

Segwit in Bitcoin: Lessons Learned

Gregory Sanders

Myself

- [Elements Project](#)
 - [Elements Alpha](#)
 - [Liquid](#)
- Did review for Segwit in Core
 - Upstream review important for downstream
- Scaling(?)
 - How do we scale protocol development?

Segwit as a Solution

- First developed as an “element” of [Elements Alpha](#)
- Solves the problem of unintentional malleability.
 - TL;DR Allows safe chaining of pre-signed transactions for smart contracting in Bitcoin.
 - Payment Channels, Lightning
- But doesn't fit into Bitcoin really.
 - We can't just change txid formulation on a whim
 - No matter the technical benefits, hard to imagine non-security-fix HF

Segwit as Deployed

- Key insight: If you can deploy a type of “extension block”. Inside this extension nearly any rules can be enforced, turning hard forks into soft forks
- Many ideas like Confidential Transactions as Softfork
- Segwit, a highly desired extension, fits the bill
 - Special new transaction types are signaled via empty scriptSig and commitment to the witnessScript in scriptPubkey
 - Think P2SH, but hiding the new data inside the extension block
 - Now signatures are out of txid calculation for both new and old nodes.
 - Backwards compatible
- [Old news now](#)

Isn't Segwit already done?

- Segwit is active in testnet, and close to release in mainnet
- One of the largest changes to Bitcoin ever
 - Touched nearly every part of the codebase: serialization, p2p, wallet, consensus
- Talking about this has two purposes:
 - Informational for those not privy to the sausage-making
 - What I see as takeaways from the exercise
- Any opinions are just mine

Minimum Viable Product (What didn't make it)

- New address format([BIP142](#))
 - Hesitancy to perpetuate base58+checksum
 - Nested P2SH for now
- Additional tweaks to commitment structure
 - [Arbitrary segwit commitment tree and fast hashing](#)
 - [Previous block witness commitment](#)
- [Validation Cost Metric](#)
- Fraud Proofs
- New scripting*

* for the most part

Minimum Viable Product

- Each proposal needs a champion
- Each proposal increases demand for review
 - Already strapped for review as-is
- Even “too many BIP numbers” can be a problem
 - Downstream developers can't figure out why signatures are failing (BIP143)
 - People still expecting BIP142 addresses

Sidenote: Scripting differences / similarities

- CHECKMULTISIG still requires an additional dummy argument in the stack
 - Null dummy softfork: [#8636](#)
- Sighash serialization overhaul ([BIP143](#))
 - O(n) hashing
 - Value under hash!
- “Minimal if” as policy: [#8526](#)
- No uncompressed pubkeys as policy [#8499](#) (?)
- Nullfail as policy [#8634](#)
- Fixes SIGHASH_SINGLE “one” [bug](#)
- [Low_s softfork](#)
- Script versioning

Segwit Developed

- #segwit-dev (still ~52 users there, for some reason)
- Contention about where it should be discussed
 - Mailing list used to announce BIPs, major changes
 - Hesitancy to flood #bitcoin(-core)-dev
 - Further partitioning of IRC development presence
- Bulk of design done pre-PR
- 4 segnet iterations, starting with segnet1 in Dec. 2015
 - Were actively used by downstream developers
 - Spam 4MB blocks

Segwit PR'd

- 32 participants
- April 9th to June 24th
- The branch where comments were targeted
- ~1,486 lines of code for implementation
- ~3,338 lines of code for tests

Segregated witness #7910

 **Closed** sipa wants to merge 128 commits into bitcoin:master from sipa:segwit-master

 Conversation 145

 Commits 128

 Files changed 80

+5,305 -571 

Segwit Rebased

- Identical diff
- June 6 to June 26
- 0.13, Compact Blocks, and Segwit
 - Contention on when each should be merged
 - 0.12 backport promises?
- Merged, activated on testnet

Segregated witness rebased #8149



iaanwj merged 27 commits into bitcoin:master from sipa:segwit-master2 on Jun 24

Conversation 24

Commits 27

Files changed 80

+5,305 -571

Segwit merged to master. Done!

bitcoin / bitcoin

Watch 1,214 Unstar 10,023 Fork 6,661

<> Code Issues 391 Pull requests 131 Projects 6 Pulse Graphs

0.13.1

New issue

No due date 91% complete

5 Open 53 Closed

- [qa] mininode: Fix order of positional args in wait_until ✓ Needs backport Tests 4
#8857 opened 12 hours ago by MarcoFalke
- Add NULLDUMMY verify flag in bitcoinconsensus.h ✓ Consensus Needs backport 2
#8848 opened a day ago by jl2012
- update bitcoin-tx to output witness data ✓ Needs backport Utils and libraries 7
#8817 opened 5 days ago by jnewbery
- Support for compact blocks together with segwit ✓ Needs backport P2P 51
#8393 opened on Jul 22 by sipa
- Add several policy limits for segwit scripts ✓ Needs backport 89
#8499 opened on Aug 11 by jl2012

5 Open 53 Closed

- [qa] Fix race condition in p2p-compactblocks test ✓ Needs backport Tests 1
#8654 by sdatfuar was merged 13 hours ago
- bitcoin-util-test.py should fail if the output file is empty ✓ Needs backport Tests 2
#8636 by jnewbery was merged a day ago
- test: Avoid ConnectionResetErrors during RPC tests ✓ Needs backport Tests 1
#8839 by iaanwj was merged a day ago
- [qa] fix nulldummy test ✓ Needs backport Tests 3
#8841 by jl2012 was merged a day ago
- [qa] blockstore: Switch to dumb dbm ✓ Needs backport Tests 8
#8834 by MarcoFalke was merged 2 days ago
- [qa] nulldummy.py: Don't run unused code ✓ Needs backport Tests 3
#8835 by MarcoFalke was merged 2 days ago
- [qa] Split up slow RPC calls to avoid pruning test timeouts ✓ Needs backport Refactoring Tests 2
#8827 by sdatfuar was merged 2 days ago
- Add bitcoin-tx JSON tests ✓ Needs backport Tests 11
#8829 by jnewbery was merged 3 days ago
- [rpc] throw JSONRPCError when utxo set can not be read ✓ Needs backport RPC/REST/ZMQ 2
#8832 by MarcoFalke was merged 3 days ago
- Add policy: null signature for failed CHECK(MULTI)SIG × Needs backport TX fees and policy 22
#8634 by jl2012 was merged 4 days ago
- Make non-minimal OP_IFNOTIF argument non-standard for P2WSH ✓ Needs backport 20
#8526 by jl2012 was merged 4 days ago
- Backports for 0.13.1 ✓ Backport 2
#8815 by iaanwj was merged 4 days ago
- Ping regularly in p2p-segwit.py to keep connection alive ✓ Needs backport Tests 5
#8803 by jl2012 was merged 4 days ago
- 0.13.0 is no longer compatible with OSX 10.7 Mac OSX 5
#8577 by jonasschnelli was closed 5 days ago
- Implement NULL DUMMY softfork (RIP147) ✓ Consensus 14

Backport Backlog

























- Fix Segwit transaction blinding via reject filter
 - Spammers could temporarily stop a transaction from propagating
- Compact blocks for Segwit
 - Plus versioning negotiation
- Segwit wallet cleanups
- Softfork/policy follow-ons
- Slew of bugfixing backports
 - <https://github.com/bitcoin/bitcoin/milestone/22>
- Weeks of somewhat tedious irc dev meetings
- Getting close!

Proposal(s)

- Any non-trivial consensus change to Bitcoin in the future should have an actively-used testnet spun up.
 - If supposed downstream users aren't actively testing, is the change even desired?
 - Regular (ab)use helps to surface issues early
- Ride-along changes should be discussed, implemented, and tested as early as possible into the development cycle.
 - Other issues will surely pop up
- Tests should take up a large fraction of the loc changes
- Any additional technical channels should be carefully spun up, logged, and spun down at appropriate times
 - Avoid loss of design history, communal knowledge

Proposal(s)

- If a PR spans a number of layers:
 - Keep commits in logical partitions
 - Split sections with empty commits marking start/end

 --- [SEGWIT] begin: P2P/node/consensus --- sipa committed on Mar 31	 ecacf d9	
 BIP143: Verification logic ... sipa committed on Dec 27, 2015	 3dd4102	
 BIP141: Other consensus critical limits, and BIP145 ... sipa committed on Jan 3	 2b1f6f9	
 [RPC] Return witness data in blockchain RPCs ... jj2012 committed with sipa on Jan 22	 7c4bf77	
 [libconsensus] Script verification API with amounts ... afk11 committed with sipa on Jan 24	 b7dbeb2	
 Add rewind logic to deal with post-fork software updates ... sipa committed on Mar 18	 6032f69	
 --- [SEGWIT] begin: wallet --- sipa committed on Mar 31	 9757b57	
 Refactor script validation to observe amounts ... sipa committed on Mar 31	 0ef1dd3	

Proposal(s)

- Stop talking about block size
 - Let's talk about "weight" and "throughput"
- Only spend time backporting major consensus changes when there is demand
- Don't do large changes like this often
 - Higher amount of risk compared to usual
 - Slows other technical debt cleanup (libconsensus, network refactor, etc)
 - Is more uninviting for review
 - Collides into regular release schedule, causes confusion/tension

Thanks!

Softforks with increased risks

- Segwit nodes must find sufficient number of upgraded nodes
 - You may be partitioned off the network if you can't find peers to serve new data
 - Same with any “extension block” style softfork
 - P2SH et al. suffer no such risks
- Mitigations:
 - Preferential connections used to mitigate
 - Find more compatible peers faster
 - [“feeler” connections](#)
- Need a long lead-time to ensure safety
- Punish bad peer behavior without redoing all networking code