

Bitcoin Core version 0.14.1 is now available from:

<https://bitcoin.org/bin/bitcoin-core-0.14.1/>

This is a new minor version release, including 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/>

Compatibility

Bitcoin Core is extensively tested on multiple operating systems using the Linux kernel, macOS 10.8+, and Windows Vista and later.

Microsoft ended support for Windows XP on [April 8th, 2014](#). No attempt is made to prevent installing or running the software on Windows XP, you can still do so at your own risk but be aware that there are known instabilities and issues. Please do not report issues about Windows XP to the issue tracker.

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

Notable changes

RPC changes

- The first positional argument of `createrawtransaction` was renamed from `transactions` to `inputs`.
- The argument of `disconnectnode` was renamed from `node` to `address`.

These interface changes break compatibility with 0.14.0, when the named arguments functionality, introduced in 0.14.0, is used. Client software using these calls with named arguments needs to be updated.

Mining

In previous versions, `getblocktemplate` required segwit support from downstream clients/miners once the feature activated on the network. In this version, it now supports non-segwit clients even after activation, by removing all segwit transactions from the returned block template. This allows non-segwit miners to continue functioning correctly even after segwit has activated.

Due to the limitations in previous versions, `getblocktemplate` also recommended non-segwit clients to not signal for the segwit version-bit. Since this is no longer an issue, `getblocktemplate` now always recommends signalling segwit for all miners. This is safe because ability to enforce the rule is the only required criteria for safe activation, not actually producing segwit-enabled blocks.

UTXO memory accounting

Memory usage for the UTXO cache is being calculated more accurately, so that the configured limit (`-dbcache`) will be respected when memory usage peaks during cache flushes. The memory accounting in prior releases is estimated to only account for half the actual peak utilization.

The default `-dbcache` has also been changed in this release to 450MiB. Users who currently set `-dbcache` to a high value (e.g. to keep the UTXO more fully cached in memory) should consider increasing this setting in order to achieve the same cache performance as prior releases. Users on low-memory systems (such as systems with 1GB or less) should consider specifying a lower value for this parameter.

Additional information relating to running on low-memory systems can be found here: [reducing-bitcoind-memory-usage.md](https://github.com/bitcoin/bitcoin/blob/master/doc/reducing-bitcoind-memory-usage.md).

0.14.1 Change log

Detailed release notes follow. This overview includes changes that affect behavior, not code moves, refactors and string updates. For convenience in locating the code changes and accompanying discussion, both the pull request and git merge commit are mentioned.

RPC and other APIs

- #10084 `142fbb2` Rename first named arg of `createrawtransaction` (MarcoFalke)
- #10139 `f15268d` Remove auth cookie on shutdown (practicalswift)
- #10146 `2fea10a` Better error handling for `submitblock` (rawodb, gmaxwell)
- #10144 `d947afc` `Prioritisetransaction` wasn't always updating ancestor fee (sdaftuar)
- #10204 `3c79602` Rename `disconnectnode` argument (jnewbery)

Block and transaction handling

- #10126 `0b5e162` Compensate for memory peak at flush time (sipa)
- #9912 `fc3d7db` Optimize `GetWitnessHash()` for non-segwit transactions (sdaftuar)
- #10133 `ab864d3` Clean up calculations of `pcoinsTip` memory usage (morcos)

P2P protocol and network code

- #9953/#10013 `d2548a4` Fix shutdown hang with `>= 8` `-addnodes` set (TheBlueMatt)
- #10176 `30fa231` net: gracefully handle `NodeId` wrapping (theuni)

Build system

- #9973 `e9611d1` depends: fix `zlib` build on `osx` (theuni)

GUI

- #10060 `ddc2dd1` Ensure an item exists on the `rpconsole` stack before adding (achow101)

Mining

- #9955/#10006 `569596c` Don't require segwit in `getblocktemplate` for segwit signalling or mining (sdaftuar)
- #9959/#10127 `b5c3440` Prevent slowdown in `CreateNewBlock` on large mempools (sdaftuar)

Tests and QA

- #10157 `55f641c` Fix the `mempool_packages.py` test (sdaftuar)

Miscellaneous

- #10037 4d8e660 Trivial: Fix typo in help getrawtransaction RPC (keystrike)
- #10120 e4c9a90 util: Work around (virtual) memory exhaustion on 32-bit w/ glibc (laanwj)
- #10130 ecc5232 bitcoin-tx input verification (awemany, jnewbery)

Credits

Thanks to everyone who directly contributed to this release:

- Alex Morcos
- Andrew Chow
- Awemany
- Cory Fields
- Gregory Maxwell
- James Evans
- John Newbery
- MarcoFalke
- Matt Corallo
- Pieter Wuille
- practicalswift
- rawodb
- Suhas Daftuar
- Wladimir J. van der Laan

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