

Bitcoin Core version 0.16.0 is now available from:

<https://bitcoincore.org/bin/bitcoin-core-0.16.0/>

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

Please report bugs using the issue tracker at GitHub:

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

To receive security and update notifications, please subscribe to:

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

How to Upgrade

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

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

Downgrading warning

Wallets created in 0.16 and later are not compatible with versions prior to 0.16 and will not work if you try to use newly created wallets in older versions. Existing wallets that were created with older versions are not affected by this.

Compatibility

Bitcoin Core is extensively tested on multiple operating systems using the Linux kernel, macOS 10.8+, and Windows Vista and later. Windows XP is not supported.

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

Notable changes

Wallet changes

Segwit Wallet

Bitcoin Core 0.16.0 introduces full support for segwit in the wallet and user interfaces. A new `-addresstype` argument has been added, which supports `legacy`, `p2sh-segwit` (default), and `bech32` addresses. It controls what kind of addresses are produced by `getnewaddress`, `getaccountaddress`, and `createmultisigaddress`. A `-changetype` argument has also been added, with the same options, and by default equal to `-addresstype`, to control which kind of change is used.

A new `address_type` parameter has been added to the `getnewaddress` and `addmultisigaddress` RPCs to specify which type of address to generate. A `change_type` argument has been added to the `fundrawtransaction` RPC to override the `-changetype` argument for specific transactions.

- All segwit addresses created through `getnewaddress` or `*multisig` RPCs explicitly get their redeemscripts added to the wallet file. This means that downgrading after creating a segwit address will work, as long as the wallet file is up to date.
- All segwit keys in the wallet get an implicit redeemscript added, without it being written to the file. This means recovery of an old backup will work, as long as you use new software.
- All keypool keys that are seen used in transactions explicitly get their redeemscripts added to the wallet files. This means that downgrading after recovering from a backup that includes a segwit address will work.

Note that some RPCs do not yet support segwit addresses. Notably, `signmessage` / `verifymessage` doesn't support segwit addresses, nor does `importmulti` at this time. Support for segwit in those RPCs will continue to be added in future versions.

P2WPKH change outputs are now used by default if any destination in the transaction is a P2WPKH or P2WSH output. This is done to ensure the change output is as indistinguishable from the other outputs as possible in either case.

BIP173 (Bech32) Address support ("bc1..." addresses)

Full support for native segwit addresses (BIP173 / Bech32) has now been added. This includes the ability to send to BIP173 addresses (including non-v0 ones), and generating these addresses (including as default new addresses, see above).

A checkbox has been added to the GUI to select whether a Bech32 address or P2SH-wrapped address should be generated when using segwit addresses. When launched with `-addresstype=bech32` it is checked by default. When launched with `-addresstype=legacy` it is unchecked and disabled.

HD-wallets by default

Due to a backward-incompatible change in the wallet database, wallets created with version 0.16.0 will be rejected by previous versions. Also, version 0.16.0 will only create hierarchical deterministic (HD) wallets. Note that this only applies to new wallets; wallets made with previous versions will not be upgraded to be HD.

Replace-By-Fee by default in GUI

The send screen now uses BIP125 RBF by default, regardless of `-walletrbf`. There is a checkbox to mark the transaction as final.

The RPC default remains unchanged: to use RBF, launch with `-walletrbf=1` or use the `replaceable` argument for individual transactions.

Wallets directory configuration (`-walletdir`)

Bitcoin Core now has more flexibility in where the wallets directory can be located. Previously wallet database files were stored at the top level of the bitcoin data directory. The behavior is now:

- For new installations (where the data directory doesn't already exist), wallets will now be stored in a new `wallets/` subdirectory inside the data directory by default.
- For existing nodes (where the data directory already exists), wallets will be stored in the data directory root by default. If a `wallets/` subdirectory already exists in the data directory root, then wallets will be stored in the `wallets/` subdirectory by default.

- The location of the wallets directory can be overridden by specifying a `-walletdir=<path>` option where `<path>` can be an absolute path to a directory or directory symlink.

Care should be taken when choosing the wallets directory location, as if it becomes unavailable during operation, funds may be lost.

Build: Minimum GCC bumped to 4.8.x

The minimum version of the GCC compiler required to compile Bitcoin Core is now 4.8. No effort will be made to support older versions of GCC. See discussion in issue #11732 for more information. The minimum version for the Clang compiler is still 3.3. Other minimum dependency versions can be found in `doc/dependencies.md` in the repository.

Support for signalling pruned nodes (BIP159)

Pruned nodes can now signal BIP159's `NODE_NETWORK_LIMITED` using service bits, in preparation for full BIP159 support in later versions. This would allow pruned nodes to serve the most recent blocks. However, the current change does not yet include support for connecting to these pruned peers.

Performance: SHA256 assembly enabled by default

The SHA256 hashing optimizations for architectures supporting SSE4, which lead to ~~50% speedups in SHA256 on supported hardware~~ (5% faster synchronization and block validation), have now been enabled by default. In previous versions they were enabled using the `--enable-experimental-asm` flag when building, but are now the default and no longer deemed experimental.

GUI changes

- Uses of "µBTC" in the GUI now also show the more colloquial term "bits", specified in BIP176.
- The option to reuse a previous address has now been removed. This was justified by the need to "resend" an invoice, but now that we have the request history, that need should be gone.
- Support for searching by TXID has been added, rather than just address and label.
- A "Use available balance" option has been added to the send coins dialog, to add the remaining available wallet balance to a transaction output.
- A toggle for unblinding the password fields on the password dialog has been added.

RPC changes

New `rescanblockchain` RPC

A new RPC `rescanblockchain` has been added to manually invoke a blockchain rescan. The RPC supports start and end-height arguments for the rescan, and can be used in a multiwallet environment to rescan the blockchain at runtime.

New `savemempool` RPC

A new `savemempool` RPC has been added which allows the current mempool to be saved to disk at any time to avoid it being lost due to crashes / power loss.

Safe mode disabled by default

Safe mode is now disabled by default and must be manually enabled (with `-disablesafemode=0`) if you wish to use it. Safe mode is a feature that disables a subset of RPC calls - mostly related to the wallet and sending - automatically in case certain problem conditions with the network are detected. However, developers have come to regard these checks as not reliable enough to act on automatically. Even with safe mode disabled, they will still cause warnings in the `warnings` field of the `getnetworkinfo` RPC and launch the `-alertnotify` command.

Renamed script for creating JSON-RPC credentials

The `share/rpcuser/rpcuser.py` script was renamed to `share/rpcauth/rpcauth.py`. This script can be used to create `rpcauth` credentials for a JSON-RPC user.

Validateaddress improvements

The `validateaddress` RPC output has been extended with a few new fields, and support for segwit addresses (both P2SH and Bech32). Specifically:

- A new field `iswitness` is True for P2WPKH and P2WSH addresses ("bc1..." addresses), but not for P2SH-wrapped segwit addresses (see below).
- The existing field `isscript` will now also report True for P2WSH addresses.
- A new field `embedded` is present for all script addresses where the script is known and matches something that can be interpreted as a known address. This is particularly true for P2SH-P2WPKH and P2SH-P2WSH addresses. The value for `embedded` includes much of the information `validateaddress` would report if invoked directly on the embedded address.
- For multisig scripts a new `pubkeys` field was added that reports the full public keys involved in the script (if known). This is a replacement for the existing `addresses` field (which reports the same information but encoded as P2PKH addresses), represented in a more useful and less confusing way. The `addresses` field remains present for non-segwit addresses for backward compatibility.
- For all single-key addresses with known key (even when wrapped in P2SH or P2WSH), the `pubkey` field will be present. In particular, this means that invoking `validateaddress` on the output of `getnewaddress` will always report the `pubkey`, even when the address type is P2SH-P2WPKH.

Low-level changes

- The deprecated RPC `getinfo` was removed. It is recommended that the more specific RPCs are used:
 - `getblockchaininfo`
 - `getnetworkinfo`
 - `getwalletinfo`
 - `getmininginfo`
- The wallet RPC `getreceivedbyaddress` will return an error if called with an address not in the wallet.
- The wallet RPC `addwitnessaddress` was deprecated and will be removed in version 0.17, set the `address_type` argument of `getnewaddress`, or option `-addresstype=[bech32|p2sh-segwit]` instead.
- `dumpwallet` now includes hex-encoded scripts from the wallet in the dumpfile, and `importwallet` now imports these scripts, but corresponding addresses may not be added correctly or a manual rescan may be required to find relevant transactions.
- The RPC `getblockchaininfo` now includes an `errors` field.
- A new `blockhash` parameter has been added to the `getrawtransaction` RPC which allows for a raw transaction to be fetched from a specific block if known, even without `-txindex` enabled.
- The `decoderawtransaction` and `fundrawtransaction` RPCs now have optional `iswitness` parameters to override the heuristic witness checks if necessary.

- The `walletpassphrase` timeout is now clamped to 2^{30} seconds.
- Using addresses with the `createmultisig` RPC is now deprecated, and will be removed in a later version. Public keys should be used instead.
- Blockchain rescans now no longer lock the wallet for the entire rescan process, so other RPCs can now be used at the same time (although results of balances / transactions may be incorrect or incomplete until the rescan is complete).
- The `logging` RPC has now been made public rather than hidden.
- An `initialblockdownload` boolean has been added to the `getblockchaininfo` RPC to indicate whether the node is currently in IBD or not.
- `minrelaytxfee` is now included in the output of `getmempoolinfo`

Other changed command-line options

- `-debuglogfile=<file>` can be used to specify an alternative debug logging file.
- bitcoin-cli now has an `-stdinrpcpass` option to allow the RPC password to be read from standard input.
- The `-usehd` option has been removed.
- bitcoin-cli now supports a new `-getinfo` flag which returns an output like that of the now-removed `getinfo` RPC.

Testing changes

- The default regtest JSON-RPC port has been changed to 18443 to avoid conflict with testnet's default of 18332.
- Segwit is now always active in regtest mode by default. Thus, if you upgrade a regtest node you will need to either `-reindex` or use the old rules by adding `vbparams=segwit:0:999999999999` to your regtest `bitcoin.conf`. Failure to do this will result in a `CheckBlockIndex()` assertion failure that will look like: `Assertion '(pindexFirstNeverProcessed != nullptr) == (pindex->nChainTx == 0)' failed.`

0.16.0 change log

Block and transaction handling

- #10953 `aeeed345` Combine scriptPubKey and amount as CTxOut in CScriptCheck (jl2012)
- #11309 `93d20a7` Minor cleanups for AcceptToMemoryPool (morcos)
- #11418 `38c201f` Add error string for CLEANSTACK script violation (maaku)
- #11411 `339da9c` Change SignatureHash input index check to an assert (jimpo)
- #11406 `e12522d` Add state message print to AcceptBlock failure message (TheBlueMatt)
- #11062 `26fee4f` Mark mempool import fails that were found in mempool as 'already there' (kallewoof)
- #11269 `61fb806` CTxMemPoolEntry::UpdateAncestorState: modifySiagOps param type (donalocconnor)
- #11747 `e970396` Fix: Open files read only if requested (Elbandi)
- #11737 `46d1ebf` Document partial validation in ConnectBlock() (sdaftuar)
- #10699 `c090262` Make all script validation flags backward compatible (sipa)
- #10279 `214046f` Add a CChainState class to validation.cpp to take another step towards clarifying internal interfaces (TheBlueMatt)
- #11824 `d9fdac1` Block ActivateBestChain to empty validationinterface queue (TheBlueMatt)
- #12127 `9501dc2` Remove unused mempool index (sdaftuar)
- #12118 `44080a9` Sort mempool by min(feerate, ancestor_feerate) (sdaftuar)
- #8498 `0e3a411` Minimize the number of times it is checked that no money... (jtimon)
- #12368 `3f5012b` Hold mempool.cs for the duration of ATMP (TheBlueMatt)

- #12401 d44cd7e Reset pblocktree before deleting LevelDB file (Sjors)
- #12415 f893824 Interrupt loading thread after shutdown request (promag)

P2P protocol and network code

- #10596 6866b49 Add vConnect to CConnman::Options (benma)
- #10663 9d31ed2 Split resolve out of connect (theuni)
- #11113 fe65c4 Ignore getheaders requests for very old side blocks (jimpo)
- #11585 5aeaa9c addrman: Add missing lock in Clear() (CAddrMan) (practicalswift)
- #11524 5ef3b69 De-duplicate connection eviction logic (tjps)
- #11580 1f4375f Do not send (potentially) invalid headers in response to getheaders (TheBlueMatt)
- #11655 aca77a4 Assert state.m_chain_sync.m_work_header in ConsiderEviction (practicalswift)
- #11744 3ff6ff5 Add missing locks in net.{cpp,h} (practicalswift)
- #11740 59d3dc8 Implement BIP159 NODE_NETWORK_LIMITED (pruned peers) *signaling only* (jonasschnelli)
- #11583 37ffa16 Do not make it trivial for inbound peers to generate log entries (TheBlueMatt)
- #11363 ba2f195 Split socket create/connect (theuni)
- #11917 bc66765 Add testnet DNS seed: seed.testnet.bitcoin.sprovoost.nl (Sjors)
- #11512 6e89de5 Use GetDesireableServiceFlags in seeds, dnsseeds, fixing static seed adding (TheBlueMatt)
- #12262 16bac24 Hardcoded seed update (laanwj)
- #12270 9cf6393 Update chainTxData for 0.16 (laanwj)
- #12392 0f61651 Fix ignoring tx data requests when fPauseSend is set on a peer (TheBlueMatt)

Wallet

- #11039 fc51565 Avoid second mapWallet lookup (promag)
- #10952 2621673 Remove vchDefaultKey and have better first run detection (achow101)
- #11007 fc5c237 Fix potential memory leak when loading a corrupted wallet file (practicalswift)
- #10976 07c92b9 Move some static functions out of wallet.h/cpp (ryanofsky)
- #11117 961901f Prepare for non-Base58 addresses (sipa)
- #10916 e6ab88a add missing lock to crypter GetKeys() (benma)
- #10767 791a0e6 Clarify wallet initialization / destruction interface (jnewbery)
- #11250 c22a53c Bump wallet version to 159900 and remove the usehd option (achow101)
- #11307 4f7e37e Display non-HD error on first run (MarcoFalke)
- #11408 69c7ece Fix parameter name typo in ErasePurpose walletdb method (PierreRochard)
- #11167 aa624b6 Full BIP173 (Bech32) support (sipa)
- #11594 0ecc630 Improve -disablewallet parameter interaction (promag)
- #10368 77ba4bf Remove helper conversion operator from wallet (kallewoof)
- #11074 99ec126 Assert that CWallet::SyncMetaData finds oldest transaction (BitonicEelis)
- #11272 e6e3fc3 CKeystore/CCrypter: move relevant implementation out of the header (jonasschnelli)
- #10286 927a1d7 Call wallet notify callbacks in scheduler thread (without cs_main) (TheBlueMatt)
- #10600 4ed8180 Make feebumper class stateless (ryanofsky)
- #11466 d080a7d Specify custom wallet directory with -walletdir param (MeshCollider)
- #11839 8ab6c0b Don't attempt mempool entry for wallet transactions on startup (instagibbs)
- #11854 2214954 Split up key and script metadata for better type safety (ryanofsky)
- #11870 ef8ba7d Remove unnecessary mempool lock in ReacceptWalletTransactions (promag)
- #11864 2ae58d5 Make CWallet::FundTransaction atomic (promag)
- #11886 df71819 Clarify getbalance meaning a tiny bit in response to questions (TheBlueMatt)

- #11923 81c89e9 Remove unused fNoncriticalErrors variable from CWalletDB::FindWalletTx (PierreRochard)
- #11726 604e08c Cleanups + nit fixes for walletdir PR (MeshCollider)
- #11403 d889c03 Segwit wallet support (sipa)
- #11970 b7450cd Add test coverage for bitcoin-cli multiwallet calls (ryanofsky)
- #11904 66e3af7 Add a lock to the wallet directory (MeshCollider)
- #12101 c7978be Clamp walletpassphrase timeout to 2^30 seconds and check its bounds (achow101)
- #12210 17180fa Deprecate addwitnessaddress (laanwj)
- #12220 f4c942e Error if relative -walletdir is specified (ryanofsky)
- #11281 8470e64 Avoid permanent cs_main/cs_wallet lock during RescanFromTime (jonasschnelli)
- #12119 9594139 Use P2WPKH change output if any destination is P2WPKH or P2WSH (Sjors)
- #12213 eadb2da Add address type option to addmultisigaddress (promag)
- #12276 7936446 Remove duplicate mapWallet lookups (promag)

RPC and other APIs

- #11008 3841aaf Enable disablesafemode by default (gmaxwell)
- #11050 7ed57d3 Avoid treating null RPC arguments different from missing arguments (ryanofsky)
- #10997 affe927 Add option -stdinrpcpass to bitcoin-cli to allow RPC password to be read from standard input (jharvell)
- #11179 e0e3cbb Push down safe mode checks (laanwj)
- #11203 d745b4c add wtxid to mempool entry output (sdaftuar)
- #11099 bc561b4 Add savemempool RPC (greenaddress)
- #10838 66a5b41 (finally) remove getinfo (TheBlueMatt)
- #10753 7fcd61b test: Check RPC argument mapping (laanwj)
- #11288 0f8e095 More user-friendly error message when partially signing (MeshCollider)
- #11031 ef8340d deprecate estimatefee (jnewbery)
- #10858 9a8e916 Add "errors" field to getblockchaininfo and unify "errors" field in get*info RPCs (achow101)
- #11021 90926db Fix getchaintxstats() (AkioNak)
- #11367 3a93270 getblockchaininfo: Add disk_size, prune_target_size (esotericnonsense)
- #11006 a1d78b5 Improve shutdown process (promag)
- #11529 ff92fbf Avoid slow transaction search with txindex enabled (promag)
- #11618 87d90ef Lock cs_main in blockToJSON/blockheaderToJSON (practicalswift)
- #11626 998c304 Make logging RPC public (laanwj)
- #11258 033c786 Add initialblockdownload to getblockchaininfo (jnewbery)
- #11087 99bc0b4 Diagnose unsuitable outputs in lockunspent() (BitonicEelis)
- #11710 9388639 cli: Reject arguments to -getinfo (laanwj)
- #11738 d4267a3 Fix sendrawtransaction hang when sending a tx already in mempool (TheBlueMatt)
- #11753 32c9b57 clarify abortrescan rpc use (instagibbs)
- #11191 ef14f2e Improve help text and behavior of RPC-logging (AkioNak)
- #10874 9e38d35 getblockchaininfo: Loop through the bip9 soft fork deployments instead of hard coding (achow101)
- #10275 497d0e0 Allow fetching tx directly from specified block in getrawtransaction (kallewoof)
- #11178 fee0370 Add iswitness parameter to decode- and fundrawtransaction RPCs (MeshCollider)
- #11667 711d16c Add scripts to dumpwallet RPC (MeshCollider)
- #11475 9bad8d6 mempoolinfo should take ::minRelayTxFee into account (mess110)
- #12001 a9a49e6 Adding ::minRelayTxFee amount to getmempoolinfo and updating help (jeffrade)

- #12198 adce1de Add deprecation error for `getinfo` (laanwj)
- #11415 69ec021 Disallow using addresses in createmultisig (achow101)
- #12278 288deac Add special error for genesis coinbase to getrawtransaction (MeshCollider)
- #11362 c6223b3 Remove nBlockMaxSize from miner opt struct as it is no longer used (gmaxwell)
- #10825 28485c7 Set regtest JSON-RPC port to 18443 to avoid conflict with testnet 18332 (fametrano)
- #11303 e542728 Fix estimatesmartfee rounding display issue (TheBlueMatt)
- #7061 8c2de82 Add RPC call "rescanblockchain " (jonasschnelli)
- #11055 95e14dc RPC getreceivedbyaddress should return error if called with address not owned by the wallet (jnewbery)
- #12366 93de37a http: Join worker threads before deleting work queue (laanwj)
- #12315 758a41e Bech32 addresses in dumpwallet (fivepiece)
- #12427 3762ac1 Make signrawtransaction accept P2SH-P2WSH redeemscripts (sipa)

GUI

- #10964 64e66bb Pass SendCoinsRecipient (208 bytes) by reference (practicalswift)
- #11169 5b8af7b Make tabs toolbar no longer have a context menu (achow101)
- #10911 9c8f365 Fix typo and access key in optionsdialog.ui (keystrike)
- #10770 ea729d5 Drop upgrade-cancel callback registration for a generic "cancelable" (TheBlueMatt)
- #11156 a3624dd Fix memory leaks in qt/guiutil.cpp (danra)
- #11268 31e72b2 [macOS] remove Growl support, remove unused code (jonasschnelli)
- #11193 c5c77bd Terminate string *pszExePath after readlink and without using memset (practicalswift)
- #11508 ffa5159 Fix crash via division by zero assertion (jonasschnelli)
- #11499 6157e8c Add upload and download info to the peerlist (debug menu) (aarongolliver)
- #11480 ffc0b11 Add toggle for unblinding password fields (tjps)
- #11316 22cdf93 Add use available balance in send coins dialog (CryptAxe, promag)
- #3716 13e352d Receive: Remove option to reuse a previous address (luke-jr)
- #11690 f0c1f8a Fix the StartupWMClass for bitcoin-qt, so gnome-shell can recognize it (eklitze)
- #10920 f6f8d54 Fix potential memory leak in newPossibleKey(ChangeCWallet *wallet) (practicalswift)
- #11698 7293d06 RPC-Console nested commands documentation (lmlsna)
- #11395 38d31f9 Enable searching by transaction id (luke-jr)
- #11556 91eeaa0 Improved copy for RBF checkbox and tooltip (Sjors)
- #11809 80f9dad Fix proxy setting options dialog crash (laanwj)
- #11616 8585bb8 Update ban-state in case of dirty-state during periodic sweep (jonasschnelli)
- #11605 f19ca12 Enable RBF by default in QT (Sjors)
- #12074 a1136f0 Optimizes boolean expression model && model->haveWatchOnly() (251Labs)
- #12035 eeb6d52 Change µBTC to bits (jb55)
- #12092 fd4ca17 Replaces numbered place marker %2 with %1 (251Labs)
- #12173 bbc91b7 Use flexible font size for QRCode image address (jonasschnelli)
- #12211 10d10d7 Avoid potential null dereference in ReceiveCoinsDialog constructor (ryanofsky)
- #12261 f359afc Bump BLOCK_CHAIN_SIZE to 200GB (laanwj)
- #11991 062c8b6 Receive: checkbox for bech32 address (Sjors)
- #11644 045a809 Fix qt build broken by 5a5e4e9 (TheBlueMatt)
- #11448 d473e6d reset addrProxy/addrSeparateProxyTor if colon char missing (mess110)
- #12377 604f289 qt: Poll ShutdownTimer after init is done (MarcoFalke)
- #12374 daaa36 qt: Make sure splash screen is freed on ApplnitMain fail (laanwj)
- #12349 ad10b90 shutdown: fix crash on shutdown with reindex-chainstate (theuni)

Build system

- #10923 2c9f5ec travis: Build with --enable-werror under OS X (practicalswift)
- #11176 df8c722 build: Rename --enable-experimental-asm to --enable-asm and enable by default (laanwj)
- #11286 11dacc6 [depends] Don't build libevent sample code (fanquake)
- #7142 801dd40 Travis: Test build against system libs (& Qt4) (luke-jr)
- #11380 390771b Remove outdated share/certs/ directory (MeshCollider)
- #11391 7632310 Remove lxcbr0 lines from gitian-build.sh (MeshCollider)
- #11435 167cef8 build: Make "make clean" remove all files created when running "make check" (practicalswift)
- #11460 e022463 [depends] mac_alias 2.0.6, ds_store 1.1.2 (fanquake)
- #11541 bb9ab0f Build: Fix Automake warnings when running autogen.sh (fanquake)
- #11611 0e70791 [build] Don't fail when passed --disable-lcov and lcov isn't available (fanquake)
- #11651 3c098a8 refactor: Make all #includes relative to project root (laanwj, MeshCollider, ryanofsky)
- #11621 1f7695b [build] Add temp_bitcoin_locale_qrc to CLEAN_QT to fix make distcheck (fanquake)
- #11755 84fa645 [Docs] Bump minimum required version of GCC to 4.8 (fanquake)
- #9254 6d3dc52 [depends] ZeroMQ 4.2.2 (fanquake)
- #11842 3c8f0a3 [build] Add missing stuff to clean-local (kallewoof)
- #11936 483bb67 [build] Warn that only libconsensus can be built without Boost (fanquake)
- #11945 7a11ba7 Improve BSD compatibility of contrib/install_db4.sh (laanwj)
- #11981 180a255 Fix gitian build after libzmq bump (theuni)
- #11903 8f68fd2 [trivial] Add required package dependencies for depends cross compilation (jonasschnelli)
- #12168 45cf8a0 #include sys/fcntl.h to just fcntl.h (without sys/) (jsarenik)
- #12095 3fa1ab4 Use BDB_LIBS/CFLAGS and pass --disable-replication (fanquake)
- #11711 6378e5c bitcoin_qt.m4: Minor fixes and clean-ups (fanquake)
- #11989 90d4104 .gitignore: add QT Creator artifacts (Sjors)
- #11577 c0ae864 Fix warnings (-Wsign-compare) when building with DEBUG_ADDRMAN (practicalswift)

Tests and QA

- #11024 3e55f13 Remove OldSetKeyFromPassphrase/OldEncrypt/OldDecrypt (practicalswift)
- #10679 31b2612 Document the non-DER-conformance of one test in tx_valid.json (schildbach)
- #11160 ede386c Improve versionbits_computeblockversion test code consistency (danra)
- #10303 f088a1b Include ms/blk stats in Connect* benchmarks (kallewoof)
- #10777 d81dccb Avoid redundant assignments. Remove unused variables (practicalswift)
- #11260 52f8877 travis: Assert default datadir isn't created, Run scripted diff only once (MarcoFalke)
- #11271 638e6c5 travis: filter out pyenv (theuni)
- #11285 3255d63 Add -usehd to excluded args in check-doc.py (MeshCollider)
- #11297 16e4184 Make sure ~/.bitcoin doesn't exist before build (MeshCollider)
- #11311 cce94c5 travis: Revert default datadir check (MarcoFalke)
- #11300 f4ed44a Add a lint check for trailing whitespace (MeshCollider)
- #11323 4ce2f3d mininode: add an optimistic write and disable nagle (theuni)
- #11370 2d85899 Add getblockchaininfo functional test (promag)
- #11365 f199b8a Add Qt GUI tests to Overview and ReceiveCoin Page (anditto)
- #11293 dbc4ae0 Deduplicate CMerkleBlock construction code, add test coverage (jamesob)
- #10440 9e8ef9d Add libFuzzer support (practicalswift)

- #10941 364da2c Add blocknotify and walletnotify functional tests (promag)
- #11420 8928093 Bump univalue subtree and fix json formatting in tests (MarcoFalke)
- #10099 424be03 Slightly Improve Unit Tests for Checkqueue (JeremyRubin)
- #11513 14b860b A few Python3 tidy ups (jnewbery)
- #11486 2ca518d Add uacomment tests (mess110)
- #11452 02ac8c8 Improve ZMQ functional test (promag)
- #10409 b5545d8 Add fuzz testing for BlockTransactions and BlockTransactionsRequest (practicalswift)
- #11389 dd56166 Support having segwit always active in regtest (sipa, ajtowns, jnewbery)
- #11562 5776582 bench: use std::chrono rather than gettimeofday (theuni)
- #11182 f7388e9 Add P2P interface to TestNode (jnewbery)
- #11552 b5f9f02 Improve wallet-accounts test (ryanofsky)
- #11638 5e3f5e4 Dead mininode code (jnewbery)
- #11646 fe503e1 Require a steady clock for bench with at least micro precision (TheBlueMatt)
- #11468 76b3349 Make comp test framework more debuggable (jnewbery)
- #11623 ee92243 Add missing locks to tests (practicalswift)
- #11035 927e528 [contrib] Add Valgrind suppressions file (practicalswift)
- #11641 7adeea3 Only allow disconnecting all NodeConns (MarcoFalke)
- #11677 3bdf242 Remove unused NodeConn members (MarcoFalke)
- #11699 66d46c7 [travis-ci] Only run linters on Pull Requests (jnewbery)
- #11654 084f52f Initialize recently introduced non-static class member lastCycles to zero in constructor (practicalswift)
- #11648 ccc70a2 Add messages.py (jnewbery)
- #11713 49667a7 Fix for mismatched extern definition in wallet tests (sipsorcery)
- #11707 0d89fa0 Fix sendheaders (jnewbery)
- #11718 9cdd2bc Move pwalletMain to wallet test fixture (laanwj)
- #11714 901ba3e Test that mempool rejects coinbase transactions (jamesob)
- #11743 3d6ad40 Add multiwallet prefix test (MarcoFalke)
- #11683 a892218 Remove unused mininode functions {ser,deser}_int_vector(...). Remove unused imports (practicalswift)
- #11712 9f2c2db Split NodeConn from NodeConnCB (jnewbery)
- #11791 13e31dd Rename NodeConn and NodeConnCB (jnewbery)
- #11835 f60b4ad Add Travis check for unused Python imports (practicalswift)
- #11849 ad1820c Assert that only one NetworkThread exists (jnewbery)
- #11877 d4991c0 Improve createrawtransaction functional tests (promag)
- #11220 2971fd0 Check specific validation error in miner tests (Sjors)
- #11947 797441e Fix rawtransactions test (laanwj)
- #11946 8049241 Remove unused variable (firstAddrnServices) (practicalswift)
- #11867 18a1bba Improve node network test (jnewbery)
- #11883 cfd99dd Add configuration file/argument testing (MeshCollider)
- #11879 d4e404a Remove redundant univalue_tests.cpp (jnewbery)
- #11748 20166f8 Adding unit tests for GetDifficulty in blockchain.cpp (merehap)
- #11517 5180a86 Improve benchmark precision (martinus)
- #11291 a332a7d Fix string concatenation to os.path.join and add exception case (dongsam)
- #11965 d38d1a3 Note on test order in test_runner (MarcoFalke)
- #11997 ddff344 util_tests.cpp: actually check ignored args (ajtowns)
- #12079 45173fa Improve prioritisetransaction test coverage (promag)

- #12150 92a810d Fix ListCoins test failure due to unset g_address_type, g_change_type (ryanofsky)
- #12133 1d2eaba Fix rare failure in p2p-segwit.py (sdaftuar)
- #12082 0910cbe Adding test case for SINGLE|ANYONECANPAY hash type in tx_valid.json (Christewart)
- #11796 4db16ec Functional test naming convention (ajtowns)
- #12227 b987ca4 test_runner: Readable output if create_cache.py fails (ryanofsky)
- #12089 126000b Make TestNodeCLI command optional in send_cli (MarcoFalke)
- #11774 6970b30 Rename functional tests (ajtowns)
- #12264 598a9c4 Fix versionbits warning test (jnewbery)
- #12217 1213be6 Add missing syncwithvalidationinterfacequeue to tests (MarcoFalke)
- #12292 eebe458 Fix names of excluded extended tests for travis (ajtowns)
- #11789 60d739e [travis-ci] Combine logs on failure (jnewbery)
- #11838 3e50024 Add getrawtransaction in_active_chain=False test (MarcoFalke)
- #12206 898f560 Sync with validationinterface queue in sync_mempools (MarcoFalke)
- #12424 ff44101 Fix rescan test failure due to unset g_address_type, g_change_type (ryanofsky)
- #12388 e2431d1 travis: Full clone for git subtree check (MarcoFalke)

Documentation

- #10680 6366941 Fix inconsistencies and grammar in various files (MeshCollider)
- #11011 7db65c3 Add a comment on the use of prevector in script (gmaxwell)
- #10878 c58128f Fix Markdown formatting issues in init.md (dongcarl)
- #11066 9e00a62 Document the preference of nullptr over NULL or (void*)0 (practicalswift)
- #11094 271e40a Hash in ZMQ hash is raw bytes, not hex (runn1ng)
- #11026 ea3ac59 Bugfix: Use testnet RequireStandard for -acceptnonstdtxn default (luke-jr)
- #11058 4b65fa5 Comments: More comments on functions/globals in standard.h (jimpo)
- #11112 3f726c9 [developer-notes] By default, declare single-argument constructors "explicit" (practicalswift)
- #11155 a084767 Trivial: Documentation fixes for CVectorWriter ctors (danra)
- #11136 108222b Docs: Add python3 to list of dependencies on some platforms (danra)
- #11216 81f8c03 Update hmac_sha256.h (utsavgupta)
- #11236 ba05971 Add note on translations to CONTRIBUTING.md (MeshCollider)
- #11173 4eb1f39 RPC: Fix currency unit string in the help text (AkioNak)
- #11135 21e2f2f Update developer notes with RPC response guidelines (promag)
- #11219 bcc8a62 explain how to recompile a modified unit test (Sjors)
- #10779 f656147 Create dependencies.md (flack)
- #10682 2a56baf Move the AreInputsStandard documentation next to its implementation (esneider)
- #11276 ee50c9e Update CONTRIBUTING.md to reduce unnecessary review workload (jonasschnelli)
- #11264 b148803 Fix broken Markdown table in dependencies.md (practicalswift)
- #10691 ce82985 Properly comment about shutdown process in init.cpp file (wraith7)
- #11330 ae233c4 Fix comments for DEFAULT_WHITELIST[FORCE]RELAY (danra)
- #11340 d6d2c85 Fix validation comments (danra)
- #11305 2847480 Update release notes and manpages for 0.16 (MarcoFalke)
- #11132 551d7bf Document assumptions that are being made to avoid NULL pointer dereferences (practicalswift)
- #11390 12ed800 Document scripted-diff (jnewbery)
- #11392 a3b4c59 Fix stale link in gitian-building.md (shooterman)
- #11401 4202273 Move gitian building to external repo (MarcoFalke)
- #11414 bbc901d Remove partial gitian build instructions from descriptors dir (fanquake)

- #11571 c95832d Fixed a couple small grammatical errors (BitsInMyBlood)
- #11624 f9b74ef Change formatting for sequence of steps (vivganes)
- #11597 6f01dcf Fix error messages in CFeeBumper (kallewoof)
- #11438 7fbf3c6 Updated Windows build doc for WSL/Xenial workaround (sipsorcery)
- #11663 41aa9c4 Add getreceivedbyaddress release notes (MarcoFalke)
- #11533 cbb54e7 Update WSL installation notes for Fall Creators update (Thoragh)
- #11680 4db82b7 Add instructions for lcov report generation (jamesob)
- #11686 54aedc0 Make ISSUE_TEMPLATE a bit shorter, mention hardware tests (TheBlueMatt)
- #11704 ea68190 Windows build doc update (sipsorcery)
- #11706 5197100 Make default issue text all comments to make issues more readable (TheBlueMatt)
- #11140 1429132 Improve #endif comments (danra)
- #11729 7a43fbb links to code style guides (Sjors)
- #11793 8879d50 Bump OS X version to 10.13 (Varunram)
- #11783 16fff80 Fix shutdown in case of errors during initialization (laanwj)
- #11804 00d25e9 Fixed outdated link with archive.is (TimothyShimmin)
- #11960 4307062 Fix link to installation script (laudaa)
- #12027 63a4dc1 Remove boost --c++ flag from osx build instructions (fernandezpablo85)
- #12062 5961b23 Increment MIT Licence copyright header year on files modified in 2017 (akx20000a)
- #12063 36a5a44 Update license year range to 2018 (akx20000a)
- #12093 5691028 Fix incorrect Markdown link (practicalswift)
- #12143 b0d626d Fix link for BIP159 pull request (azuchi)
- #12112 3c62868 Remove the ending slashes from RPC URI format (jackycjh)
- #12166 e839d65 Clarify -walletdir usage (jnewbery)
- #12241 b030133 Fix incorrect link in /test/ README.md (fanquake)
- #12187 b5e4b9b Updating benchmarkmarking.md with an updated sample output (jeffrade)
- #12294 7cf1aea Create NetBSD build instructions and fix compilation (fanquake)
- #12251 cc5870a initwallet: Do not translate highly technical addresstype help (MarcoFalke)
- #11984 efae366 Update OpenBSD build instructions for 6.2 (cont'd) (laanwj)
- #12293 9d9c418 Mention that HD is enabled if hdmasterkeyid is present in getwalletinfo RPC help (fanquake)
- #12077 c04cb48 Correct sendmany curl example (251Labs)
- #10677 b3ecb7b Document that addmultisigaddress is intended for non-watchonly addresses (instagibbs)
- #12177 cad504b Fix address_type help text of getnewaddress and getrawchangeaddress (mruddy)

Refactoring

- #9964 b6a4891 Add const to methods that do not modify the object for which it is called (practicalswift)
- #10965 655970d Replace deprecated throw() with noexcept specifier (C++11) (practicalswift)
- #10645 c484ec6 Use nullptr (C++11) instead of zero (0) as the null pointer constant (practicalswift)
- #10901 22e301a Fix constness of ArgsManager methods (promag)
- #10969 4afb5aa Declare single-argument (non-converting) constructors "explicit" (practicalswift)
- #11071 dbf6bd6 Use static_assert(..., ...) (C++11) instead of assert(...) where appropriate (practicalswift)
- #10809 c559884 optim: mark a few classes final (theuni)
- #10843 2ab7c63 Add attribute [[noreturn]] (C++11) to functions that will not return (practicalswift)
- #11151 7fd49d0 Fix header guards using reserved identifiers (danra)
- #11138 2982511 Compat: Simplify bswap_16 implementation (danra)
- #11161 745bbdc Remove redundant explicitly defined copy ctors (danra)

- #11144 cee4fe1 Move local include to before system includes (danra)
- #10781 60dd9cc Python cleanups (practicalswift)
- #10701 50fae68 Remove the virtual specifier for functions with the override specifier (practicalswift)
- #11164 38a54a5 Fix boost headers included as user instead of system headers (danra)
- #11143 3aa60b7 Fix include path for bitcoin-config.h (danra)
- #8330 59e1789 Structure Packing Optimizations in C{Mutable}Transaction (JeremyRubin)
- #10845 39ae413 Remove unreachable code (practicalswift)
- #11238 6acdb1f Add assertions before potential null deferences (MeshCollider)
- #11259 089b742 Remove duplicate destination decoding (promag)
- #11232 2f0d3e6 Ensure that data types are consistent (jjz)
- #10793 efb4383 Changing &var[0] to var.data() (MeshCollider)
- #11196 e278f86 Switch memory_cleane implementation to BoringSSL's to ensure memory clearing even with -lto (maaku)
- #10888 9821274 range-based loops and const qualifications in net.cpp (benma)
- #11351 6c4fecf Refactor: Modernize disallowed copy constructors/assignment (danra)
- #11385 94c9015 Remove some unused functions and methods (sipa)
- #11301 8776787 add m_added_nodes to connman options (benma)
- #11432 058c0f9 Remove unused fTry from push_lock (promag)
- #11107 e93fff1 Fix races in AppInitMain and others with lock and atomic bools (MeshCollider)
- #9572 17f2ace Skip witness sighash cache for non-segwit transactions (jl2012)
- #10961 da0478e Improve readability of DecodeBase58Check(...) (practicalswift)
- #11133 a865b38 Document assumptions that are being made to avoid division by zero (practicalswift)
- #11073 3bb77eb Remove dead store in ecdsa_signature_parse_der_lax (BitonicEelis)
- #10898 470c730 Fix invalid checks (NULL checks after dereference, redundant checks, etc.) (practicalswift)
- #11495 50d72b3 [trivial] Make namespace explicit for is_regular_file (jnewbery)
- #11511 db2f83e [Init] Remove redundant exit(EXIT_FAILURE) instances and replace with return false (donalocconnor)
- #10866 ef8a634 Fix -Wthread-safety-analysis warnings. Compile with -Wthread-safety-analysis if available (practicalswift)
- #11221 0dec4cc Refactor: simpler read (gnuser)
- #10696 ef3758d Remove redundant nullptr checks before deallocation (practicalswift)
- #11043 5e9be16 Use std::unique_ptr (C++11) where possible (practicalswift)
- #11353 05a7619 Small refactor of CCoinsViewCache::BatchWrite() (danra)
- #10749 2adbddb Use compile-time constants instead of unnamed enumerations (remove "enum hack") (practicalswift)
- #11603 a933cb1 Move RPC registration out of AppInitParameterInteraction (ryanofsky)
- #11722 26efc22 Switched sync.{cpp,h} to std threading primitives (tjps)
- #10493 fbce66a Use range-based for loops (C++11) when looping over map elements (practicalswift)
- #11337 0d7e0a3 Fix code constness in CBlockIndex::GetAncestor() overloads (danra)
- #11516 0e722e8 crypto: Add test cases covering the relevant HMAC-SHA{256,512} key length boundaries (practicalswift)
- #10574 5d132e8 Remove includes in .cpp files for things the corresponding .h file already included (practicalswift)
- #11884 66479c0 Remove unused include in hash.cpp (kallewoof)
- #10839 c66adb2 Don't use pass by reference to const for cheaply-copied types (bool, char, etc.) (practicalswift)
- #10657 79399c8 Utils: Improvements to ECDSA key-handling code (str4d)

- #12250 e37ca2b Make CKey::Load references const (ryanofsky)
- #12108 9220426 Remove unused fQuit var from checkqueue.h (donaloconnor)
- #12159 f3c7062 Use the character based overload for std::string::find (kekimusmaximus)
- #12266 3448907 Move scheduler/threadGroup into common-init instead of per-app (TheBlueMatt)

Miscellaneous

- #11246 777519b github-merge: Coalesce git fetches (laanwj)
- #10871 c9a4aa8 Handle getinfo in bitcoin-cli w/ -getinfo (revival of #8843) (achow101)
- #11419 093074b Utils: Fix launchctl not being able to stop bitcoind (OmeGak)
- #11394 6e4e98e Perform a weaker subtree check in Travis (sipa)
- #11702 4122112 [build] Add a script for installing db4 (jamesob)
- #11794 dd49862 Prefix leveldb debug logging (laanwj)
- #11781 24df9af Add -debuglogfile option (laanwj)
- #10773 c17f11f Shell script cleanups (practicalswift)
- #11829 7630a1f Test datadir specified in conf file exists (MeshCollider)
- #11836 d44535d Rename rpcuser.py to rpcauth.py (hkjn)
- #11831 d48ab83 Always return true if ApplInitMain got to the end (TheBlueMatt)
- #11943 1808660 contrib: fix typo in install_db4.sh help message (laanwj)
- #12075 c991b30 [scripts] Add missing univalue file to copyright_header.py (fanquake)
- #12197 000ac4f Log debug build status and warn when running benchmarks (laanwj)
- #10672 6ab0e4c Avoid division by zero in the case of a corrupt estimates file (practicalswift)
- #11273 cdd6bbf Ignore old format estimation file (Xekyo)
- #11951 1fb34e0 Remove dead feeest-file read code for old versions (TheBlueMatt)
- #11421 9ccafb1 Merge current secp256k1 subtree (MarcoFalke)
- #11573 2631d55 [Util] Update tinyformat.h (fanquake)
- #10529 331352f Improve bitcoind systemd service file (Flowdalic)
- #11620 70fec9e [build] .gitignore: add background.tiff (Sjors)
- #11558 68e021e Minimal code changes to allow msvc compilation (sipsorcery)
- #11284 10bee0d Fix invalid memory access in CScript::operator+= (guidovranken, ajtowns)
- #10939 a1f7f18 [init] Check non-emptiness of -blocknotify command prior to executing (practicalswift)
- #11467 937613d Fix typos. Use nullptr instead of NULL (practicalswift)
- #11834 5bea05b [verify-commits] Fix gpg.sh's echoing for commits with '\n' (TheBlueMatt)
- #11830 a13e443 rpcuser.py: Use 'python' not 'python2' (hkjn)
- #12194 7abb0f0 Add change type option to fundrawtransaction (promag)
- #12269 2ae7cf8 Update defaultAssumeValid to block 506067 (gmaxwell)
- #11952 9ab9963 univalue: Bump subtree (MarcoFalke)
- #12367 09fc859 Fix two fast-shutdown bugs (TheBlueMatt)
- #12422 4d54e7a util: Make LockDirectory thread-safe, consistent, and fix OpenBSD 6.2 build (laanwj)

Credits

Thanks to everyone who directly contributed to this release:

- 251
- Aaron Clauson
- Aaron Golliver
- aaron-hanson

- Adam Langley
- Akio Nakamura
- Akira Takizawa
- Alejandro Avilés
- Alex Morcos
- Alin Rus
- Anditto Heristyo
- Andras Elso
- Andreas Schildbach
- Andrew Chow
- Anthony Towns
- azuchi
- Carl Dong
- Chris Moore
- Chris Stewart
- Christian Gentry
- Cory Fields
- Cristian Mircea Messel
- CryptAxe
- Dan Raviv
- Daniel Edgecumbe
- danra
- david60
- Donal O'Connor
- dongsamb
- Dusty Williams
- Eelis
- esneider
- Evan Klitzke
- fanquake
- Ferdinando M. Ametrano
- fivepiece
- flack
- Florian Schmaus
- gnuser
- Gregory Maxwell
- Gregory Sanders
- Henrik Jonsson
- Jack Grigg
- Jacky C
- James Evans
- James O'Beirne
- Jan Sarenik
- Jeff Rade
- Jeremiah Buddenhagen
- Jeremy Rubin
- Jim Posen
- jjz
- Joe Harvell
- Johannes Kanig

- John Newbery
- Johnson Lau
- Jonas Nick
- Jonas Schnellli
- João Barbosa
- Jorge Timón
- Karel Bílek
- Karl-Johan Alm
- klemens
- Kyuntae Ethan Kim
- laudaa
- Lawrence Nahum
- Lucas Betschart
- Luke Dashjr
- Luke Mlsna
- MarcoFalke
- Mark Friedenbach
- Marko Bencun
- Martin Ankerl
- Matt Corallo
- mruddy
- Murch
- NicolasDorier
- Pablo Fernandez
- Paul Berg
- Pedro Branco
- Pierre Rochard
- Pieter Wuille
- practicalswift
- Randolph Richardson
- Russell Yanofsky
- Samuel Dobson
- Sean Erle Johnson
- Shooter
- Sjors Provoost
- Suhas Daftuar
- Thomas Snider
- Thoragh
- Tim Shimmin
- Tomas van der Wansem
- Utsav Gupta
- Varunram Ganesh
- Vivek Ganesan
- Werner Lemberg
- William Casarin
- Willy Ko
- Wladimir J. van der Laan

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