

Bitcoin Core version 0.18.0 is now available from:

<https://bitcoincore.org/bin/bitcoin-core-0.18.0/>

This is a new major version release, including 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 for older versions), then run the installer (on Windows) or just copy over `/Applications/Bitcoin-Qt` (on Mac) or `bitcoind/bitcoin-qt` (on Linux).

The first time you run version 0.15.0 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 or later. Upgrading directly from 0.7.x and earlier without redownloading the blockchain is not supported. However, as usual, old wallet versions are still supported.

## Compatibility

Bitcoin Core is supported and extensively tested on operating systems using the Linux kernel, macOS 10.10+, and Windows 7 and newer. It is not recommended to use Bitcoin Core on unsupported systems.

Bitcoin Core should also work on most other Unix-like systems but is not as 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. Additionally, Bitcoin Core does not yet change appearance when macOS "dark mode" is activated.

In addition to previously-supported CPU platforms, this release's pre-compiled distribution also provides binaries for the RISC-V platform.

If you are using the `systemd` unit configuration file located at `contrib/init/bitcoind.service`, it has been changed to use `/var/lib/bitcoind` as the data directory instead

of `~bitcoin/.bitcoin`. When switching over to the new configuration file, please make sure that the filesystem on which `/var/lib/bitcoind` will exist has enough space (check using `df -h /var/lib/bitcoind`), and optionally copy over your existing data directory. See the systemd init file section for more details.

## Known issues

### Wallet GUI

For advanced users who have both (1) enabled coin control features, and (2) are using multiple wallets loaded at the same time: The coin control input selection dialog can erroneously retain wrong-wallet state when switching wallets using the dropdown menu. For now, it is recommended not to use coin control features with multiple wallets loaded.

## Notable changes

### Mining

- Calls to `getblocktemplate` will fail if the segwit rule is not specified. Calling `getblocktemplate` without segwit specified is almost certainly a misconfiguration since doing so results in lower rewards for the miner. Failed calls will produce an error message describing how to enable the segwit rule.

### Configuration option changes

- A warning is printed if an unrecognized section name is used in the configuration file. Recognized sections are `[test]`, `[main]`, and `[regtest]`.
- Four new options are available for configuring the maximum number of messages that ZMQ will queue in memory (the “high water mark”) before dropping additional messages. The default value is 1,000, the same as was used for previous releases. See the ZMQ documentation for details.
- The `rpccallowip` option can no longer be used to automatically listen on all network interfaces. Instead, the `rpcbind` parameter must be used to specify the IP addresses to listen on. Listening for RPC commands over a public network connection is insecure and should be disabled, so a warning is now printed if a user selects such a configuration. If you need to expose RPC in order to use a tool like Docker, ensure you only bind RPC to your localhost, e.g. `docker run [...] -p 127.0.0.1:8332:8332` (this is an extra `:8332` over the normal Docker port specification).
- The `rpcpassword` option now causes a startup error if the password set in the configuration file contains a hash character (`#`), as it’s ambiguous

whether the hash character is meant for the password or as a comment.

- The `whitelistforcerelay` option is used to relay transactions from whitelisted peers even when not accepted to the mempool. This option now defaults to being off, so that changes in policy and disconnect/ban behavior will not cause a node that is whitelisting another to be dropped by peers. Users can still explicitly enable this behavior with the command line option (and may want to consider contacting the Bitcoin Core project to let us know about their use-case, as this feature could be deprecated in the future).

## systemd init file

The systemd init file (`contrib/init/bitcoind.service`) has been changed to use `/var/lib/bitcoind` as the data directory instead of `~bitcoin/.bitcoin`. This change makes Bitcoin Core more consistent with other services, and makes the systemd init config more consistent with existing Upstart and OpenRC configs.

The configuration, PID, and data directories are now completely managed by systemd, which will take care of their creation, permissions, etc. See `systemd.exec(5)` for more details.

When using the provided init files under `contrib/init`, overriding the `datadir` option in `/etc/bitcoin/bitcoin.conf` will have no effect. This is because the command line arguments specified in the init files take precedence over the options specified in `/etc/bitcoin/bitcoin.conf`.

## Documentation

- A new short document about the JSON-RPC interface describes cases where the results of an RPC might contain inconsistencies between data sourced from different subsystems, such as wallet state and mempool state. A note is added to the REST interface documentation indicating that the same rules apply.
- Further information is added to the JSON-RPC documentation about how to secure this interface.
- A new document about the `bitcoin.conf` file describes how to use it to configure Bitcoin Core.
- A new document introduces Bitcoin Core's BIP174 Partially-Signed Bitcoin Transactions (PSBT) interface, which is used to allow multiple programs to collaboratively work to create, sign, and broadcast new transactions. This is useful for offline (cold storage) wallets, multisig wallets, coinjoin implementations, and many other cases where two or more programs need to interact to generate a complete transaction.

- The output script descriptor documentation has been updated with information about new features in this still-developing language for describing the output scripts that a wallet or other program wants to receive notifications for, such as which addresses it wants to know received payments. The language is currently used in multiple new and updated RPCs described in these release notes and is expected to be adapted to other RPCs and to the underlying wallet structure.

## Build system changes

- A new `--disable-bip70` option may be passed to `./configure` to prevent Bitcoin-Qt from being built with support for the BIP70 payment protocol or from linking libssl. As the payment protocol has exposed Bitcoin Core to libssl vulnerabilities in the past, builders who don't need BIP70 support are encouraged to use this option to reduce their exposure to future vulnerabilities.
- The minimum required version of Qt (when building the GUI) has been increased from 5.2 to 5.5.1 (the depends system provides 5.9.7)

## New RPCs

- `getnodeaddresses` returns peer addresses known to this node. It may be used to find nodes to connect to without using a DNS seeder.
- `listwalletdir` returns a list of wallets in the wallet directory (either the default wallet directory or the directory configured by the `-walletdir` parameter).
- `getrpcinfo` returns runtime details of the RPC server. At the moment, it returns an array of the currently active commands and how long they've been running.
- `deriveaddresses` returns one or more addresses corresponding to an output descriptor.
- `getdescriptorinfo` accepts a descriptor and returns information about it, including its computed checksum.
- `joinpsbts` merges multiple distinct PSBTs into a single PSBT. The multiple PSBTs must have different inputs. The resulting PSBT will contain every input and output from all of the PSBTs. Any signatures provided in any of the PSBTs will be dropped.
- `analyzepsbt` examines a PSBT and provides information about what the PSBT contains and the next steps that need to be taken in order to complete the transaction. For each input of a PSBT, `analyzepsbt` provides information about what information is missing for that input, including whether a UTXO needs to be provided, what pubkeys still need

to be provided, which scripts need to be provided, and what signatures are still needed. Every input will also list which role is needed to complete that input, and **analyzepsbt** will also list the next role in general needed to complete the PSBT. **analyzepsbt** will also provide the estimated fee rate and estimated virtual size of the completed transaction if it has enough information to do so.

- **utxupdatepsbt** searches the set of Unspent Transaction Outputs (UTXOs) to find the outputs being spent by the partial transaction. PSBTs need to have the UTXOs being spent to be provided because the signing algorithm requires information from the UTXO being spent. For segwit inputs, only the UTXO itself is necessary. For non-segwit outputs, the entire previous transaction is needed so that signers can be sure that they are signing the correct thing. Unfortunately, because the UTXO set only contains UTXOs and not full transactions, **utxupdatepsbt** will only add the UTXO for segwit inputs.

## Updated RPCs

Note: some low-level RPC changes mainly useful for testing are described in the Low-level Changes section below.

- **getpeerinfo** now returns an additional **minfeefilter** field set to the peer's BIP133 fee filter. You can use this to detect that you have peers that are willing to accept transactions below the default minimum relay fee.
- The mempool RPCs, such as **getrawmempool** with **verbose=true**, now return an additional "bip125-replaceable" value indicating whether the transaction (or its unconfirmed ancestors) opts-in to asking nodes and miners to replace it with a higher-feerate transaction spending any of the same inputs.
- **settxfee** previously silently ignored attempts to set the fee below the allowed minimums. It now prints a warning. The special value of "0" may still be used to request the minimum value.
- **getaddressinfo** now provides an **ischange** field indicating whether the wallet used the address in a change output.
- **importmulti** has been updated to support P2WSH, P2WPKH, P2SH-P2WPKH, and P2SH-P2WSH. Requests for P2WSH and P2SH-P2WSH accept an additional **witnessscript** parameter.
- **importmulti** now returns an additional **warnings** field for each request with an array of strings explaining when fields are being ignored or are inconsistent, if there are any.
- **getaddressinfo** now returns an additional **solvable** boolean field when Bitcoin Core knows enough about the address's scriptPubKey, optional

redeemScript, and optional witnessScript in order for the wallet to be able to generate an unsigned input spending funds sent to that address.

- The `getaddressinfo`, `listunspent`, and `scantxoutset` RPCs now return an additional `desc` field that contains an output descriptor containing all key paths and signing information for the address (except for the private key). The `desc` field is only returned for `getaddressinfo` and `listunspent` when the address is solvable.
- `importprivkey` will preserve previously-set labels for addresses or public keys corresponding to the private key being imported. For example, if you imported a watch-only address with the label “cold wallet” in earlier releases of Bitcoin Core, subsequently importing the private key would default to resetting the address’s label to the default empty-string label (“”). In this release, the previous label of “cold wallet” will be retained. If you optionally specify any label besides the default when calling `importprivkey`, the new label will be applied to the address.
- See the Mining section for changes to `getblocktemplate`.
- `getmininginfo` now omits `currentblockweight` and `currentblocktx` when a block was never assembled via RPC on this node.
- The `getrawtransaction` RPC & REST endpoints no longer check the unspent UTXO set for a transaction. The remaining behaviors are as follows:
  1. If a blockhash is provided, check the corresponding block.
  2. If no blockhash is provided, check the mempool.
  3. If no blockhash is provided but `txindex` is enabled, also check `txindex`.
- `unloadwallet` is now synchronous, meaning it will not return until the wallet is fully unloaded.
- `importmulti` now supports importing of addresses from descriptors. A “desc” parameter can be provided instead of the “scriptPubKey” in a request, as well as an optional range for ranged descriptors to specify the start and end of the range to import. Descriptors with key origin information imported through `importmulti` will have their key origin information stored in the wallet for use with creating PSBTs. More information about descriptors can be found [here](#).
- `listunspent` has been modified so that it also returns `witnessScript`, the witness script in the case of a P2WSH or P2SH-P2WSH output.
- `createwallet` now has an optional `blank` argument that can be used to create a blank wallet. Blank wallets do not have any keys or HD seed. They cannot be opened in software older than 0.18. Once a blank wallet has a HD seed set (by using `sethdseed`) or private keys, scripts, addresses, and other watch only things have been imported, the wallet is no longer blank and can be opened in 0.17.x. Encrypting a blank wallet will also set a HD seed for it.

## Deprecated or removed RPCs

- `signrawtransaction` is removed after being deprecated and hidden behind a special configuration option in version 0.17.0.
- The ‘account’ API is removed after being deprecated in v0.17. The ‘label’ API was introduced in v0.17 as a replacement for accounts. See the release notes from v0.17 for a full description of the changes from the ‘account’ API to the ‘label’ API.
- `addwitnessaddress` is removed after being deprecated in version 0.16.0.
- `generate` is deprecated and will be fully removed in a subsequent major version. This RPC is only used for testing, but its implementation reached across multiple subsystems (wallet and mining), so it is being deprecated to simplify the wallet-node interface. Projects that are using `generate` for testing purposes should transition to using the `generatetoaddress` RPC, which does not require or use the wallet component. Calling `generatetoaddress` with an address returned by the `getnewaddress` RPC gives the same functionality as the old `generate` RPC. To continue using `generate` in this version, restart bitcoind with the `-deprecatedrpc=generate` configuration option.
- Be reminded that parts of the `validateaddress` command have been deprecated and moved to `getaddressinfo`. The following deprecated fields have moved to `getaddressinfo`: `ismine`, `iswatchonly`, `script`, `hex`, `pubkeys`, `sigsrequired`, `pubkey`, `embedded`, `iscompressed`, `label`, `timestamp`, `hdkeypath`, `hdmasterkeyid`.
- The `addresses` field has been removed from the `validateaddress` and `getaddressinfo` RPC methods. This field was confusing since it referred to public keys using their P2PKH address. Clients should use the `embedded.address` field for P2SH or P2WSH wrapped addresses, and `pubkeys` for inspecting multisig participants.

## REST changes

- A new `/rest/blockhashbyheight/` endpoint is added for fetching the hash of the block in the current best blockchain based on its height (how many blocks it is after the Genesis Block).

## Graphical User Interface (GUI)

- A new Window menu is added alongside the existing File, Settings, and Help menus. Several items from the other menus that opened new windows have been moved to this new Window menu.
- In the Send tab, the checkbox for “pay only the required fee” has been removed. Instead, the user can simply decrease the value in the Custom

Feeerate field all the way down to the node's configured minimum relay fee.

- In the Overview tab, the watch-only balance will be the only balance shown if the wallet was created using the `createwallet` RPC and the `disable_private_keys` parameter was set to true.
- The launch-on-startup option is no longer available on macOS if compiled with macOS min version greater than 10.11 (use `CXXFLAGS="-mmacosx-version-min=10.11"` `CFLAGS="-mmacosx-version-min=10.11"` for setting the deployment sdk version)

## Tools

- A new `bitcoin-wallet` tool is now distributed alongside Bitcoin Core's other executables. Without needing to use any RPCs, this tool can currently create a new wallet file or display some basic information about an existing wallet, such as whether the wallet is encrypted, whether it uses an HD seed, how many transactions it contains, and how many address book entries it has.

## Planned changes

This section describes planned changes to Bitcoin Core that may affect other Bitcoin software and services.

- Since version 0.16.0, Bitcoin Core's built-in wallet has defaulted to generating P2SH-wrapped segwit addresses when users want to receive payments. These addresses are backwards compatible with all widely-used software. Starting with Bitcoin Core 0.20 (expected about a year after 0.18), Bitcoin Core will default to native segwit addresses (bech32) that provide additional fee savings and other benefits. Currently, many wallets and services already support sending to bech32 addresses, and if the Bitcoin Core project sees enough additional adoption, it will instead default to bech32 receiving addresses in Bitcoin Core 0.19 (approximately November 2019). P2SH-wrapped segwit addresses will continue to be provided if the user requests them in the GUI or by RPC, and anyone who doesn't want the update will be able to configure their default address type. (Similarly, pioneering users who want to change their default now may set the `addresstype=bech32` configuration option in any Bitcoin Core release from 0.16.0 up.)

## Deprecated P2P messages

- BIP 61 reject messages are now deprecated. Reject messages have no use case on the P2P network and are only logged for debugging by most network nodes. Furthermore, they increase bandwidth and can be harmful for privacy and security. It has been possible to disable BIP 61 messages



since v0.17 with the `-enablebip61=0` option. BIP 61 messages will be disabled by default in a future version, before being removed entirely.

## Low-level changes

This section describes RPC changes mainly useful for testing, mostly not relevant in production. The changes are mentioned for completeness.

### RPC

- The `submitblock` RPC previously returned the reason a rejected block was invalid the first time it processed that block, but returned a generic “duplicate” rejection message on subsequent occasions it processed the same block. It now always returns the fundamental reason for rejecting an invalid block and only returns “duplicate” for valid blocks it has already accepted.
- A new `submitheader` RPC allows submitting block headers independently from their block. This is likely only useful for testing.
- The `signrawtransactionwithkey` and `signrawtransactionwithwallet` RPCs have been modified so that they also optionally accept a `witnessScript`, the witness script in the case of a P2WSH or P2SH-P2WSH output. This is compatible with the change to `listunspent`.
- For the `walletprocesspsbt` and `walletcreatefundedpsbt` RPCs, if the `bip32derivs` parameter is set to true but the key metadata for a public key has not been updated yet, then that key will have a derivation path as if it were just an independent key (i.e. no derivation path and its master fingerprint is itself).

### Configuration

- The `-usehd` configuration option was removed in version 0.16. From that version onwards, all new wallets created are hierarchical deterministic wallets. This release makes specifying `-usehd` an invalid configuration option.

### Network

- This release allows peers that your node automatically disconnected for misbehavior (e.g. sending invalid data) to reconnect to your node if you have unused incoming connection slots. If your slots fill up, a misbehaving node will be disconnected to make room for nodes without a history of problems (unless the misbehaving node helps your node in some other way, such as by connecting to a part of the Internet from which you don’t have many other peers). Previously, Bitcoin Core banned the IP addresses of

misbehaving peers for a period of time (default of 1 day); this was easily circumvented by attackers with multiple IP addresses. If you manually ban a peer, such as by using the `setban` RPC, all connections from that peer will still be rejected.

## Wallet

- The key metadata will need to be upgraded the first time that the HD seed is available. For unencrypted wallets this will occur on wallet loading. For encrypted wallets this will occur the first time the wallet is unlocked.
- Newly encrypted wallets will no longer require restarting the software. Instead such wallets will be completely unloaded and reloaded to achieve the same effect.
- A sub-project of Bitcoin Core now provides Hardware Wallet Interaction (HWI) scripts that allow command-line users to use several popular hardware key management devices with Bitcoin Core. See their project page for details.

## Security

- This release changes the Random Number Generator (RNG) used from OpenSSL to Bitcoin Core's own implementation, although entropy gathered by Bitcoin Core is fed out to OpenSSL and then read back in when the program needs strong randomness. This moves Bitcoin Core a little closer to no longer needing to depend on OpenSSL, a dependency that has caused security issues in the past. The new implementation gathers entropy from multiple sources, including from hardware supporting the `rdseed` CPU instruction.

## Changes for particular platforms

- On macOS, Bitcoin Core now opts out of application CPU throttling (“app nap”) during initial blockchain download, when catching up from over 100 blocks behind the current chain tip, or when reindexing chain data. This helps prevent these operations from taking an excessively long time because the operating system is attempting to conserve power.

## 0.18.0 change log

### Consensus

- #14247 Fix crash bug with duplicate inputs within a transaction (TheBlueMatt)

## Mining

- #14811 Mining: Enforce that segwit option must be set in GBT (jnewbery)

## Block and transaction handling

- #13310 Report progress in ReplayBlocks while rolling forward (promag)
- #13783 validation: Pass tx pool reference into CheckSequenceLocks (MarcoFalke)
- #14834 validation: Assert that pindexPrev is non-null when required (kallewoof)
- #14085 index: Fix for indexers skipping genesis block (jimpo)
- #14963 mempool, validation: Explain `cs_main` locking semantics (MarcoFalke)
- #15193 Default `-whitelistforcerelay` to off (sdaftuar)
- #15429 Update `assumevalid`, `minimumchainwork`, and `getchaintxstats` to height 563378 (gmaxwell)
- #15552 Granular `invalidateblock` and `RewindBlockIndex` (MarcoFalke)
- #14841 Move `CheckBlock()` call to critical section (hebasto)

## P2P protocol and network code

- #14025 Remove dead code for `nVersion=10300` (MarcoFalke)
- #12254 BIP 158: Compact Block Filters for Light Clients (jimpo)
- #14073 blockfilter: Avoid out-of-bounds script access (jimpo)
- #14140 Switch `nPrevNodeCount` to `vNodesSize` (pstratem)
- #14027 Skip stale tip checking if outbound connections are off or if reindexing (gmaxwell)
- #14532 Never bind `INADDR_ANY` by default, and warn when doing so explicitly (luke-jr)
- #14733 Make peer timeout configurable, speed up very slow test and ensure correct code path tested (zallarak)
- #14336 Implement poll (pstratem)
- #15051 `IsReachable` is the inverse of `IsLimited` (DRY). Includes unit tests (mmachicao)
- #15138 Drop `IsLimited` in favor of `IsReachable` (Empact)
- #14605 Return of the Banman (dongcarl)
- #14970 Add `dnsseed.emzy.de` to DNS seeds (Emzy)
- #14929 Allow connections from misbehavior banned peers (gmaxwell)
- #15345 Correct comparison of `addr count` (dongcarl)
- #15201 Add missing locking annotation for `vNodes`. `vNodes` is guarded by `cs_vNodes` (practicalswift)
- #14626 Select orphan transaction uniformly for eviction (sipa)
- #15486 Ensure tried collisions resolve, and allow feeler connections to existing outbound netgroups (sdaftuar)

## Wallet

- #13962 Remove unused `dummy_tx` variable from `FillPSBT` (dongcarl)
- #13967 Don't report `minversion` wallet entry as unknown (instagibbs)
- #13988 Add checks for `settxfee` reasonableness (ajtowns)
- #12559 Avoid locking `cs_main` in some wallet RPC (promag)
- #13631 Add `CMerkleTx::IsImmatureCoinBase` method (Empact)
- #14023 Remove accounts RPCs (jnewbery)
- #13825 Kill accounts (jnewbery)
- #10605 Add `AssertLockHeld` assertions in `CWallet::ListCoins` (ryanofsky)
- #12490 Remove deprecated wallet rpc features from `bitcoin_server` (jnewbery)
- #14138 Set `encrypted_batch` to nullptr after delete. Avoid double free in the case of `NDEBUG` (practicalswift)
- #14168 Remove `ENABLE_WALLET` from `libbitcoin_server.a` (jnewbery)
- #12493 Reopen `CDBEnv` after encryption instead of shutting down (achow101)
- #14282 Remove `-usehd` option (jnewbery)
- #14146 Remove trailing separators from `-walletdir` arg (PierreRochard)
- #14291 Add `ListWalletDir` utility function (promag)
- #14468 Deprecate `generate` RPC method (jnewbery)
- #11634 Add missing `cs_wallet/cs_KeyStore` locks to wallet (practicalswift)
- #14296 Remove `addwitnessaddress` (jnewbery)
- #14451 Add BIP70 deprecation warning and allow building GUI without BIP70 support (jameshilliard)
- #14320 Fix duplicate fileid detection (ken2812221)
- #14561 Remove `fs::relative` call and fix `listwalletdir` tests (promag)
- #14454 Add SegWit support to `importmulti` (MeshCollider)
- #14410 `rpcwallet`: `ischange` field for `getaddressinfo` RPC (mrwhythat)
- #14350 Add `WalletLocation` class (promag)
- #14689 Require a public key to be retrieved when signing a P2PKH input (achow101)
- #14478 Show error to user when corrupt wallet unlock fails (MeshCollider)
- #14411 Restore ability to list incoming transactions by label (ryanofsky)
- #14552 Detect duplicate wallet by comparing the db filename (ken2812221)
- #14678 Remove redundant `KeyOriginInfo` access, already done in `CreateSig` (instagibbs)
- #14477 Add ability to convert solvability info to descriptor (sipa)
- #14380 Fix assert crash when specified change output spend size is unknown (instagibbs)
- #14760 Log env path in `BerkeleyEnvironment::Flush` (promag)
- #14646 Add expansion cache functions to descriptors (unused for now) (sipa)
- #13076 Fix `ScanForWalletTransactions` to return an enum indicating scan result: `success` / `failure` / `user_abort` (Empact)

- #14821 Replace CAffectedKeysVisitor with descriptor based logic (sipa)
- #14957 Initialize `stop_block` in `CWallet::ScanForWalletTransactions` (Empact)
- #14565 Overhaul `importmulti` logic (sipa)
- #15039 Avoid leaking `nLockTime` fingerprint when anti-fee-sniping (MarcoFalke)
- #14268 Introduce `SafeDbt` to handle `Dbt` with free or `memory_cleanse` raii-style (Empact)
- #14711 Remove uses of `chainActive` and `mapBlockIndex` in wallet code (ryanofsky)
- #15279 Clarify `rescanblockchain` doc (MarcoFalke)
- #15292 Remove `boost::optional`-related false positive `-Wmaybe-uninitialized` warnings on GCC compiler (hebasto)
- #13926 Tools `bitcoin-wallet` - a tool for creating and managing wallets offline (jnewbery)
- #11911 Free `BerkeleyEnvironment` instances when not in use (ryanofsky)
- #15235 Do not import private keys to wallets with private keys disabled (achow101)
- #15263 Descriptor expansions only need pubkey entries for PKH/WPKH (sipa)
- #15322 Add missing `cs_db` lock (promag)
- #15297 Releases dangling files on `BerkeleyEnvironment::Close` (promag)
- #14491 Allow descriptor imports with `importmulti` (MeshCollider)
- #15365 Add lock annotation for `mapAddressBook` (MarcoFalke)
- #15226 Allow creating blank (empty) wallets (alternative) (achow101)
- #15390 [wallet-tool] Close `bdb` when flushing wallet (jnewbery)
- #15334 Log absolute paths for the wallets (hebasto)
- #14978 Factor out PSBT utilities from RPCs for use in GUI code; related refactoring (gwillen)
- #14481 Add P2SH-P2WSH support to `listunspent` RPC (MeshCollider)
- #14021 Import key origin data through descriptors in `importmulti` (achow101)
- #14075 Import watch only pubkeys to the keypool if private keys are disabled (achow101)
- #15368 Descriptor checksums (sipa)
- #15433 Use a single wallet batch for `UpgradeKeyMetadata` (jonasschnelli)
- #15408 Remove unused `TransactionError` constants (MarcoFalke)
- #15583 Log and ignore errors in `ListWalletDir` and `IsBerkeleyBtree` (promag)
- #14195 Pass `privkey` export DER compression flag correctly (fingera)
- #15299 Fix assertion in `CKey::SignCompact` (promag)
- #14437 Start to separate wallet from node (ryanofsky)
- #15749 Fix: `importmulti` only imports origin info for PKH outputs (sipa)

## RPC and other APIs

- #12842 Prevent concurrent `savemempool` (promag)
- #13987 Report `minfeefilter` value in `getpeerinfo` RPC (ajtowns)
- #13891 Remove `getinfo` deprecation warning (jnewbery)
- #13399 Add `submitheader` (MarcoFalke)
- #12676 Show `bip125-replaceable` flag, when retrieving mempool entries (dexX7)
- #13723 PSBT key path cleanups (sipa)
- #14008 Preserve a format of RPC command definitions (kostyantyn)
- #9332 Let wallet `importmulti` RPC accept labels for standard scriptPubKeys (ryanofsky)
- #13983 Return more specific reject reason for `submitblock` (MarcoFalke)
- #13152 Add `getnodeaddresses` RPC command (chris-belcher)
- #14298 rest: Improve performance for JSON calls (alecalve)
- #14297 Remove warning for removed `estimatefee` RPC (jnewbery)
- #14373 Consistency fixes for RPC descriptions (ch4ot1c)
- #14150 Add key origin support to descriptors (sipa)
- #14518 Always throw in `getblockstats` if `-txindex` is required (promag)
- #14060 ZMQ: add options to configure outbound message high water mark, aka `SNDHWM` (mruddy)
- #13381 Add possibility to preserve labels on `importprivkey` (marcoagner)
- #14530 Use `RPCHelpMan` to generate RPC doc strings (MarcoFalke)
- #14720 Correctly name RPC arguments (MarcoFalke)
- #14726 Use `RPCHelpMan` for all RPCs (MarcoFalke)
- #14796 Pass argument descriptions to `RPCHelpMan` (MarcoFalke)
- #14670 http: Fix HTTP server shutdown (promag)
- #14885 Assert that named arguments are unique in `RPCHelpMan` (promag)
- #14877 Document default values for optional arguments (MarcoFalke)
- #14875 `RPCHelpMan`: Support required arguments after optional ones (MarcoFalke)
- #14993 Fix data race (UB) in `InterruptRPC()` (practicalswift)
- #14653 `rpcwallet`: Add missing transaction categories to RPC helptexts (andrewtoth)
- #14981 Clarify RPC `getrawtransaction`'s time help text (benthecarman)
- #12151 Remove `cs_main` lock from `blockToJSON` and `blockheaderToJSON` (promag)
- #15078 Document `bytessent_per_msg` and `bytesrecv_per_msg` (MarcoFalke)
- #15057 Correct `reconsiderblockhelp` text, add test (MarcoFalke)
- #12153 Avoid permanent `cs_main` lock in `getblockheader` (promag)
- #14982 Add `getrpcinfo` command (promag)
- #15122 Expand help text for `importmulti` changes (jnewbery)
- #15186 remove duplicate solvable field from `getaddressinfo` (fanquake)
- #15209 zmq: log outbound message high water mark when reusing socket (fanquake)

- #15177 rest: Improve tests and documentation of /headers and /block (promag)
- #14353 rest: Add blockhash call, fetch blockhash by height (jonasschnelli)
- #15248 Compile on GCC4.8 (MarcoFalke)
- #14987 RPCHelpMan: Pass through Result and Examples (MarcoFalke)
- #15159 Remove lookup to UTXO set from GetTransaction (amitiuttarwar)
- #15245 remove deprecated mentions of signrawtransaction from fundraw help (instagibbs)
- #14667 Add `deriveaddresses` RPC util method (Sjors)
- #15357 Don't ignore `-maxtxfee` when wallet is disabled (JBaczuk)
- #15337 Fix for segfault if combinepsbt called with empty inputs (benthecarman)
- #14918 RPCHelpMan: Check default values are given at compile-time (MarcoFalke)
- #15383 mining: Omit uninitialized currentblockweight, currentblocktx (MarcoFalke)
- #13932 Additional utility RPCs for PSBT (achow101)
- #15401 Actually throw help when passed invalid number of params (MarcoFalke)
- #15471 rpc/gui: Remove 'Unknown block versions being mined' warning (laanwj)
- #15497 Consistent range arguments in scantxoutset/importmulti/deriveaddresses (sipa)
- #15510 deriveaddresses: add range to CRPCConvertParam (Sjors)
- #15582 Fix overflow bug in analyzepsbt fee: CAmount instead of int (sipa)
- #13424 Consistently validate txid / blockhash length and encoding in rpc calls (Empact)
- #15750 Remove the addresses field from the getaddressinfo return object (jnewbery)

## GUI

- #13634 Compile `boost::signals2` only once (MarcoFalke)
- #13248 Make proxy icon from statusbar clickable (mess110)
- #12818 TransactionView: highlight replacement tx after fee bump (Sjors)
- #13529 Use new Qt5 connect syntax (promag)
- #14162 Also log and print messages or questions like bitcoind (MarcoFalke)
- #14385 Avoid system harfbuzz and bz2 (theuni)
- #14450 Fix QCompleter popup regression (hebasto)
- #14177 Set C locale for amountWidget (hebasto)
- #14374 Add `Blocksdire` to Debug window (hebasto)
- #14554 Remove unused `adjustedTime` parameter (hebasto)
- #14228 Enable system tray icon by default if available (hebasto)
- #14608 Remove the "Pay only required fee..." checkbox (hebasto)
- #14521 qt, docs: Fix `bitcoin-qt -version` output formatting (hebasto)
- #13966 When private key is disabled, only show watch-only balance

(ken2812221)

- #14828 Remove hidden columns in coin control dialog (promag)
- #14783 Fix `boost::signals2::no_slots_error` in early calls to `InitWarning` (promag)
- #14854 Cleanup `SplashScreen` class (hebasto)
- #14801 Use `window()` instead of obsolete `topLevelWidget()` (hebasto)
- #14573 Add Window menu (promag)
- #14979 Restore < Qt5.6 compatibility for `addAction` (jonasschnelli)
- #14975 Refactoring with `QString::toNsString()` (hebasto)
- #15000 Fix broken notificador on GNOME (hebasto)
- #14375 Correct misleading “overridden options” label (hebasto)
- #15007 Notificador class refactoring (hebasto)
- #14784 Use `WalletModel*` instead of the wallet name as map key (promag)
- #11625 Add `BitcoinApplication` & `RPCCConsole` tests (ryanofsky)
- #14517 Fix start with the `-min` option (hebasto)
- #13216 implements concept for different disk sizes on intro (marcoagner)
- #15114 Replace remaining 0 with `nullptr` (Empact)
- #14594 Fix minimized window bug on Linux (hebasto)
- #14556 Fix confirmed transaction labeled “open” (#13299) (hebasto)
- #15149 Show current wallet name in window title (promag)
- #15136 “Peers” tab overhaul (hebasto)
- #14250 Remove redundant `stopThread()` and `stopExecutor()` signals (hebasto)
- #15040 Add workaround for `QProgressDialog` bug on macOS (hebasto)
- #15101 Add `WalletController` (promag)
- #15178 Improve “help-console” message (hebasto)
- #15210 Fix window title update (promag)
- #15167 Fix wallet selector size adjustment (hebasto)
- #15208 Remove macOS launch-at-startup when compiled with > macOS 10.11, fix memory mismanagement (jonasschnelli)
- #15163 Correct units for “-dbcache” and “-prune” (hebasto)
- #15225 Change the receive button to respond to keypool state changing (achow101)
- #15280 Fix shutdown order (promag)
- #15203 Fix issue #9683 “gui, wallet: random abort (segmentation fault) (dooglus)
- #15091 Fix model overlay header sync (jonasschnelli)
- #15153 Add Open Wallet menu (promag)
- #15183 Fix `m_assumed_blockchain_size` variable value (marcoagner)
- #15063 If BIP70 is disabled, attempt to fall back to BIP21 parsing (luke-jr)
- #15195 Add Close Wallet action (promag)
- #15462 Fix async open wallet call order (promag)
- #15801 Bugfix: GUI: Options: Initialise prune setting range before loading current value, and remove upper bound limit (luke-jr)



## Build system

- #13955 gitian: Bump descriptors for (0.)18 (fanquake)
- #13899 Enable -Wredundant-decls where available. Remove redundant redeclarations (practicalswift)
- #13665 Add RISC-V support to gitian (ken2812221)
- #14062 Generate MSVC project files via python script (ken2812221)
- #14037 Add README.md to linux release tarballs (hebasto)
- #14183 Remove unused Qt 4 dependencies (ken2812221)
- #14127 Avoid getifaddrs when unavailable (greenaddress)
- #14184 Scripts and tools: increased timeout downloading (cisba)
- #14204 Move `interfaces/*` to `libbitcoin_server` (laanwj)
- #14208 Actually remove `ENABLE_WALLET` (jnewbery)
- #14212 Remove libssl from LDADD unless GUI (MarcoFalke)
- #13578 Upgrade zeromq to 4.2.5 and avoid deprecated zeromq API functions (mruddy)
- #14281 lcov: filter `/usr/lib/` from coverage reports (MarcoFalke)
- #14325 gitian: Use versioned unsigned tarballs instead of generically named ones (achow101)
- #14253 During ‘make clean’, remove some files that are currently missed (murrayn)
- #14455 Unbreak `make clean` (jamesob)
- #14495 Warn (don’t fail!) on spelling errors (practicalswift)
- #14496 Pin to specific versions of Python packages we install from PyPI in Travis (practicalswift)
- #14568 Fix Qt link order for Windows build (ken2812221)
- #14252 Run functional tests and benchmarks under the undefined behaviour sanitizer (UBSan) (practicalswift)
- #14612 Include full version number in released file names (achow101)
- #14840 Remove duplicate libconsensus linking in test make (AmirAbrams)
- #14564 Adjust configure so that only BIP70 is disabled when protobuf is missing instead of the GUI (jameshilliard)
- #14883 Add `--retry 5` to curl opts in `install_db4.sh` (qubenix)
- #14701 Add `CLIENT_VERSION_BUILD` to `CFBundleGetInfoString` (fanquake)
- #14849 Qt 5.9.7 (fanquake)
- #15020 Add names to Travis jobs (gkrizek)
- #15047 Allow to configure `-with-sanitizers=fuzzer` (MarcoFalke)
- #15154 Configure: bitcoin-tx doesn’t need libevent, so don’t pull it in (luke-jr)
- #15175 Drop macports support (Empact)
- #15308 Restore compatibility with older boost (Empact)
- #15407 msvc: Fix silent merge conflict between #13926 and #14372 part II (ken2812221)
- #15388 Makefile.am: add rule for `src/bitcoin-wallet` (Sjors)
- #15393 Bump minimum Qt version to 5.5.1 (Sjors)

- #15285 Prefer Python 3.4 even if newer versions are present on the system (Sjors)
- #15398 msvc: Add rapidcheck property tests (ken2812221)
- #15431 msvc: scripted-diff: Remove NDEBUG pre-define in project file (ken2812221)
- #15549 gitian: Improve error handling (laanwj)
- #15548 use full version string in setup.exe (MarcoFalke)
- #11526 Visual Studio build configuration for Bitcoin Core (sipsorcery)
- #15110 build\_msvc: Fix the build problem in `libbitcoin_server` (Mr-Leshiy)
- #14372 msvc: build secp256k1 and leveldb locally (ken2812221)
- #15325 msvc: Fix silent merge conflict between #13926 and #14372 (ken2812221)
- #15391 Add compile time verification of assumptions we're currently making implicitly/tacitly (practicalswift)
- #15503 msvc: Use a single file to specify the include path (ken2812221)
- #13765 contrib: Add gitian build support for github pull request (ken2812221)
- #15809 gitignore: plist and dat (jamesob)

## Tests and QA

- #15405 appveyor: Clean cache when build configuration changes (Sjors)
- #13953 Fix deprecation in `bitcoin-util-test.py` (isghe)
- #13963 Replace usage of `tostring()` with `tobytes()` (dongcarl)
- #13964 ci: Add appveyor ci (ken2812221)
- #13997 appveyor: fetch the latest port data (ken2812221)
- #13707 Add usage note to `check-rpc-mappings.py` (masonicboom)
- #14036 travis: Run unit tests `-with-sanitizers=undefined` (MarcoFalke)
- #13861 Add testing of `value_ret` for `SelectCoinsBnB` (Empact)
- #13863 travis: Move script sections to files in `.travis/` subject to shellcheck (scravy)
- #14081 travis: Fix missing differentiation between unit and functional tests (scravy)
- #14042 travis: Add `cxxflags=-wno-psabi` at arm job (ken2812221)
- #14051 Make `combine_logs.py` handle multi-line logs (jnewbery)
- #14093 Fix accidental truncation from `int` to `bool` (practicalswift)
- #14108 Add missing locking annotations and locks (`g_cs_orphans`) (practicalswift)
- #14088 Don't `assert(...)` with side effects (practicalswift)
- #14086 appveyor: Use `clcache` to speed up build (ken2812221)
- #13954 Warn (don't fail!) on spelling errors. Fix typos reported by codespell (practicalswift)
- #12775 Integration of property based testing into Bitcoin Core (Christewart)
- #14119 Read reject reasons from debug log, not P2P messages (Marco-

Falke)

- #14189 Fix silent merge conflict in `wallet_importmulti` (MarcoFalke)
- #13419 Speed up `knapsack_solver_test` by not recreating wallet 100 times (lucash-dev)
- #14199 Remove redundant BIP174 test from `rpc_psbt.json` (arasplitzu)
- #14179 Fixups to “Run all tests even if wallet is not compiled” (MarcoFalke)
- #14225 Reorder tests and move most of extended tests up to normal tests (ken2812221)
- #14236 `generate` -> `generatetoaddress` change to allow tests run without wallet (sanket1729)
- #14287 Use `MakeUnique` to construct objects owned by `unique_ptrs` (practicalswift)
- #14007 Run functional test on Windows and enable it on Appveyor (ken2812221)
- #14275 Write the notification message to different files to avoid race condition in `feature_notifications.py` (ken2812221)
- #14306 appveyor: Move AppVeyor YAML to dot-file-style YAML (Mitchell-Cash)
- #14305 Enforce critical class instance attributes in functional tests, fix segwit test specificity (JustinTArthur)
- #12246 Bugfix: Only run bitcoin-tx tests when bitcoin-tx is enabled (luke-jr)
- #14316 Exclude all tests with difference parameters in `--exclude` list (ken2812221)
- #14381 Add missing call to `skip_if_no_cli()` (practicalswift)
- #14389 travis: Set codespell version to avoid breakage (MarcoFalke)
- #14398 Don’t access out of bounds array index: `array[sizeof(array)]` (Empact)
- #14419 Remove `rpc_zmq.py` (jnewbery)
- #14241 appveyor: Script improvement (ken2812221)
- #14413 Allow closed RPC handler in `assert_start_raises_init_error` (ken2812221)
- #14324 Run more tests with wallet disabled (MarcoFalke)
- #13649 Allow arguments to be forwarded to flake8 in `lint-python.sh` (jamesob)
- #14465 Stop node before removing the notification file (ken2812221)
- #14460 Improve ‘CAmount’ tests (hebasto)
- #14456 forward timeouts properly in `send_blocks_and_test` (jamesob)
- #14527 Revert “Make qt wallet test compatible with qt4” (MarcoFalke)
- #14504 Show the progress of functional tests (isghe)
- #14559 appveyor: Enable multiwallet tests (ken2812221)
- #13515 travis: Enable qt for all jobs (ken2812221)
- #14571 Test that nodes respond to `getdata` with `notfound` (MarcoFalke)
- #14569 Print dots by default in functional tests (ken2812221)
- #14631 Move deterministic address import to `setup_nodes` (jnewbery)

- #14630 test: Remove travis specific code (MarcoFalke)
- #14528 travis: Compile once on xenial (MarcoFalke)
- #14092 Dry run `bench_bitcoin` as part `make check` to allow for quick identification of assertion/sanitizer failures in benchmarking code (practicalswift)
- #14664 `example_test.py`: fixup coinbase height argument, derive number clearly (instagibbs)
- #14522 Add invalid P2P message tests (jamesob)
- #14619 Fix value display name in `test_runner` help text (merland)
- #14672 Send fewer spam messages in `p2p_invalid_messages` (jamesob)
- #14673 travis: Fail the ubsan travis build in case of newly introduced ubsan errors (practicalswift)
- #14665 appveyor: Script improvement part II (ken2812221)
- #14365 Add Python dead code linter (vulture) to Travis (practicalswift)
- #14693 `test_node`: `get_mem_rss` fixups (MarcoFalke)
- #14714 `util.h`: explicitly include required QString header (11l1)
- #14705 travis: Avoid timeout on verify-commits check (MarcoFalke)
- #14770 travis: Do not specify `sudo` in `.travis` (scravy)
- #14719 Check specific reject reasons in `feature_block` (MarcoFalke)
- #14771 Add `BOOST_REQUIRE` to getters returning optional (MarcoFalke)
- #14777 Add regtest for JSON-RPC batch calls (domob1812)
- #14764 travis: Run thread sanitizer on unit tests (MarcoFalke)
- #14400 Add Benchmark to test input de-duplication worst case (JeremyRubin)
- #14812 Fix `p2p_invalid_messages` on macOS (jamesob)
- #14813 Add `wallet_encryption` error tests (MarcoFalke)
- #14820 Fix `descriptor_tests` not checking ToString output of public descriptors (ryanofsky)
- #14794 Add AddressSanitizer (ASan) Travis build (practicalswift)
- #14819 Bugfix: `test/functional/mempool_accept`: Ensure oversize transaction is actually oversize (luke-jr)
- #14822 bench: Destroy wallet txs instead of leaking their memory (MarcoFalke)
- #14683 Better `combine_logs.py` behavior (jamesob)
- #14231 travis: Save cache even when build or test fail (ken2812221)
- #14816 Add CScriptNum decode python implementation in functional suite (instagibbs)
- #14861 Modify `rpc_bind` to conform to #14532 behaviour (dongcarl)
- #14864 Run scripted-diff in subshell (dongcarl)
- #14795 Allow `test_runner` command line to receive parameters for each test (marcoagner)
- #14788 Possible fix the permission error when the tests open the cookie file (ken2812221)
- #14857 `wallet_keypool_topup.py`: Test for all keypool address types (instagibbs)
- #14886 Refactor importmulti tests (jnewbery)

- #14908 Removed implicit CTransaction constructor calls from tests and benchmarks (lucash-dev)
- #14903 Handle ImportError explicitly, improve comparisons against None (daniel-s-ingram)
- #14884 travis: Enforce python 3.4 support through linter (Sjors)
- #14940 Add test for truncated pushdata script (MarcoFalke)
- #14926 consensus: Check that final transactions are valid (MarcoFalke)
- #14937 travis: Fix travis would always be green even if it fail (ken2812221)
- #14953 Make `g_insecure_rand_ctx` `thread_local` (MarcoFalke)
- #14931 mempool: Verify prioritization is dumped correctly (MarcoFalke)
- #14935 Test for expected return values when calling functions returning a success code (practicalswift)
- #14969 Fix `cuckoocache_tests` TSAN failure introduced in 14935 (practicalswift)
- #14964 Fix race in `mempool_accept` (MarcoFalke)
- #14829 travis: Enable functional tests in the threadsanitizer (tsan) build job (practicalswift)
- #14985 Remove `thread_local` from `test_bitcoin` (MarcoFalke)
- #15005 Bump timeout to run tests in travis thread sanitizer (MarcoFalke)
- #15013 Avoid race in `p2p_timeouts` (MarcoFalke)
- #14960 lint/format-strings: Correctly exclude escaped percent symbols (luke-jr)
- #14930 pruning: Check that `verifychain` can be called when pruned (MarcoFalke)
- #15022 Upgrade Travis OS to Xenial (gkrizek)
- #14738 Fix running `wallet_listtransactions.py` individually through `test_runner.py` (kristapsk)
- #15026 Rename `rpc_timewait` to `rpc_timeout` (MarcoFalke)
- #15069 Fix `rpc_net.py` pong race condition (Empact)
- #14790 Allow running `rpc_bind.py` `-nonloopback` test without IPv6 (kristapsk)
- #14457 add invalid tx templates for use in functional tests (jamesob)
- #14855 Correct ineffectual `WithOrVersion` from `transactions_tests` (Empact)
- #15099 Use `std::vector` API for construction of test data (domob1812)
- #15102 Run `invalid_txs.InputMissing` test in `feature_block` (MarcoFalke)
- #15059 Add basic test for BIP34 (MarcoFalke)
- #15108 Tidy up `wallet_importmulti.py` (amitiuttarwar)
- #15164 Ignore shellcheck warning SC2236 (promag)
- #15170 refactor/lint: Add ignored shellcheck suggestions to an array (koalaman)
- #14958 Remove race between connecting and shutdown on separate connections (promag)
- #15166 Pin shellcheck version (practicalswift)
- #15196 Update all `subprocess.check_output` functions to be Python 3.4

compatible (gkrizek)

- #15043 Build fuzz targets into separate executables (MarcoFalke)
- #15276 travis: Compile once on trusty (MarcoFalke)
- #15246 Add tests for invalid message headers (MarcoFalke)
- #15301 When testing with `-usecli`, unify RPC arg to cli arg conversion and handle dicts and lists (achow101)
- #15247 Use wallet to retrieve raw transactions (MarcoFalke)
- #15303 travis: Remove unused `functional_tests_config` (MarcoFalke)
- #15330 Fix race in `p2p_invalid_messages` (MarcoFalke)
- #15324 Make bloom tests deterministic (MarcoFalke)
- #15328 travis: Revert “run extended tests once daily” (MarcoFalke)
- #15327 Make test `updatecoins_simulation_test` deterministic (practicalswift)
- #14519 add utility to easily profile node performance with perf (jamesob)
- #15349 travis: Only exit early if compilation took longer than 30 min (MarcoFalke)
- #15350 Drop RPC connection if `-usecli` (promag)
- #15370 test: Remove unused `-force` option (MarcoFalke)
- #14543 minor `p2p_sendheaders` fix of height in coinbase (instagibbs)
- #13787 Test for Windows encoding issue (ken2812221)
- #15378 Added missing tests for RPC wallet errors (benthecarman)
- #15238 remove some magic mining constants in functional tests (instagibbs)
- #15411 travis: Combine `-disable-bip70` into existing job (MarcoFalke)
- #15295 fuzz: Add `test/fuzz/test_runner.py` and run it in travis (MarcoFalke)
- #15413 Add missing `cs_main` locks required when accessing `pcoinsdbview`, `pcoinsTip` or `pblocktree` (practicalswift)
- #15399 fuzz: Script validation flags (MarcoFalke)
- #15410 txindex: interrupt threadGroup before calling destructor (MarcoFalke)
- #15397 Remove manual byte editing in `wallet_tx_clone` func test (instagibbs)
- #15415 functional: allow custom cwd, use `tmpdir` as default (Sjors)
- #15404 Remove `-txindex` to start nodes (amitiuttarwar)
- #15439 remove `byte.hex()` to keep compatibility (AkioNak)
- #15419 Always refresh cache to be out of ibd (MarcoFalke)
- #15507 Bump timeout on tests that timeout on windows (MarcoFalke)
- #15506 appveyor: fix cache issue and reduce dependencies build time (ken2812221)
- #15485 add `rpc_misc.py`, mv test `getmemoryinfo`, add test `mallocinfo` (adamjonas)
- #15321 Add `cs_main` lock annotations for `mapBlockIndex` (MarcoFalke)
- #14128 lint: Make sure we read the command line inputs using UTF-8 decoding in python (ken2812221)
- #14115 lint: Make all linters work under the default macos dev environment

(build-osx.md) (practicalswift)

- #15219 lint: Enable python linters via an array (Empact)

## Platform support

- #13866 utils: Use `_w fopen` and `_w freopen` on windows (ken2812221)
- #13886 utils: Run commands using UTF-8 string on windows (ken2812221)
- #14192 utils: Convert `fs::filesystem_error` messages from local multi-byte to UTF-8 on windows (ken2812221)
- #13877 utils: Make `fs::path::string()` always return UTF-8 string on windows (ken2812221)
- #13883 utils: Convert windows args to UTF-8 string (ken2812221)
- #13878 utils: Add `fstream` wrapper to allow to pass unicode filename on windows (ken2812221)
- #14426 utils: Fix broken windows filelock (ken2812221)
- #14686 Fix windows build error if `--disable-bip70` (ken2812221)
- #14922 windows: Set `_WIN32_WINNT` to 0x0601 (Windows 7) (ken2812221)
- #13888 Call unicode API on Windows (ken2812221)
- #15468 Use `fsbridge::ifstream` to fix Windows path issue (ken2812221)
- #13734 Drop `boost::scoped_array` and use `wchar_t` API explicitly on Windows (ken2812221)
- #13884 Enable bdb unicode support for Windows (ken2812221)

## Miscellaneous

- #13935 contrib: Adjust output to current test format (AkioNak)
- #14097 validation: Log `FormatStateMessage` on `ConnectBlock` error in `ConnectTip` (MarcoFalke)
- #13724 contrib: Support ARM and RISC-V symbol check (ken2812221)
- #13159 Don't close old debug log file handle prematurely when trying to re-open (on `SIGHUP`) (practicalswift)
- #14186 bitcoin-cli: don't translate command line options (HashUnlimited)
- #14057 logging: Only log using `config file path_to_bitcoin.conf` message on startup if conf file exists (leishman)
- #14164 Update univalue subtree (MarcoFalke)
- #14272 init: Remove deprecated args from hidden args (MarcoFalke)
- #14494 Error if `#` is used in `rpcpassword` in conf (MeshCollider)
- #14742 Properly generate salt in `rpcauth.py` (dongcarl)
- #14708 Warn unrecognised sections in the config file (AkioNak)
- #14756 Improve `rpcauth.py` by using `argparse` and `getpass` modules (promag)
- #14785 scripts: Fix detection of copyright holders (cornelius)
- #14831 scripts: Use `#!/usr/bin/env bash` instead of `#!/bin/bash` (vim88)
- #14869 Scripts: Add trusted key for samuel dobson (laanwj)
- #14809 Tools: improve `verify-commits.py` script (jlopp)

- #14624 Some simple improvements to the RNG code (sipa)
- #14947 scripts: Remove python 2 import workarounds (practicalswift)
- #15087 Error if rpcpassword contains hash in conf sections (MeshCollider)
- #14433 Add checksum in gitian build scripts for ossl (TheCharlatan)
- #15165 contrib: Allow use of github api authentication in github-merge (laanwj)
- #14409 utils and libraries: Make 'blocksdir' always net specific (hebasto)
- #14839 threads: Fix uninitialized members in `sched_param` (fanquake)
- #14955 Switch all RNG code to the built-in PRNG (sipa)
- #15258 Scripts and tools: Fix `devtools/copyright_header.py` to always honor exclusions (Empact)
- #12255 Update bitcoin.service to conform to init.md (dongcarl)
- #15266 memory: Construct globals on first use (MarcoFalke)
- #15347 Fix build after pr 15266 merged (hebasto)
- #15351 Update linearize-hashes.py (OverlordQ)
- #15358 util: Add `setuptools_helpers()` (MarcoFalke)
- #15216 Scripts and tools: Replace script name with a special parameter (hebasto)
- #15250 Use RdSeed when available, and reduce RdRand load (sipa)
- #15278 Improve PID file error handling (hebasto)
- #15270 Pull leveldb subtree (MarcoFalke)
- #15456 Enable PID file creation on WIN (riordant)
- #12783 macOS: disable AppNap during sync (krab)
- #13910 Log progress while verifying blocks at level 4 (domob1812)
- #15124 Fail AppInitMain if either disk space check fails (Empact)
- #15117 Fix invalid memory write in case of failing `mmap(...)` in `PosixLockedPageAllocator::AllocateLocked` (practicalswift)
- #14357 streams: Fix broken `streams_vector_reader` test. Remove unused `seek(size_t)`
- #11640 Make `LOCK`, `LOCK2`, `TRY_LOCK` work with `CWaitableCriticalSection` (ryanofsky)
- #14074 Use `std::unordered_set` instead of `set` in blockfilter interface (jimpo)
- #15275 Add gitian PGP key for hebasto (hebasto)

## Documentation

- #14120 Notes about control port and read access to cookie (JBaczuk)
- #14135 correct GetDifficulty doc after #13288 (fanquake)
- #14013 Add new regtest ports in man following #10825 ports reattributions (ariard)
- #14149 Remove misleading checkpoints comment in CMainParams (MarcoFalke)
- #14153 Add disable-wallet section to OSX build instructions, update line in Unix instructions (bitstein)
- #13662 Explain when reindex-chainstate can be used instead of reindex



(Sjors)

- #14207 `-help-debug` implies `-help` (laanwj)
- #14213 Fix reference to `lint-locale-dependence.sh` (hebasto)
- #14206 Document `-checklevel` levels (laanwj)
- #14217 Add GitHub PR template (MarcoFalke)
- #14331 doxygen: Fix member comments (MarcoFalke)
- #14264 Split depends installation instructions per arch (MarcoFalke)
- #14393 Add missing `apt-get install` (poiuty)
- #14428 Fix macOS files description in `qt/README.md` (hebasto)
- #14390 release process: RPC documentation (karel-3d)
- #14472 `getblocktemplate`: use `SegWit` in example (Sjors)
- #14497 Add `doc/bitcoin-conf.md` (hebasto)
- #14526 Document lint tests (fanquake)
- #14511 Remove explicit storage requirement from `README.md` (merland)
- #14600 Clarify commit message guidelines (merland)
- #14617 FreeBSD: Document Python 3 requirement for `'gmake check'` (murrayn)
- #14592 Add external interface consistency guarantees (MarcoFalke)
- #14625 Make clear function argument case in dev notes (dongcarl)
- #14515 Update OpenBSD build guide for 6.4 (fanquake)
- #14436 Add comment explaining recent `Rejects-DoS` behavior (jamesob)
- #14684 `conf`: Remove deprecated options from docs, Other cleanup (MarcoFalke)
- #14731 Improve scripted-diff developer docs (dongcarl)
- #14778 A few minor formatting fixes and clarifications to `descriptors.md` (jnewbery)
- #14448 Clarify `rpcwallet` flag url change (JBaczuk)
- #14808 Clarify RPC `rawtransaction` documentation (jlopp)
- #14804 Less confusing documentation for `torpassword` (fanquake)
- #14848 Fix broken Gmane URL in `security-check.py` (cyounkins-bot)
- #14882 `developer-notes.md`: Point out that `UniValue` deviates from upstream (Sjors)
- #14909 Update minimum required Qt (fanquake)
- #14914 Add nice table to `files.md` (emilengler)
- #14741 Indicate `-rpcauth` option password hashing alg (dongcarl)
- #14950 Add NSIS setup/install steps to windows docs (fanquake)
- #13930 Better explain `GetAncestor` check for `m_failed_blocks` in `AcceptBlockHeader` (Sjors)
- #14973 Improve Windows native build instructions (murrayn)
- #15073 `Botbot.me` (IRC logs) not available anymore (anduck)
- #15038 Get more info about GUI-related issue on Linux (hebasto)
- #14832 Add more Doxygen information to Developer Notes (ch4ot1c)
- #15128 Fix download link in `doc/README.md` (merland)
- #15127 Clarifying testing instructions (benthecarman)
- #15132 Add FreeBSD build notes link to `doc/README.md` (fanquake)
- #15173 Explain what `.python-version` does (Sjors)

- #15223 Add information about security to the JSON-RPC doc (harding)
- #15249 Update python docs to reflect that wildcard imports are disallowed (Empact)
- #15176 Get rid of badly named `doc/README_osx.md` (merland)
- #15272 Correct logging return type and RPC example (fanquake)
- #15244 Gdb attaching to process during tests has non-sudo solution (instagibbs)
- #15332 Small updates to `getrawtransaction` description (amitiuttarwar)
- #15354 Add missing `bitcoin-wallet` tool manpages (MarcoFalke)
- #15343 netaddress: Make IPv4 loopback comment more descriptive (dongcarl)
- #15353 Minor textual improvements in `translation_strings_policy.md` (merland)
- #15426 importmulti: add missing description of keypool option (harding)
- #15425 Add missing newline to listunspent help for witnessScript (harding)
- #15348 Add separate productivity notes document (dongcarl)
- #15416 Update FreeBSD build guide for 12.0 (fanquake)
- #15222 Add info about factors that affect dependency list (merland)
- #13676 Explain that mempool memory is added to `-dbcache` (Sjors)
- #15273 Slight tweak to the verify-commits script directions (droark)
- #15477 Remove misleading hint in `getrawtransaction` (MarcoFalke)
- #15489 Update release process for snap package (MarcoFalke)
- #15524 doc: Remove berkeleydb PPA from linux build instructions (MarcoFalke)
- #15559 Correct `analyzepsbt` rpc doc (fanquake)
- #15194 Add comment describing `fDisconnect` behavior (dongcarl)
- #15754 getrpcinfo docs (benthecarman)
- #15763 Update bips.md for 0.18.0 (sipa)
- #15757 List new RPCs in `psbt.md` and `descriptors.md` (sipa)
- #15765 correct `bitcoinconsensus_version` in `shared-libraries.md` (fanquake)
- #15792 describe `onlynet` option in `doc/tor.md` (jonatack)
- #15802 mention creating application support bitcoin folder on OSX (JimmyMow)
- #15799 Clarify RPC versioning (MarcoFalke)

## Credits

Thanks to everyone who directly contributed to this release:

- 1111
- 251
- Aaron Clauson
- Adam Jonas
- Akio Nakamura
- Alexander Leishman

- Alexey Ivanov
- Alexey Poghilenkov
- Amir Abrams
- Amiti Uttarwar
- Andrew Chow
- andrewtoth
- Anthony Towns
- Antoine Le Calvez
- Antoine Riard
- Antti Majakivi
- araspitzu
- Arvid Norberg
- Ben Carman
- Ben Woosley
- benthecarman
- bitcoinhodler
- Carl Dong
- Chakib Benziane
- Chris Moore
- Chris Stewart
- chris-belcher
- Chun Kuan Lee
- Cornelius Schumacher
- Cory Fields
- Craig Younkins
- Cristian Mircea Messel
- Damian Mee
- Daniel Ingram
- Daniel Kraft
- David A. Harding
- DesWurstes
- dexX7
- Dimitri Deijs
- Dimitris Apostolou
- Douglas Roark
- DrahtBot
- Emanuele Cisbani
- Emil Engler
- Eric Scrivner
- fridokus
- Gal Buki
- Gleb Naumenko
- Glenn Willen
- Graham Krizek
- Gregory Maxwell
- Gregory Sanders

- gustavonalle
- Harry Moreno
- Hennadii Stepanov
- Isidoro Ghezzi
- Jack Mallers
- James Hilliard
- James O'Beirne
- Jameson Lopp
- Jeremy Rubin
- Jesse Cohen
- Jim Posen
- John Newbery
- Jon Layton
- Jonas Schnelli
- João Barbosa
- Jordan Baczuk
- Jorge Timón
- Julian Fleischer
- Justin Turner Arthur
- Karel Bílek
- Karl-Johan Alm
- Kaz Wesley
- ken2812221
- Kostiantyn Stepaniuk
- Kristaps Kaupe
- Lawrence Nahum
- Lenny Maiorani
- liuyujun
- lucash-dev
- luciana
- Luke Dashjr
- marcaiaf
- marcoagner
- MarcoFalke
- Martin Erlandsson
- Marty Jones
- Mason Simon
- Michael Ford
- Michael Goldstein
- Michael Polzer
- Mitchell Cash
- mruddy
- Murray Nesbitt
- OverlordQ
- Patrick Strateman
- Pierre Rochard

- Pieter Wuille
- poiuty
- practicalswift
- priscoan
- qubenix
- riordant
- Russell Yanofsky
- Samuel Dobson
- sanket1729
- Sjors Provoost
- Stephan Oeste
- Steven Roose
- Suhas Daftuar
- TheCharlatan
- Tim Ruffing
- Vidar Holen
- vim88
- Walter
- whythat
- Wladimir J. van der Laan
- Zain Iqbal Allarakhia

As well as everyone that helped translating on Transifex.