BIP 0141 - Segregated Witness (Consensus layer) (soft fork)

return to the Bitcoin Improvement Proposals

Table 1: Data sheet for Segregated Witness (Consensus layer)

Title	Segregated Witness (Consensus layer)
Layer	Consensus (soft fork)
Author	Eric Lombrozo, Johnson Lau, Pieter Wuille
Comments-Summary	No comments yet.
Comments-URI	https://github.com/bitcoin/bips/wiki/Comments:BIP-0141
Status	Final
Type	Standards Track
Created	2015-12-21
Post History	
Description	https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki
License	PD

> **Note**: The following is an excerpt from the official Bitcoin site. It is provided here as a convenience and is not authoritative. Refer to the original document(s) as the authoritative reference.

Abstract

This BIP defines a new structure called a “witness” that is committed to blocks separately from the transaction merkle tree. This structure contains data required to check transaction validity but not required to determine transaction effects. In particular, scripts and signatures are moved into this new structure.

The witness is committed in a tree that is nested into the block's existing merkle root via the coinbase transaction for the purpose of making this BIP soft fork compatible. A future hard fork can place this tree in its own branch.

Motivation

The entirety of the transaction's effects are determined by output consumption (spends) and new output creation. Other transaction data, and signatures in particular, are only required to validate the blockchain state, not to determine it.

By removing this data from the transaction structure committed to the transaction merkle tree, several problems are fixed:

1. Nonintentional malleability becomes impossible. Since signature data is no longer part of the transaction hash, changes to how the transaction was signed are no longer relevant to transaction identification. As a solution of transaction malleability, this is superior to the canonical signature *approach (BIP62):*

- *It prevents involuntary transaction malleability for any type of scripts, as long as all inputs are signed (with at least one CHECKSIG or CHECKMULTISIG operation)*
- *In the case of an m-of-n CHECKMULTISIG script, a transaction is malleable only with agreement of private key holders (as opposed to only 1 private key holder with BIP62)*
- *It prevents involuntary transaction malleability due to unknown ECDSA signature malleability*
- *It allows creation of unconfirmed transaction dependency chains without counterparty risk, an important feature for offchain protocols such as the Lightning Network*

2. Transmission of signature data becomes optional. It is needed only if a peer is trying to validate a transaction instead of just checking its existence. This reduces the size of Simple Payment Verification (SPV) proofs and potentially improves the privacy of SPV clients as they can download more transactions using the same bandwidth.
3. Some constraints could be bypassed with a soft fork by moving part of the transaction data to a structure unknown to current protocol, for example:

- *Size of witness could be ignored / discounted when calculating the block size, effectively increasing the block size to some extent*
- *Hard coded constants, such as maximum data push size (520 bytes) or sigops limit could be reevaluated or removed*
- *New script system could be introduced without any limitation from the existing script semantic. For example, a new transaction digest algorithm for transaction signature verification is described in BIP143.*