0.21.1 Release Notes

Bitcoin Core version 0.21.1 is now available from:

https://bitcoincore.org/bin/bitcoin-core-0.21.1/

This minor release includes 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 in some cases), then run the installer (on Windows) or just copy over `/Applications/Bitcoin-Qt` (on Mac) or `bitcoind/bitcoin-qt` (on Linux).

Upgrading directly from a version of Bitcoin Core that has reached its EOL is possible, but it might take some time if the data directory needs to be migrated. Old wallet versions of Bitcoin Core are generally supported.

Compatibility

Bitcoin Core is supported and extensively tested on operating systems using the Linux kernel, macOS 10.12+, and Windows 7 and newer. Bitcoin Core should also work on most other Unix-like systems but is not as frequently tested on them. It is not recommended to use Bitcoin Core on unsupported systems.

From Bitcoin Core 0.20.0 onwards, macOS versions earlier than 10.12 are no longer supported. Additionally, Bitcoin Core does not yet change appearance when macOS "dark mode" is activated.

Notable changes

Taproot Soft Fork

Included in this release are the mainnet and testnet activation parameters for the taproot soft fork (BIP341) which also adds support for schnorr signatures (BIP340) and tapscript (BIP342).

If activated, these improvements will allow users of single-signature scripts, multisignature scripts, and complex contracts to all use identical-appearing

commitments that enhance their privacy and the fungibility of all bitcoins. Spenders will enjoy lower fees and the ability to resolve many multisig scripts and complex contracts with the same efficiency, low fees, and large anonymity set as single-sig users. Taproot and schnorr also include efficiency improvements for full nodes such as the ability to batch signature verification. Together, the improvements lay the groundwork for future potential upgrades that may improve efficiency, privacy, and fungibility further.

Activation for taproot is being managed using a variation of BIP9 versionbits called Speedy Trial (described in BIP341). Taproot's versionbit is bit 2, and nodes will begin tracking which blocks signal support for taproot at the beginning of the first retarget period after taproot's start date of 24 April 2021. If 90% of blocks within a 2,016-block retarget period (about two weeks) signal support for taproot prior to the first retarget period beginning after the time of 11 August 2021, the soft fork will be locked in, and taproot will then be active as of block 709632 (expected in early or mid November).

Should taproot not be locked in via Speedy Trial activation, it is expected that a follow-up activation mechanism will be deployed, with changes to address the reasons the Speedy Trial method failed.

This release includes the ability to pay taproot addresses, although payments to such addresses are not secure until taproot activates. It also includes the ability to relay and mine taproot transactions after activation. Beyond those two basic capabilities, this release does not include any code that allows anyone to directly use taproot. The addition of taproot-related features to Bitcoin Core's wallet is expected in later releases once taproot activation is assured.

All users, businesses, and miners are encouraged to upgrade to this release (or a subsequent compatible release) unless they object to activation of taproot. If taproot is locked in, then upgrading before block 709632 is highly recommended to help enforce taproot's new rules and to avoid the unlikely case of seeing falsely confirmed transactions.

Miners who want to activate Taproot should preferably use this release to control their signaling. The `getblocktemplate` RPC results will automatically be updated to signal once the appropriate start has been reached and continue signaling until the timeout occurs or taproot activates. Alternatively, miners may manually start signaling on bit 2 at any time; if taproot activates, they will need to ensure they update their nodes before block 709632 or non-upgraded nodes could cause them to mine on an invalid chain. See the versionbits FAQ for details.

For more information about taproot, please see the following resources:

- Technical specifications
  - BIP340 Schnorr signatures for secp256k1
  - BIP341 Taproot: SegWit version 1 spending rules
  - BIP342 Validation of Taproot scripts

- Popular articles;
  - Taproot Is Coming: What It Is, and How It Will Benefit Bitcoin
  - What do Schnorr Signatures Mean for Bitcoin?
  - The Schnorr Signature & Taproot Softfork Proposal
- Development history overview
  - Taproot
  - Schnorr signatures
  - Tapscript
  - Soft fork activation
- Other
  - Questions and answers related to taproot
  - Taproot review

Updated RPCs

- Due to BIP 350 being implemented, behavior for all RPCs that accept addresses is changed when a native witness version 1 (or higher) is passed. These now require a Bech32m encoding instead of a Bech32 one, and Bech32m encoding will be used for such addresses in RPC output as well. No version 1 addresses should be created for mainnet until consensus rules are adopted that give them meaning (e.g. through BIP 341). Once that happens, Bech32m is expected to be used for them, so this shouldn't affect any production systems, but may be observed on other networks where such addresses already have meaning (like signet).

0.21.1 change log

Consensus

- #21377 Speedy trial support for versionbits (ajtowns)
- #21686 Speedy trial activation parameters for Taproot (achow101)

P2P protocol and network code

- #20852 allow CSubNet of non-IP networks (vasild)
- #21043 Avoid UBSan warning in ProcessMessage(...) (practicalswift)

Wallet

- #21166 Introduce DeferredSignatureChecker and have SignatureExtractor-Class subclass it (achow101)
- #21083 Avoid requesting fee rates multiple times during coin selection (achow101)

RPC and other APIs

- #21201 Disallow sendtoaddress and sendmany when private keys disabled (achow101)

Build system

- #21486 link against -lsocket if required for `*ifaddrs` (fanquake)
- #20983 Fix MSVC build after gui#176 (hebasto)

Tests and QA

- #21380 Add fuzzing harness for versionbits (ajtowns)
- #20812 fuzz: Bump FuzzedDataProvider.h (MarcoFalke)
- #20740 fuzz: Update FuzzedDataProvider.h from upstream (LLVM) (practicalswift)
- #21446 Update vcpkg checkout commit (sipsorcery)
- #21397 fuzz: Bump FuzzedDataProvider.h (MarcoFalke)
- #21081 Fix the unreachable code at `feature_taproot` (brunoerg)
- #20562 Test that a fully signed tx given to signrawtx is unchanged (achow101)
- #21571 Make sure non-IP peers get discouraged and disconnected (vasild, MarcoFalke)
- #21489 fuzz: cleanups for versionbits fuzzer (ajtowns)

Miscellaneous

- #20861 BIP 350: Implement Bech32m and use it for v1+ segwit addresses (sipa)

Documentation

- #21384 add signet to bitcoin.conf documentation (jonatack)
- #21342 Remove outdated comment (hebasto)

Credits

Thanks to everyone who directly contributed to this release:

- Aaron Clauson
- Andrew Chow
- Anthony Towns
- Bruno Garcia
- Fabian Jahr
- fanquake
- Hennadii Stepanov
- Jon Atack

- Luke Dashjr
- MarcoFalke
- Pieter Wuille
- practicalswift
- randymcmillan
- Sjors Provoost
- Vasil Dimov
- W. J. van der Laan

As well as to everyone that helped with translations on Transifex.