Bitcoin Core version 0.12.1 is now available from:

https://bitcoin.org/bin/bitcoin-core-0.12.1/

This is a new minor version release, including the BIP9, BIP68 and BIP112 softfork, various bugfixes and updated translations.

Please report bugs using the issue tracker at github:

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

Upgrading and downgrading

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).

Downgrade warning

Downgrade to a version < 0.12.0

Because release 0.12.0 and later will obfuscate the chainstate on every fresh sync or reindex, the chainstate is not backwards-compatible with pre-0.12 versions of Bitcoin Core or other software.

If you want to downgrade after you have done a reindex with 0.12.0 or later, you will need to reindex when you first start Bitcoin Core version 0.11 or earlier.

Notable changes

First version bits BIP9 softfork deployment

This release includes a soft fork deployment to enforce BIP68, BIP112 and BIP113 using the BIP9 deployment mechanism.

The deployment sets the block version number to 0x20000001 between midnight 1st May 2016 and midnight 1st May 2017 to signal readiness for deployment. The version number consists of 0x20000000 to indicate version bits together with setting bit 0 to indicate support for this combined deployment, shown as “csv” in the `getblockchaininfo` RPC call.

For more information about the soft forking change, please see https://github.com/bitcoin/bitcoin/pull/7648

This specific backport pull-request can be viewed at https://github.com/bitcoin/bitcoin/pull/7543

BIP68 soft fork to enforce sequence locks for relative lock-time

BIP68 introduces relative lock-time consensus-enforced semantics of the sequence number field to enable a signed transaction input to remain invalid for a defined period of time after confirmation of its corresponding outpoint.

For more information about the implementation, see https://github.com/bitcoin/bitcoin/pull/7184

BIP112 soft fork to enforce OP_CHECKSEQUENCEVERIFY

BIP112 redefines the existing OP_NOP3 as OP_CHECKSEQUENCEVERIFY (CSV) for a new opcode in the Bitcoin scripting system that in combination with BIP68 allows execution pathways of a script to be restricted based on the age of the output being spent.

For more information about the implementation, see https://github.com/bitcoin/bitcoin/pull/7524

BIP113 locktime enforcement soft fork

Bitcoin Core 0.11.2 previously introduced mempool-only locktime enforcement using GetMedianTimePast(). This release seeks to consensus enforce the rule.

Bitcoin transactions currently may specify a locktime indicating when they may be added to a valid block. Current consensus rules require that blocks have a block header time greater than the locktime specified in any transaction in that block.

Miners get to choose what time they use for their header time, with the consensus rule being that no node will accept a block whose time is more than two hours in the future. This creates a incentive for miners to set their header times to future values in order to include locktimed transactions which weren't supposed to be included for up to two more hours.

The consensus rules also specify that valid blocks may have a header time greater than that of the median of the 11 previous blocks. This GetMedianTimePast() time has a key feature we generally associate with time: it can't go backwards.

BIP113 specifies a soft fork enforced in this release that weakens this perverse incentive for individual miners to use a future time by requiring that valid blocks have a computed GetMedianTimePast() greater than the locktime specified in any transaction in that block.

Mempool inclusion rules currently require transactions to be valid for immediate inclusion in a block in order to be accepted into the mempool. This release begins applying the BIP113 rule to received transactions, so transaction whose time is greater than the GetMedianTimePast() will no longer be accepted into the mempool.

Implication for miners: you will begin rejecting transactions that would not be valid under BIP113, which will prevent you from producing invalid blocks when BIP113 is enforced on the network. Any transactions which are valid under the current rules but not yet valid under the BIP113 rules will either be mined by other miners or delayed until they are valid under BIP113. Note, however, that time-based locktime transactions are more or less unseen on the network currently.

Implication for users: GetMedianTimePast() always trails behind the current time, so a transaction locktime set to the present time will be rejected by nodes running this release until the median time moves forward. To compensate, subtract one hour (3,600 seconds) from your locktimes to allow those transactions to be included in mempools at approximately the expected time.

For more information about the implementation, see https://github.com/bitcoin/bitcoin/pull/6566

Miscellaneous

The p2p alert system is off by default. To turn on, use `-alert` with startup configuration.

0.12.1 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

- #7739 `7ffc2bd` Add abandoned status to listtransactions (jonasschnelli)

Block and transaction handling

- #7543 `834aaef` Backport BIP9, BIP68 and BIP112 with softfork (btcdrak)

P2P protocol and network code

- #7804 `90f1d24` Track block download times per individual block (sipa)
- #7832 `4c3a00d` Reduce block timeout to 10 minutes (laanwj)

Validation

- #7821 `4226aac` init: allow shutdown during 'Activating best chain…' (laanwj)

- #7835 `46898e7` Version 2 transactions remain non-standard until CSV activates (sdaftuar)

Build system

- #7487 `00d57b4` Workaround Travis-side CI issues (luke-jr)
- #7606 `a10da9a` No need to set -L and –location for curl (MarcoFalke)
- #7614 `ca8f160` Add curl to packages (now needed for depends) (luke-jr)
- #7776 `a784675` Remove unnecessary executables from gitian release (laanwj)

Wallet

- #7715 `19866c1` Fix calculation of balances and available coins. (morcos)

Miscellaneous

- #7617 `f04f4fd` Fix markdown syntax and line terminate LogPrint (MarcoFalke)
- #7747 `4d035bc` added depends cross compile info (accraze)
- #7741 `a0cea89` Mark p2p alert system as deprecated (btcdrak)
- #7780 `c5f94f6` Disable bad-chain alert (btcdrak)

Credits

Thanks to everyone who directly contributed to this release:

- accraze
- Alex Morcos
- BtcDrak
- Jonas Schnelli
- Luke Dashjr
- MarcoFalke
- Mark Friedenbach
- NicolasDorier
- Pieter Wuille
- Suhas Daftuar
- Wladimir J. van der Laan

As well as everyone that helped translating on Transifex.