtowns)
- #12196 8fcea0 Add scantxoutset RPC method (jonasschnelli)
- #13557 b654723 BIP 174 PSBT Serializations and RPCs (achow101)
- #13697 f030410 Support output descriptors in scantxoutset (sipa)
- #13927 bced8ea Use pushKV in some new PSBT RPCs (domob1812)
- #13918 a9c56b6 Replace median fee rate with feerate percentiles in getblockstats (marcinja)
- #13721 9f23c16 Bugfixes for BIP 174 combining and deserialization (achow101)
- #13960 517010e Fix PSBT deserialization of 0-input transactions (achow101)

GUI

- #12416 c997f88 Fix Windows build errors introduced in #10498 (practicalswift)
- #11733 e782099 Remove redundant locks (practicalswift)
- #12426 bfa3911 Initialize members in WalletModel (MarcoFalke)
- #12489 e117cfe Bugfix: respect user defined configuration file (-conf) in QT settings (jonasschnelli)
- #12421 be263fa navigate to transaction history page after send (Sjors)
- #12580 ce56fdd Show a transaction's virtual size in its details dialog (dooglus)
- #12501 c8ea91a Improved "custom fee" explanation in tooltip (randolf)
- #12616 cff95a6 Set modal overlay hide button as default (promag)
- #12620 8a43bdc Remove TransactionTableModel::TxIDRole (promag)
- #12080 56cc022 Add support to search the address book (promag)
- #12621 2bac3e4 Avoid querying unnecessary model data when filtering transactions (promag)
- #12721 e476826 remove "new" button during receive-mode in addressbook (jonasschnelli)
- #12723 310dc61 Qt5: Warning users about invalid-BIP21 URI bitcoin:// (krab)
- #12610 25cf18f Multiwallet for the GUI (jonasschnelli)
- #12779 f4353da Remove unused method setupAmountWidget(...) (practicalswift)
- #12795 68484d6 do not truncate .dat extension for wallets in gui (instagibbs)
- #12870 1d54004 make clean removes `src/qt/moc_` files (Sjors)
- #13055 bdda14d Don't log to console by default (laanwj)
- #13141 57c57df fixes broken link on readme (marcoagner)
- #12928 ef006d9 Initialize non-static class members that were previously neither initialized where defined nor in constructor (practicalswift)
- #13158 81c533c Improve sendcoinsdialog readability (marcoagner)
- #11491 40c34a0 Add proxy icon in statusbar (mess110)
- #13264 2a7c53b Satoshi unit (GreatSock)
- #13097 e545503 Support wallets loaded dynamically (promag)
- #13284 f8be434 fix visual "overflow" of amount input (brandonrninefive)
- #13275 a315b79 use `[default wallet]` as name for wallet with no name (jonasschnelli)
- #13273 3fd0c23 Qt/Bugfix: fix handling default wallet with no name (jonasschnelli)
- #13341 25d2df2 Stop translating command line options (laanwj)
- #13043 6e249e4 OptionsDialog: add prune setting (Sjors)
- #13506 6579d80 load wallet in UI after possible init aborts (jonasschnelli)
- #13458 dc53f7f Drop qt4 support (laanwj)
- #13528 b877c39 Move BitcoinGUI initializers to class, fix initializer order warning (laanwj)

- #13536 baf3a3a coincontrol: Remove unused qt4 workaround (MarcoFalke)
- #13537 10ffca7 Peer table: Visualize inbound/outbound state for every row (wodry)
- #13791 2c14c1f Reject dialogs if key escape is pressed (promag)

Build system

- #12371 c9ca4f6 Add gitian PGP key: akx20000 (ghost)
- #11966 f4f4f51 clientversion: Use full commit hash for commit-based version descriptions (luke-jr)
- #12417 ae0fbf0 Upgrade mac_alias to 2.0.7 (droark)
- #12444 1f055ef gitian: Bump descriptors for (0.)17 (theuni)
- #12402 59e032b expat 2.2.5, ccache 3.4.1, miniupnpc 2.0.20180203 (fanquake)
- #12029 daa84b3 Add a makefile target for Doxygen documentation (Ov3rlo4d)
- #12466 6645eaf Only use D_DARWIN_C_SOURCE when building miniupnpc on darwin (fanquake)
- #11986 765a3eb zeromq 4.2.3 (fanquake)
- #12373 f13d756 Add build support for profiling (murrayn)
- #12631 a312e20 gitian: Alphabetize signing keys & add kallewoof key (kallewoof)
- #12607 29fad97 Remove ccache (fanquake)
- #12625 c4219ff biplist 1.0.3 (fanquake)
- #12666 05042d3 configure: UniValue 1.0.4 is required for pushKV(, bool) (luke-jr)
- #12678 6324c68 Fix a few compilation issues with Clang 7 and -Werror (vasild)
- #12692 de6bdfd Add configure options for various -fsanitize flags (eklitzke)
- #12901 7e23972 Show enabled sanitizers in configure output (practicalswift)
- #12899 3076993 macOS: Prevent Xcode 9.3 build warnings (AkioNak)
- #12715 8fd6243 Add 'make clean' rule (hkjn)
- #13133 a024a18 Remove python2 from configure.ac (ken2812221)
- #13005 cb088b1 Make -enable-debug to pick better options (practicalswift)
- #13254 092b366 Remove improper qt/moc_* cleaning glob from the general Makefile (Empact)
- #13306 f5a7733 split warnings out of CXXFLAGS (theuni)
- #13385 7c7508c Guard against accidental introduction of new Boost dependencies (practicalswift)
- #13041 5779dc4 Add linter checking for accidental introduction of locale dependence (practicalswift)
- #13408 70a03c6 crypto: cleanup sha256 build (theuni)
- #13435 cf7ca60 When build fails due to lib missing, indicate which one (Empact)

- #13445 8eb76f3 Reset default -g -O2 flags when enable debug (ken2812221)
- #13465 81069a7 Avoid concurrency issue when make multiple target (ken2812221)
- #13454 45c00f8 Make sure LC_ALL=C is set in all shell scripts (practical-swift)
- #13480 31145a3 Avoid copies in range-for loops and add a warning to detect them (theuni)
- #13486 66e1a08 Move rpc/util.cpp from libbitcoin-util to libbitcoin-server (ken2812221)
- #13580 40334c7 Detect if char equals int8_t (ken2812221)
- #12788 287e4ed Tune wildcards for LIBSECP256K1 target (kallewoof)
- #13611 b55f0c3 bugfix: Use __cpuid_count for gnu C to avoid gitian build fail (ken2812221)
- #12971 a6d14b1 Upgrade Qt to 5.9.6 (TheCharlatan)
- #13543 6c6a300 Add RISC-V support (laanwj)
- #13177 dcb154e GCC-7 and glibc-2.27 back compat code (ken2812221)
- #13659 90b1c7e add missing leveldb defines (theuni)
- #13368 c0f1569 Update gitian-build.sh for docker (achow101)
- #13171 19d8ca5 Change gitian-descriptors to use bionic instead (ken2812221)
- #13604 75bea05 Add depends 32-bit arm support for bitcoin-qt (TheCharlatan)
- #13623 9cdb19f Migrate gitian-build.sh to python (ken2812221)
- #13689 8c36432 disable Werror when building zmq (greenaddress)
- #13617 cf7f9ae release: Require macos 10.10+ (fanquake)
- #13750 c883653 use MacOS friendly sed syntax in qt.mk (Sjors)
- #13095 415f2bf update ax_boost_chrono/unit_test_framework (fanquake)
- #13732 e8ffec6 Fix Qt's rcc determinism (Fuzzbawls)
- #13782 8284f1d Fix opensslcode compile issue in gitian-build (ken2812221)
- #13696 2ab7208 Add aarch64 qt depends support for cross compiling bitcoin-qt (TheCharlatan)
- #13705 b413ba0 Add format string linter (practicalswift)
- #14000 48c8459 fix qt determinism (theuni)
- #14018 3e4829a Bugfix: NSIS: Exclude Makefile* from docs (luke-jr)
- #12906 048ac83 Avoid interface keyword to fix windows gitian build (ryanofsky)
- #13314 a9b6957 Fix FreeBSD build by including utilstrencodings.h (laanwj)

Tests and QA

- #12252 8d57319 Require all tests to follow naming convention (ajtowns)
- #12295 935eb8d Enable flake8 warnings for all currently non-violated rules (practicalswift)

- #11858 b4d8549 Prepare tests for Windows (MarcoFalke)
- #11771 2dbc4a4 Change invalidtxrequest to use BitcoinTestFramework (jnewbery)
- #12200 d09968f Bind functional test nodes to 127.0.0.1 (Sjors)
- #12425 26dc2da Add some script tests (richardkiss)
- #12455 23481fa Fix bip68 sequence test to reflect updated rpc error message (Empact)
- #12477 acd1e61 Plug memory leaks and stack-use-after-scope (MarcoFalke)
- #12443 07090c5 Move common args to bitcoin.conf (MarcoFalke)
- #12570 39dcac2 Add test cases for HexStr (`std::reverse_iterator` and corner cases) (kostaz)
- #12582 6012f1c Fix ListCoins test failure due to unset `g_wallet_allow_fallback_fee` (ryanofsky)
- #12516 7f99964 Avoid unintentional unsigned integer wraparounds in tests (practicalswift)
- #12512 955fd23 Don't test against the mempool min fee information in `mempool_limit.py` (Empact)
- #12600 29088b1 Add a test for large tx output scripts with segwit input (richardkiss)
- #12627 791c3ea Fix some tests to work on native windows (MarcoFalke)
- #12405 0f58d7f travis: Full clone for git subtree check (MarcoFalke)
- #11772 0630974 Change invalidblockrequest to use BitcoinTestFramework (jnewbery)
- #12681 1846296 Fix ComputeTimeSmart test failure with `-DDEBUG_LOCKORDER` (ryanofsky)
- #12682 9f04c8e travis: Clone depth 1 unless `$check_doc` (MarcoFalke)
- #12710 00d1680 Append scripts to new `test_list` array to fix bad assignment (jeffrade)
- #12720 872c921 Avoiding 'file' function name from python2 (jeffrade)
- #12728 4ba3d4f rename TestNode to TestP2PConn in tests (jnewbery)
- #12746 2405ce1 Remove unused argument `max_invalid` from `check_estimates(...)` (practicalswift)
- #12718 185d484 Require exact match in `assert_start_raises_init_error` (jnewbery, MarcoFalke)
- #12076 6d36f59 Use `node.datadir` instead of `tmpdir` in test framework (MarcoFalke)
- #12772 b43aba8 ci: Bump travis timeout for make check to 50m (jnewbery)
- #12806 18606eb Fix function names in `feature_blockdir` (MarcoFalke)
- #12811 0d8fc8d Make summary row bold-red if any test failed and show failed tests at end of table (laanwj)
- #12790 490644d Use `blockmaxweight` where tests previously had `block-maxsize` (conscott)
- #11773 f0f9732 Change `feature_block.py` to use BitcoinTestFramework (jnewbery)
- #12839 40f4baf Remove travis checkout depth (laanwj)

- #11817 2a09a78 Change `feature_csv_activation.py` to use Bitcoin-TestFramework (jnewbery)
- #12284 fa5825d Remove assigned but never used local variables. Enable Travis checking for unused local variables (practicalswift)
- #12719 9beded5 Add note about test suite naming convention in developer-notes.md (practicalswift)
- #12861 c564424 Stop `feature_block.py` from blowing up memory (jnewbery)
- #12851 648252e travis: Run verify-commits only on cron jobs (MarcoFalke)
- #12853 2106c4c Match full plain text by default (MarcoFalke)
- #11818 9a2db3b I accidentally (deliberately) killed it (the Comparison-TestFramework) (jnewbery)
- #12766 69310a3 Tidy up REST interface functional tests (romanz)
- #12849 83c7533 Add logging in loops in `p2p_sendhears.py` (ccd1e12)
- #12895 d6f10b2 Add note about test suite name uniqueness requirement to developer notes (practicalswift)
- #12856 27278df Add Metaclass for BitcoinTestFramework (WillAyd)
- #12918 6fc5a05 Assert on correct variable (kallewoof)
- #11878 a04440f Add Travis check for duplicate includes (practicalswift)
- #12917 cf8073f Windows fixups for functional tests (MarcoFalke)
- #12926 dd1ca9e Run unit tests in parallel (sipa)
- #12920 b1fdcf1 Fix sign for expected values (kallewoof)
- #12947 979f598 Wallet hd functional test speedup and clarification (instagibbs)
- #12993 0d69921 Remove compatibility code not needed now when we're on Python 3 (practicalswift)
- #12996 6a278e0 Remove redundant `bytes(...)` calls (practicalswift)
- #12949 6b46288 Avoid copies of CTransaction (MarcoFalke)
- #13007 0d12570 Fix dangling wallet pointer in vpwallets (promag)
- #13048 cac6d11 Fix `feature_block` flakiness (jnewbery)
- #12510 d5b2e98 Add `rpc_bind` test to default-run tests (laanwj)
- #13022 896a9d0 Attach node index to `test_node` AssertionError and print messages (jamesob)
- #13024 018c7e5 Add rpcauth pair that generated by rpcauth.py (ken2812221)
- #13013 a0079d4 bench: Amend `mempool_eviction` test for witness txs (MarcoFalke)
- #13051 e074097 Normalize executable location (MarcoFalke)
- #13056 106d929 Make rpcauth.py testable and add unit tests (nixbox)
- #13073 a785bc3 add rpcauth-test to AC_CONFIG_LINKS to fix out-of-tree make check (laanwj)
- #12830 25ad2f7 Clarify address book error messages, add tests (jamesob)
- #13082 24106a8 don't test against min relay fee information in `mining_prioritisetransaction.py` (kristapsk)
- #13003 8d045a0 Add test for orphan handling (MarcoFalke)

- #13105 9e9b48d Add `--failfast` option to functional test runner (jamesob)
- #13130 3186ad4 Fix race in `rpc_deprecated.py` (jnewbery)
- #13136 baf6b4e Fix flake8 warnings in several wallet functional tests (jnewbery)
- #13094 bf9b03d Add test for 64-bit Windows PE, modify 32-bit test results (ken2812221)
- #13183 9458b05 travis: New travis job for `check_docs` steps (glaksmono)
- #12265 1834d4d fundrawtransaction: lock watch-only shared address (kallewoof)
- #13188 4a50ec0 Remove unused option `--srcdir` (MarcoFalke)
- #12755 612ba35 Better stderr testing (jnewbery)
- #13198 196c5a9 Avoid printing to console during cache creation (sdaftuar)
- #13075 cb9bbf7 Remove 'account' API from wallet functional tests (jnewbery)
- #13221 ffa86af travis: Rename the build stage `check_doc` to `lint` (practicalswift)
- #13205 3cbd25f Remove spurious error log in `p2p_segwit.py` (jnewbery)
- #13291 536120e Don't include `torcontrol.cpp` into the test file (Empact)
- #13281 2ac6315 Move linters to test/lint, add readme (MarcoFalke)
- #13215 f8a29ca travis: Build tests on ubuntu 18.04 with docker (ken2812221)
- #13349 24f7011 bench: Don't return a bool from main (laanwj)
- #13347 87a9d03 travis: Skip cache for lint stage (MarcoFalke)
- #13355 0b1c0c4 Fix "gmake check" under OpenBSD 6.3 (probably *BSD): Avoid using GNU grep specific regexp handling (practicalswift)
- #13353 d4f6dac Fixup setting of `PATH` env var (MarcoFalke)
- #13352 e24bf1c Avoid checking reject code for now (MarcoFalke)
- #13383 2722a1f bench: Use non-throwing `parsedouble(...)` instead of throwing `boost::lexical_cast(...)` (practicalswift)
- #13367 264efdc Increase `includeconf` test coverage (MarcoFalke)
- #13404 3d3d8ae speed up of `tx_validationcache_tests` by reusing of `CTransaction` (lucash-dev)
- #13421 531a033 Remove `portseed_offset` from test runner (MarcoFalke)
- #13440 5315660 Log as utf-8 (MarcoFalke)
- #13066 fa4b906 Migrate `verify-commits` script to python, run in travis (ken2812221)
- #13447 4b1edd3 travis: Increase `travis_wait` time while verifying commits (ken2812221)
- #13350 f532d52 Add logging to provide anchor points when debugging `p2p_sendheaders` (lmanners)
- #13406 4382f19 travis: Change mac goal to all deploy (ken2812221)
- #13457 b222138 Drop variadic macro (MarcoFalke)
- #13512 3a45493 mininode: Expose connection state through `is_connected` (MarcoFalke)
- #13496 9ab4c2a Harden `lint-filenames.sh` (wodry)
- #13219 08516e0 bench: Add block assemble benchmark (MarcoFalke)

- #13530 b1dc39d bench: Add missing pow.h header (laanwj)
- #12686 2643fa5 Add -ftrapv to CFLAGS and CXXFLAGS when -enable-debug is used. Enable -ftrapv in Travis (practicalswift)
- #12882 d96bdd7 Make `test_bitcoin` pass under ThreadSanitizer (clang). Fix lock-order-inversion (potential deadlock) (practicalswift)
- #13535 2328039 `wallet_basic`: Specify minimum required amount for listunspent (MarcoFalke)
- #13551 c93c360 Fix incorrect documentation for test case `cuckoocache_hit_rate_ok` (practicalswift)
- #13563 b330f3f bench: Simplify coinselection (promag)
- #13517 a6ed99a Remove need to handle the network thread in tests (MarcoFalke)
- #13522 686e97a Fix `p2p_sendheaders` race (jnewbery)
- #13467 3dc2dcf Make `p2p_segwit` easier to debug (jnewbery)
- #13598 0212187 bench: Fix incorrect behaviour in `prevector.cpp` (AkioNak)
- #13565 b05ded1 Fix `AreInputsStandard` test to reference the proper script-PubKey (Empact)
- #13145 d3dae3d Use common `getPath` method to create temp directory in tests (winder)
- #13645 2ea7eb6 skip `rpc_zmq` functional test as necessary (jamesob)
- #13626 8f1106d Fix some TODOs in `p2p_segwit` (MarcoFalke)
- #13138 8803c91 Remove accounts from `wallet_importprunedfunds.py` (jnewbery)
- #13663 cbc9b50 Avoid read/write to default datadir (MarcoFalke)
- #13682 f8a32a3 bench: Remove unused variable (practicalswift)
- #13638 6fcd5e Use `MAX_SCRIPT_ELEMENT_SIZE` from `script.py` (domob1812)
- #13687 9d26b69 travis: Check that `~/.bitcoin` is never created (MarcoFalke)
- #13715 e1260a7 fixes mininode's `P2PConnection` sending messages on closing transport (marcoagner)
- #13729 aa9429a travis: Avoid unnecessarily setting env variables on the lint build (Empact)
- #13747 ab28b5b Skip `P2PConnection`'s `is_closing()` check when not available (domob1812)
- #13650 7a9bca6 travis: Don't store debug info if `-enable-debug` is set (ken2812221)
- #13711 f98d1e0 bench: Add benchmark for unserialize `prevector` (AkioNak)
- #13771 365384f travis: Retry to fetch docker image (MarcoFalke)
- #13806 4d550ff Fix `bench/block_assemble` assert failure (jamesob)
- #13779 d25079a travis: Improve readability of `travis.yml` and log outputs (scravy)
- #13822 0fb9c87 bench: Make coinselection output groups pass eligibility filter (achow101)

- #13247 e83d82a Add tests to SingleThreadedSchedulerClient() and document the memory model (skeees)
- #13811 660abc1 travis: Run `bench_bitcoin` once (MarcoFalke)
- #13837 990e182 Extract `rpc_timewait` as test param (MarcoFalke)
- #13851 9c4324d fix locale for lint-shell (scravy)
- #13823 489b51b quote path in authproxy for external multiwallets (MarcoFalke)
- #13849 2b67354 travis: Use only travis jobs: instead of mix of jobs+matrix (scravy)
- #13859 2384323 Add emojis to `test_runner` path and wallet filename (MarcoFalke)
- #13916 8ac7125 `wait_for_verack` by default (MarcoFalke)
- #13669 f66e1c7 Cleanup `create_transaction` implementations (conscott)
- #13924 09ada21 Simplify comparison in `rpc_blockchain.py` (domob1812)
- #13913 a08533c Remove redundant `checkmempool/checkblockindex` `extra_args` (MarcoFalke)
- #13915 a04888a Add test for max number of entries in locator (MarcoFalke)
- #13867 1b04b55 Make extended tests pass on native Windows (MarcoFalke)
- #13944 0df7a6c Port usage of deprecated `optparse` module to `argparse` module (Kvaciral)
- #13928 b8eb0df `blocktools` enforce named args for amount (MarcoFalke)
- #13054 bffb35f Enable automatic detection of undefined names in Python tests scripts. Remove wildcard imports (practicalswift)
- #14069 cf3d7f9 Use `assert not BOOST_CHECK_*` from multithreaded tests (skeees)
- #14071 fab0fbe Stop `txindex` thread before calling destructor (MarcoFalke)

Miscellaneous

- #11909 8897135 contrib: Replace developer keys with list of pgp fingerprints (MarcoFalke)
- #12394 fe53d5f `gitian-builder.sh`: fix `-setup` doc, since `lxc` is default (Sjors)
- #12468 294a766 Add missing newline in `init.cpp` log message (Aesti)
- #12308 dcfe218 contrib: Add support for out-of-tree builds in `genmanpages.sh` (laanwj)
- #12451 aae64a2 Bump `leveldb` subtree (MarcoFalke)
- #12527 d77b4a7 `gitian-build.sh`: fix `signProg` being recognized as two parameters (ken2812221)
- #12588 d74b01d `utils`: Remove deprecated `pyzmq` call from `python zmq` example (kosciiej)

- #10271 bc67982 Use `std::thread::hardware_concurrency`, instead of Boost, to determine available cores (fanquake)
- #12097 14475e2 scripts: Lint-whitespace: use perl instead of grep -p (Sjors)
- #12098 17c44b2 scripts: Lint-whitespace: add param to check last n commits (Sjors)
- #11900 842f61a script: Simplify checkminimalpush checks, add safety assert (instagibbs)
- #12567 bb98aec util: Print timestamp strings in logs using iso 8601 formatting (practicalswift)
- #12572 d8d9162 script: Lint-whitespace: find errors more easily (AkioNak)
- #10694 ae5bcc7 Remove redundant code in MutateTxSign(CMutableTransaction&, const std::string&) (practicalswift)
- #12659 3d16f58 Improve Fatal LevelDB Log Messages (eklitzke)
- #12643 0f0229d util: Remove unused `sync_chain` (MarcoFalke)
- #12102 7fb8fb4 Apply hardening measures in bitcoind systemd service file (Flowdalic)
- #12652 55f490a bitcoin-cli: Provide a better error message when bitcoind is not running (practicalswift)
- #12630 c290508 Provide useful error message if datadir is not writable (murrayn)
- #11881 624bee9 Remove Python2 support (jnewbery)
- #12821 082e26c contrib: Remove unused import string (MarcoFalke)
- #12829 252c1b0 Python3 fixup (jnewbery)
- #12822 ff48f62 Revert 7deba93bdc76616011a9f493cbc203d60084416f and fix expired-key-sigs properly (TheBlueMatt)
- #12820 5e53b80 contrib: Fix check-doc script regexes (MarcoFalke)
- #12713 4490871 Track negated options in the option parser (eklitzke)
- #12708 b2e5fe8 Make verify-commits.sh test that merges are clean (sipa)
- #12891 3190785 logging: Add lint-logs.sh to check for newline termination (jnewbery)
- #12923 a7cbe38 util: Pass `pthread_self()` to `pthread_setschedparam` instead of 0 (laanwj)
- #12871 fb17fae Add shell script linting: Check for shellcheck warnings in shell scripts (practicalswift)
- #12970 5df84de logging: Bypass timestamp formatting when not logging (theuni)
- #12987 fe8fa22 tests/tools: Enable additional Python flake8 rules for automatic linting via Travis (practicalswift)
- #12972 0782508 Add python3 script shebang lint (ken2812221)
- #13004 58bbc55 Print to console by default when not run with -daemon (practicalswift)
- #13039 8b4081a Add logging and error handling for file syncing (laanwj)
- #13020 4741ca5 Consistently log CValidationState on call failure (Empact)
- #13031 826acc9 Fix for utiltime to compile with msvc (sipsorcery)
- #13119 81743b5 Remove script to clean up datadirs (MarcoFalke)

- #12954 5a66642 util: Refactor logging code into a global object (jimpo)
- #12769 35eb9d6 Add systemd service to bitcoind in debian package (ghost)
- #13146 0bc980b rpcauth: Make it possible to provide a custom password (laanwj)
- #13148 b62b437 logging: Fix potential use-after-free in logprintstr(...) (practicalswift)
- #13214 0612d96 Enable Travis checking for two Python linting rules we are currently not violating (practicalswift)
- #13197 6826989 util: Warn about ignored recursive -includeconf calls (kallewoof)
- #13176 d9ebb63 Improve CRollingBloomFilter performance: replace modulus with FastMod (martinus)
- #13228 d792e47 Add script to detect circular dependencies between source modules (sipa)
- #13320 e08c130 Ensure gitian-build.sh uses bash (jhfrontz)
- #13301 e4082d5 lint: Add linter to error on `#include <*.cpp>` (Empact)
- #13374 56f6936 utils and libraries: checking for bitcoin address in translations (kaplanmaxe)
- #13230 7c32b41 Simplify include analysis by enforcing the developer guide's include syntax (practicalswift)
- #13450 32bf4c6 Add linter: Enforce the source code file naming convention described in the developer notes (practicalswift)
- #13479 fa2ea37 contrib: Fix cve-2018-12356 by hardening the regex (loganaden)
- #13448 a90ca40 Add linter: Make sure we explicitly open all text files using UTF-8 encoding in Python (practicalswift)
- #13494 d67eff8 Follow-up to #13454: Fix broken build by exporting `LC_ALL=C` (practicalswift)
- #13510 03f3925 Scripts and tools: Obsolete `#!/bin/bash` shebang (DesWurstes)
- #13577 c9eb8d1 logging: Avoid nstart may be used uninitialized in apinitmain warning (mruddy)
- #13603 453ae5e bitcoin-tx: Stricter check for valid integers (domob1812)
- #13118 c05c93c RPCAuth Detection in Logs (Linrono)
- #13647 4027ec1 Scripts and tools: Fix `BIND_NOW` check in security-check.py (conradopl)
- #13692 f5d166a contrib: Clone core repo in gitian-build (MarcoFalke)
- #13699 4c6d1b9 contrib: Correct version check (kallewoof)
- #13695 dcc0cff lint: Add linter for circular dependencies (Empact)
- #13733 0d1ebf4 utils: Refactor argsmanager a little (AtsukiTak)
- #13714 29b4ee6 contrib: Add lxc network setup for bionic host (ken2812221)
- #13764 f8685f4 contrib: Fix test-security-check fail in ubuntu 18.04 (ken2812221)
- #13809 77168f7 contrib: Remove debian and rpm subfolder (MarcoFalke)
- #13799 230652c Ignore unknown config file options; warn instead of error

- (sipa)
- #13894 [df9f712](#) shutdown: Stop threads before resetting ptrs (MarcoFalke)
- #13925 [71dec5c](#) Merge leveldb subtree (MarcoFalke)
- #13939 [ef86f26](#) lint: Make format string linter understand basic template parameter syntax (practicalswift)
- #14105 [eb202ea](#) util: Report parse errors in configuration file (laanwj)
- #12604 [9903537](#) Add DynamicMemoryUsage() to CDBWrapper to estimate LevelDB memory use (eklitze)
- #12495 [047865e](#) Increase LevelDB `max_open_files` (eklitze)
- #12784 [e80716d](#) Fix bug in memory usage calculation (unintended integer division) (practicalswift)
- #12618 [becd8dd](#) Set `SCHED_BATCH` priority on the loadblk thread (eklitze)
- #12854 [5ca1509](#) Add P2P, Network, and Qt categories to the desktop icon (luke-jr)
- #11862 [4366f61](#) Network specific conf sections (ajtowns)
- #13441 [4a7e64f](#) Prevent shared conf files from failing with different available options in different binaries (achow101)
- #13471 [5eca4e8](#) For AVX2 code, also check for AVX, XSAVE, and OS support (sipa)
- #13503 [c655b2c](#) Document FreeBSD quirk. Fix FreeBSD build: Use `std::min(...)` to allow for compilation under certain FreeBSD versions (practicalswift)
- #13725 [07ce278](#) Fix bitcoin-cli `-version` (Empact)

Documentation

- #12306 [216f9a4](#) Improvements to UNIX documentation (axvr)
- #12309 [895fbd7](#) Explain how to update chainTxData in release process (laanwj)
- #12317 [85123be](#) Document method for reviewers to verify chainTxData (jnewbery)
- #12331 [d32528e](#) Properly alphabetize output of CLI `-help` option (murrayn)
- #12322 [c345148](#) Remove step making cloned repository world-writable for Windows build (murrayn)
- #12354 [b264528](#) add gpg key for fivepiece (fivepiece)
- #11761 [89005dd](#) initial QT documentation (Sjors)
- #12232 [fdc2188](#) Improve “Turn Windows Features On or Off” step (MCFX2)
- #12487 [4528f74](#) init: Remove translation for `-blockmaxsize` option help (laanwj)
- #12546 [a4a5fc7](#) Minor improvements to Compatibility Notes (randolf)
- #12434 [21e2670](#) dev-notes: Members should be initialized (MarcoFalke)
- #12452 [71f56da](#) clarified systemd installation instructions in init.md for Ubuntu users (DaveFromBinary)

- #12615 1f93491 allow for SIGNER containing spaces (ken2812221)
- #12603 85424d7 PeerLogicValidation interface (jamesob)
- #12581 12ac2f0 Mention configure without wallet in FreeBSD instructions (dbolser)
- #12619 8a709fb Give hint about gitian not able to download (kallewoof)
- #12668 de2fcaa do update before fetching packages in WSL build guide (nvercamm)
- #12586 e7721e6 Update osx brew install instruction (fanquake)
- #12760 7466a26 Improve documentation on standard communication channels (jimpo)
- #12797 0415b1e init: Fix help message for checkblockindex (MarcoFalke)
- #12800 2d97611 Add note about our preference for scoped enumerations (“enum class”) (practicalswift)
- #12798 174d016 Refer to witness reserved value as spec. in the BIP (MarcoFalke)
- #12759 d3908e2 Improve formatting of developer notes (eklitze)
- #12877 2b54155 Use bitcoind in Tor documentation (knoxcard)
- #12896 b15485e Fix conflicting statements about initialization in developer notes (practicalswift)
- #12850 319991d add qrencode to brew install instructions (buddilla)
- #12007 cd8e45b Clarify the meaning of fee delta not being a fee rate in prioritisetransaction RPC (honzik666)
- #12927 06ead15 fixed link, replaced QT with Qt (trulex)
- #12852 ebd786b devtools: Setup ots git integration (MarcoFalke)
- #12933 3cf76c2 Refine header include policy (MarcoFalke)
- #12951 6df0c6c Fix comment in FindForkInGlobalIndex (jamesob)
- #12982 a63b4e3 Fix inconsistent namespace formatting guidelines (ryanofsky)
- #13026 9b3a67e Fix include comment in src/interfaces/wallet.h (promag)
- #13012 d1e3c5e Add comments for chainparams.h, validation.cpp (jamesob)
- #13064 569e381 List support for BIP173 in bips.md (sipa)
- #12997 646b7f6 build-windows: Switch to Artful, since Zesty is EOL (MarcoFalke)
- #12384 c5f7efe Add version footnote to tor.md (Willtech)
- #13165 627c376 Mention good first issue list in CONTRIBUTING.md (fanquake)
- #13295 fb77310 Update OpenBSD build instructions for OpenBSD 6.3 (practicalswift)
- #13340 3a8e3f4 remove leftover check-doc documentation (fanquake)
- #13346 60f0358 update bitcoin-dot-org links in release-process.md (fanquake)
- #13372 f014933 split FreeBSD build instructions out of build-unix.md (steveurusso)
- #13366 861de3b Rename “OS X” to the newer “macOS” convention (giulio92)

- #13369 f8bcef3 update transifex doc link (mess110)
- #13312 b22115d Add a note about the source code filename naming convention (practicalswift)
- #13460 1939536 Remove note to install all boost dev packages (MarcoFalke)
- #13476 9501938 Fix incorrect shell quoting in FreeBSD build instructions (murrayn)
- #13402 43fa355 Document validationinterace callback blocking deadlock potential (TheBlueMatt)
- #13488 d6cf4bd Improve readability of “Squashing commits” (wodry)
- #13531 ee02deb Clarify that mempool txiter is `const_iterator` (MarcoFalke)
- #13418 01f9098 More precise explanation of parameter onlynet (wodry)
- #13592 1756cb4 Modify policy to not translate command-line help (ken2812221)
- #13588 b77c38e Improve doc of options addnode, connect, seednode (wodry)
- #13614 17e9106 Update command line help for -printtoconsole and -debuglogfile (satwo, fanquake)
- #13605 8cc048e corrected text to reflect new(er) process of specifying fingerprints (jhfrontz)
- #13481 b641f60 Rewrite some validation docs as lock annotations (MarcoFalke)
- #13680 30640f8 Remove outdated comment about miner ignoring CPFP (jamesob)
- #13625 7146672 Add release notes for -printtoconsole and -debuglogfile changes (satwo)
- #13718 f7f574d Specify preferred Python string formatting technique (masonicboom)
- #12764 10b9a81 Remove field in getblocktemplate help that has never been used (conscott)
- #13742 d2186b3 Adjust bitcoincore.org links (MarcoFalke)
- #13706 94dd89e Minor improvements to release-process.md (MitchellCash)
- #13775 ef4fac0 Remove newlines from error message (practicalswift)
- #13803 feb7dd9 add note to contributor docs about warranted PR’s (kallewoof)
- #13814 67af7ef Add BIP174 to list of implemented BIPs (sipa)
- #13835 c1cba35 Fix memory consistency model in comment (skeees)
- #13824 aa30e4b Remove outdated net comment (MarcoFalke)
- #13853 317477a correct versions in dependencies.md (fanquake)
- #13872 37ab117 Reformat -help output for help2man (real-or-random)
- #13717 8c3c402 Link to python style guidelines from developer notes (masonicboom)
- #13895 1cd5f2c fix GetWarnings docs to reflect behavior (Empact)
- #13911 3e3a50a Revert translated string change, clarify wallet log messages (PierreRochard)

- #13908 d6faea4 upgrade rescan time warning from minutes to >1 hour (masonicboom)
- #13905 73a09b4 fixed bitcoin-cli -help output for help2man (hebasto)
- #14100 2936dbc Change documentation for =0 for non-boolean options (laanwj)
- #14096 465a583 Add reference documentation for descriptors language (sipa)
- #12757 0c5f67b Clarify include guard naming convention (practicalswift)
- #13844 d3325b0 Correct the help output for **-prune** (hebasto)

Credits

Thanks to everyone who directly contributed to this release:

- 251
- 532479301
- Aaron Clauson
- Akio Nakamura
- Akira Takizawa
- Alex Morcos
- Alex Vear
- Alexey Ivanov
- Alin Rus
- Andrea Comand
- Andrew Chow
- Anthony Towns
- AtsukiTak
- Ben Woosley
- Bernhard M. Wiedemann
- Brandon Ruggles
- buddilla
- ccdle12
- Chris Moore
- Chun Kuan Lee
- Clem Taylor
- Conor Scott
- Conrado Gouvea
- Cory Fields
- Cristian Mircea Messel
- ctp-tsteenholdt
- Damian Williamson
- Dan Bolser
- Daniel Kraft
- Darko Janković
- DaveFromBinary

- David A. Harding
- DesWurstes
- Dimitris Apostolou
- donaloconnor
- Douglas Roark
- DrahtBot
- Drew Rasmussen
- e0
- Ernest Hemingway
- Ethan Heilman
- Evan Klitzke
- fanquake
- Felix Wolfsteller
- fivepiece
- Florian Schmaus
- Fuzzbawls
- Gabriel Davidian
- Giulio Lombardo
- Gleb
- Grady Laksmono
- GreatSock
- Gregory Maxwell
- Gregory Sanders
- Hennadii Stepanov
- Henrik Jonsson
- Indospace.io
- James O'Beirne
- Jan Čapek
- Jeff Frontz
- Jeff Rade
- Jeremy Rubin
- JeremyRand
- Jesse Cohen
- Jim Posen
- joemphilips
- John Bampton
- John Newbery
- johnlow95
- Johnson Lau
- Jonas Nick
- Jonas Schnelli
- João Barbosa
- Jorge Timón
- Josh Hartshorn
- Julian Fleischer
- kallewoof

- Karel Bilek
- Karl-Johan Alm
- Ken Lee
- Kevin Pan
- Kosta Zertsekel
- Kristaps Kaupe
- Kvaciral
- Lawrence Nahum
- Linrono
- lmanners
- Loganaden Velvindron
- Lowell Manners
- lucash.dev@gmail.com
- Luke Dashjr
- lutangar
- Marcin Jachymiak
- marcoagner
- MarcoFalke
- Mark Erhardt
- Mark Friedenbach
- Martin Ankerl
- Mason Simon
- Matt Corallo
- Matteo Sumberaz
- Max Kaplan
- MeshCollider
- Michał Zabielski
- Mitchell Cash
- mruddy
- mryandao
- murrayn
- Nick Vercammen
- Nicolas Dorier
- Nikolay Mitev
- okayplanet
- Pierre Rochard
- Pieter Wuille
- practicalswift
- Qasim Javed
- Randolph Richardson
- Richard Kiss
- Roman Zeyde
- Russell Yanofsky
- Samuel B. Atwood
- Sebastian Kung
- Sjors Provoost

- Steve Lee
- steverusso
- Suhas Daftuar
- Tamas Blummer
- TheCharlatan
- Thomas Kerin
- Thomas Snider
- Tim Ruffing
- Varunram
- Vasil Dimov
- Will Ayd
- William Robinson
- winder
- Wladimir J. van der Laan
- wodry

And to those that reported security issues:

- awemany (for CVE-2018-17144, previously credited as “anonymous reporter”)

As well as everyone that helped translating on Transifex.