Bitcoin Core version 0.9.0 is now available from:

https://bitcoin.org/bin/0.9.0/

This is a new major version release, bringing both new features and bug fixes.

Please report bugs using the issue tracker at github:

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

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), uninstall all earlier versions of Bitcoin, then run the installer (on Windows) or just copy over /Applications/Bitcoin-Qt (on Mac) or bitcoind/bitcoin-qt (on Linux).

If you are upgrading from version 0.7.2 or earlier, the first time you run 0.9.0 your blockchain files will be re-indexed, which will take anywhere from 30 minutes to several hours, depending on the speed of your machine.

On Windows, do not forget to uninstall all earlier versions of the Bitcoin client first, especially if you are switching to the 64-bit version.

Windows 64-bit installer

New in 0.9.0 is the Windows 64-bit version of the client. There have been frequent reports of users running out of virtual memory on 32-bit systems during the initial sync. Because of this it is recommended to install the 64-bit version if your system supports it.

NOTE: Release candidate 2 Windows binaries are not code-signed; use PGP and the SHA256SUMS.asc file to make sure your binaries are correct. In the final 0.9.0 release, Windows setup.exe binaries will be code-signed.

OSX 10.5 / 32-bit no longer supported

0.9.0 drops support for older Macs. The minimum requirements are now: * A 64-bit-capable CPU (see http://support.apple.com/kb/ht3696); * Mac OS 10.6 or later (see https://support.apple.com/kb/ht1633).

Downgrading warnings

The 'chainstate' for this release is not always compatible with previous releases, so if you run 0.9 and then decide to switch back to a 0.8.x release you might get a blockchain validation error when starting the old release (due to 'pruned outputs' being omitted from the index of unspent transaction outputs).

Running the old release with the -reindex option will rebuild the chainstate data structures and correct the problem.

Also, the first time you run a 0.8.x release on a 0.9 wallet it will rescan the blockchain for missing spent coins, which will take a long time (tens of minutes on a typical machine).

Rebranding to Bitcoin Core

To reduce confusion between Bitcoin-the-network and Bitcoin-the-software we have renamed the reference client to Bitcoin Core.

OP_RETURN and data in the block chain

On OP_RETURN: There was been some confusion and misunderstanding in the community, regarding the OP_RETURN feature in 0.9 and data in the blockchain. This change is not an endorsement of storing data in the blockchain. The OP_RETURN change creates a provably-prunable output, to avoid data storage schemes – some of which were already deployed – that were storing arbitrary data such as images as forever-unspendable TX outputs, bloating bitcoin's UTXO database.

Storing arbitrary data in the blockchain is still a bad idea; it is less costly and far more efficient to store non-currency data elsewhere.

Autotools build system

For 0.9.0 we switched to an autotools-based build system instead of individual (q)makefiles.

Using the standard "./autogen.sh; ./configure; make" to build Bitcoin-Qt and bitcoind makes it easier for experienced open source developers to contribute to the project.

Be sure to check doc/build-*.md for your platform before building from source.

Bitcoin-cli

Another change in the 0.9 release is moving away from the bitcoind executable functioning both as a server and as a RPC client. The RPC client functionality ("tell the running bitcoin daemon to do THIS") was split into a separate executable, 'bitcoin-cli'. The RPC client code will eventually be removed from bitcoind, but will be kept for backwards compatibility for a release or two.

`walletpassphrase` RPC

The behavior of the `walletpassphrase` RPC when the wallet is already unlocked has changed between 0.8 and 0.9.

The 0.8 behavior of `walletpassphrase` is to fail when the wallet is already unlocked:

```
> walletpassphrase 1000
walletunlocktime = now + 1000
> walletpassphrase 10
Error: Wallet is already unlocked (old unlock time stays)
```

The new behavior of `walletpassphrase` is to set a new unlock time overriding the old one:

```
> walletpassphrase 1000
walletunlocktime = now + 1000
> walletpassphrase 10
walletunlocktime = now + 10 (overriding the old unlock time)
```

Transaction malleability-related fixes

This release contains a few fixes for transaction ID (TXID) malleability issues:

- -nospendzeroconfchange command-line option, to avoid spending zero-confirmation change
- IsStandard() transaction rules tightened to prevent relaying and mining of mutated transactions
- Additional information in listtransactions/gettransaction output to report wallet transactions that conflict with each other because they spend the same outputs.
- Bug fixes to the getbalance/listaccounts RPC commands, which would report incorrect balances for double-spent (or mutated) transactions.
- New option: -zapwallettxes to rebuild the wallet's transaction information

Transaction Fees

This release drops the default fee required to relay transactions across the network and for miners to consider the transaction in their blocks to 0.01mBTC per kilobyte.

Note that getting a transaction relayed across the network does NOT guarantee that the transaction will be accepted by a miner; by default, miners fill their blocks with 50 kilobytes of high-priority transactions, and then with 700 kilobytes of the highest-fee-per-kilobyte transactions.

The minimum relay/mining fee-per-kilobyte may be changed with the minrelaytxfee option. Note that previous releases incorrectly used the mintxfee setting to determine which low-priority transactions should be considered for inclusion in blocks.

The wallet code still uses a default fee for low-priority transactions of 0.1mBTC per kilobyte. During periods of heavy transaction volume, even this fee may not be enough to get transactions confirmed quickly; the mintxfee option may be used to override the default.

0.9.0 Release notes

RPC:

- New notion of 'conflicted' transactions, reported as confirmations: -1
- 'listreceivedbyaddress' now provides tx ids
- Add raw transaction hex to 'gettransaction' output
- Updated help and tests for 'getreceivedby(account|address)'
- In 'getblock', accept 2nd 'verbose' parameter, similar to getrawtransaction, but defaulting to 1 for backward compatibility
- Add 'verifychain', to verify chain database at runtime
- Add 'dumpwallet' and 'importwallet' RPCs
- 'keypoolrefill' gains optional size parameter
- Add 'getbestblockhash', to return tip of best chain
- Add 'chainwork' (the total work done by all blocks since the genesis block) to 'getblock' output
- Make RPC password resistant to timing attacks
- Clarify help messages and add examples
- Add 'getrawchangeaddress' call for raw transaction change destinations
- Reject insanely high fees by default in 'sendrawtransaction'
- Add RPC call 'decodescript' to decode a hex-encoded transaction script
- Make 'validateaddress' provide redeemScript
- Add 'getnetworkhashps' to get the calculated network hashrate
- New RPC 'ping' command to request ping, new 'pingtime' and 'pingwait' fields in 'getpeerinfo' output
- Adding new 'addrlocal' field to 'getpeerinfo' output
- Add verbose boolean to 'getrawmempool'
- Add rpc command 'getunconfirmedbalance' to obtain total unconfirmed balance
- Explicitly ensure that wallet is unlocked in `importprivkey`
- Add check for valid keys in `importprivkey`

Command-line options:

- New option: -nospendzeroconfchange to never spend unconfirmed change outputs
- New option: -zapwallettxes to rebuild the wallet's transaction information
- Rename option '-tor' to '-onion' to better reflect what it does
- Add '-disablewallet' mode to let bitcoind run entirely without wallet (when built with wallet)
- Update default '-rpcsslciphers' to include TLSv1.2
- make '-logtimestamps' default on and rework help-message
- RPC client option: '-rpcwait', to wait for server start
- Remove '-logtodebugger'
- Allow `-noserver` with bitcoind

Block-chain handling and storage:

- Update leveldb to 1.15
- Check for correct genesis (prevent cases where a datadir from the wrong network is accidentally loaded)
- Allow txindex to be removed and add a reindex dialog
- Log aborted block database rebuilds
- Store orphan blocks in serialized form, to save memory
- Limit the number of orphan blocks in memory to 750
- Fix non-standard disconnected transactions causing mempool orphans
- Add a new checkpoint at block 279,000

Wallet:

- Bug fixes and new regression tests to correctly compute the balance of wallets containing double-spent (or mutated) transactions
- Store key creation time. Calculate whole-wallet birthday.
- Optimize rescan to skip blocks prior to birthday
- Let user select wallet file with -wallet=foo.dat
- Consider generated coins mature at 101 instead of 120 blocks
- Improve wallet load time
- Don't count txins for priority to encourage sweeping
- Don't create empty transactions when reading a corrupted wallet
- Fix rescan to start from beginning after importprivkey
- Only create signatures with low S values

Mining:

- Increase default -blockmaxsize/prioritysize to 750K/50K
- 'getblocktemplate' does not require a key to create a block template
- Mining code fee policy now matches relay fee policy

Protocol and network:

- Drop the fee required to relay a transaction to 0.01mBTC per kilobyte
- Send tx relay flag with version
- New 'reject' P2P message (BIP 0061, see https://gist.github.com/gavinandresen/7079034 for draft)
- Dump addresses every 15 minutes instead of 10 seconds
- Relay OP_RETURN data TxOut as standard transaction type
- Remove CENT-output free transaction rule when relaying
- Lower maximum size for free transaction creation
- Send multiple inv messages if mempool.size > MAX_INV_SZ
- Split MIN_PROTO_VERSION into INIT_PROTO_VERSION and MIN_PEER_PROTO_VERSION
- Do not treat fFromMe transaction differently when broadcasting
- Process received messages one at a time without sleeping between messages
- Improve logging of failed connections
- Bump protocol version to 70002
- Add some additional logging to give extra network insight
- Added new DNS seed from bitcoinstats.com

Validation:

- Log reason for non-standard transaction rejection
- Prune provably-unspendable outputs, and adapt consistency check for it.
- Detect any sufficiently long fork and add a warning
- Call the -alertnotify script when we see a long or invalid fork
- Fix multi-block reorg transaction resurrection
- Reject non-canonically-encoded serialization sizes
- Reject dust amounts during validation
- Accept nLockTime transactions that finalize in the next block

Build system:

- Switch to autotools-based build system
- Build without wallet by passing `--disable-wallet` to configure, this removes the BerkeleyDB dependency
- Upgrade gitian dependencies (libpng, libz, libupnpc, boost, openssl) to more recent versions
- Windows 64-bit build support
- Solaris compatibility fixes
- Check integrity of gitian input source tarballs
- Enable full GCC Stack-smashing protection for all OSes

GUI:

- Switch to Qt 5.2.0 for Windows build
- Add payment request (BIP 0070) support
- Improve options dialog
- Show transaction fee in new send confirmation dialog
- Add total balance in overview page
- Allow user to choose data directory on first start, when data directory is missing, or when the -choosedatadir option is passed
- Save and restore window positions
- Add vout index to transaction id in transactions details dialog
- Add network traffic graph in debug window
- Add open URI dialog
- Add Coin Control Features
- Improve receive coins workflow: make the 'Receive' tab into a form to request payments, and move historical address list functionality to File menu.
- Rebrand to `Bitcoin Core`
- Move initialization/shutdown to a thread. This prevents "Not responding" messages during startup. Also show a window during shutdown.
- Don't regenerate autostart link on every client startup
- Show and store message of normal bitcoin:URI
- Fix richtext detection hang issue on very old Qt versions
- OS X: Make use of the 10.8+ user notification center to display Growl-like notifications

- OS X: Added NSHighResolutionCapable flag to Info.plist for better font rendering on Retina displays.
- OS X: Fix bitcoin-qt startup crash when clicking dock icon
- Linux: Fix Gnome bitcoin: URI handler

Miscellaneous:

- Add Linux script (contrib/qos/tc.sh) to limit outgoing bandwidth
- Add '-regtest' mode, similar to testnet but private with instant block generation with 'setgenerate' RPC.
- Add 'linearize.py' script to contrib, for creating bootstrap.dat
- Add separate bitcoin-cli client

Credits

Thanks to everyone who contributed to this release:

- Andrey
- Ashley Holman
- b6393ce9-d324-4fe1-996b-acf82dbc3d53
- bitsofproof
- Brandon Dahler
- Calvin Tam
- Christian Decker
- Christian von Roques
- Christopher Latham
- Chuck
- coblee
- constantined
- Cory Fields
- Cozz Lovan
- daniel
- Daniel Larimer
- David Hill
- Dmitry Smirnov
- Drak
- Eric Lombrozo
- fanquake
- fcicq
- Florin
- frewil
- Gavin Andresen
- Gregory Maxwell
- gubatron
- Guillermo Céspedes Tabárez
- Haakon Nilsen
- HaltingState

- Han Lin Yap
- harry
- Ian Kelling
- Jeff Garzik
- Johnathan Corgan
- Jonas Schnelli
- Josh Lehan
- Josh Triplett
- Julian Langschaedel
- Kangmo
- Lake Denman
- Luke Dashjr
- Mark Friedenbach
- Matt Corallo
- Michael Bauer
- Michael Ford
- Michagogo
- Midnight Magic
- Mike Hearn
- Nils Schneider
- Noel Tiernan
- Olivier Langlois
- patrick s
- Patrick Strateman
- paveljanik
- Peter Todd
- phantomcircuit
- phelixbtc
- Philip Kaufmann
- Pieter Wuille
- Rav3nPL
- R E Broadley
- regergregregerrge
- Robert Backhaus
- Roman Mindalev
- Rune K. Svendsen
- Ryan Niebur
- Scott Ellis
- Scott Willeke
- Sergey Kazenyuk
- Shawn Wilkinson
- Sined
- sje
- Subo1978
- super3
- Tamas Blummer

- theuni
- Thomas Holenstein
- Timon Rapp
- Timothy Stranex
- Tom Geller
- Torstein Husebø
- Vaclav Vobornik
- vhf / victor felder
- Vinnie Falco
- Warren Togami
- Wil Bown
- Wladimir J. van der Laan