Bitcoin-Qt version 0.8.0 is now available from: http://sourceforge.net/projects/bitcoin/files/Bitcoin/bitcoin-0.8.0/

This is a major release designed to improve performance and handle the increasing volume of transactions on the network.

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), 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 after the upgrade a re-indexing process will be started that will take anywhere from 30 minutes to several hours, depending on the speed of your machine.

Incompatible Changes

This release no longer maintains a full index of historical transaction ids by default, so looking up an arbitrary transaction using the getrawtransaction RPC call will not work. If you need that functionality, you must run once with -txindex=1 -reindex=1 to rebuild block-chain indices (see below for more details).

Improvements

Mac and Windows binaries are signed with certificates owned by the Bitcoin Foundation, to be compatible with the new security features in OSX 10.8 and Windows 8.

LevelDB, a fast, open-source, non-relational database from Google, is now used to store transaction and block indices. LevelDB works much better on machines with slow I/O and is faster in general. Berkeley DB is now only used for the wallet.dat file (public and private wallet keys and transactions relevant to you).

Pieter Wuille implemented many optimizations to the way transactions are verified, so a running, synchronized node uses less working memory and does much less I/O. He also implemented parallel signature checking, so if you have a multi-CPU machine all CPUs will be used to verify transactions.

New Features

“Bloom filter” support in the network protocol for sending only relevant transactions to lightweight clients.

contrib/verifysfbinaries is a shell-script to verify that the binary downloads at sourceforge have not been tampered with. If you are able, you can help make everybody's downloads more secure by running this occasionally to check PGP signatures against download file checksums.

contrib/spendfrom is a python-language command-line utility that demonstrates how to use the "raw transactions" JSON-RPC api to send coins received from particular addresses (also known as "coin control").

New/changed settings (command-line or bitcoin.conf file)

dbcache : controls LevelDB memory usage.

par : controls how many threads to use to validate transactions. Defaults to the number of CPUs on your machine, use -par=1 to limit to a single CPU.

txindex : maintains an extra index of old, spent transaction ids so they will be found by the getrawtransaction JSON-RPC method.

reindex : rebuild block and transaction indices from the downloaded block data.

New JSON-RPC API Features

lockunspent / listlockunspent allow locking transaction outputs for a period of time so they will not be spent by other processes that might be accessing the same wallet.

addnode / getaddednodeinfo methods, to connect to specific peers without restarting.

importprivkey now takes an optional boolean parameter (default true) to control whether or not to rescan the blockchain for transactions after importing a new private key.

Important Bug Fixes

Privacy leak: the position of the "change" output in most transactions was not being properly randomized, making network analysis of the transaction graph to identify users' wallets easier.

Zero-confirmation transaction vulnerability: accepting zero-confirmation transactions (transactions that have not yet been included in a block) from somebody you do not trust is still not recommended, because there will always be ways for attackers to double-spend zero-confirmation transactions. However, this release includes a bug fix that makes it a little bit more difficult for attackers to double-spend a certain type ("lockTime in the future") of zero-confirmation transaction.

Dependency Changes

Qt 4.8.3 (compiling against older versions of Qt 4 should continue to work)

Thanks to everybody who contributed to this release:

Alexander Kjeldaas Andrey Alekseenko Arnav Singh Christian von Roques Eric Lombrozo Forrest Voight Gavin Andresen Gregory Maxwell Jeff Garzik Luke Dashjr Matt Corallo Mike Cassano Mike Hearn Peter Todd Philip Kaufmann Pieter Wuille Richard Schwab Robert Backhaus Rune K. Svendsen Sergio Demian Lerner Wladimir J. van der Laan burger2 default fanquake grimd34th justmoon redshark1802 tucenaber xanatos