```
BIP: 144
Layer: Peer Services
Title: Segregated Witness (Peer Services)
Author: Eric Lombrozo <elombrozo@gmail.com>
        Pieter Wuille <pieter.wuille@gmail.com>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0144
Status: Final
Type: Standards Track
Created: 2016-01-08
License: PD
```

Abstract

This BIP defines new messages and serialization formats for propagation of transactions and blocks committing to segregated witness structures.

Motivation

In addition to defining witness structures and requiring commitments in future blocks (BIP141 - Consensus segwit BIP), new mechanisms must be defined to allow peers to advertise support for segregated witness and to relay the witness structures and request them from other peers without breaking compatibility with older nodes.

Specification

Serialization

A new serialization format for tx messages is added to the peer-to-peer protocol.

The serialization has the following structure:

Field Size	Name	Type	Description
4	version	int32_t	Transaction data format version
1	marker	char	Must be zero
1	flag	char	Must be nonzero
1+	txin_count	var_int	Number of transaction inputs
41+	txins	txin[]	A list of one or more transaction inputs
1+	txout_count	var_int	Number of transaction outputs
9+	txouts	txouts[]	A list of one or more transaction outputs
1+	script_witnesses	script_witnesses[]	The witness structure as a serialized byte array
4	lock_time	uint32_t	The block number or timestamp until which the transacti

Parsers supporting this BIP will be able to distinguish between the old serialization format (without the witness) and this one. The marker byte is set to

zero so that this structure will never parse as a valid transaction in a parser that does not support this BIP. If parsing were to succeed, such a transaction would contain no inputs and a single output.

If the witness is empty, the old serialization format must be used.

Currently, the only witness objects type supported are script witnesses which consist of a stack of byte arrays. It is encoded as a var_int item count followed by each item encoded as a var_int length followed by a string of bytes. Each txin has its own script witness. The number of script witnesses is not explicitly encoded as it is implied by txin_count. Empty script witnesses are encoded as a zero byte. The order of the script witnesses follows the same order as the associated txins.

- **Rationale for not having an independent message type with its own serialization**: this would require separate "tx" and "block" messages, and all RPC calls operating on raw transactions would need to be duplicated, or need inefficient or nondeterministic guesswork to know which type is to be used.
- **Rationale for not using just a single 0x00 byte as marker**: that would lead to empty transactions (no inputs, no outputs, which are used in some tests) to be interpreted as new serialized data.
- **Rationale for the 0x01 flag byte in between**: this will allow us to easily add more extra non-committed data to transactions (like txouts being spent, ...). It can be interpreted as a bitvector.

Handshake

A node will signal that it can provide witnesses using the following service bit

```
NODE_WITNESS = (1 << 3)
```

Hashes

Transaction hashes used in the transaction merkle tree and txin outpoints are always computed using the old non-witness serialization.

Support for a new hash including the witness data is added that is computed from the new witness serialization. (Note that transactions with an empty witness always use the old serialization, and therefore, they have witness hash equal to normal hash.)

Relay

New inv types MSG_WITNESS_TX (0x40000001, or (1«30)+MSG_TX) and MSG_WITNESS_BLOCK (0x40000002, or (1«30)+MSG_BLOCK) are added, only for use in getdata. Inventory messages themselves still use just

MSG_TX and MSG_BLOCK, similar to MSG_FILTERED_BLOCK. A further inv type MSG_FILTERED_WITNESS_BLOCK (0x40000003, or (1«30)+MSG_FILTERED_BLOCK) is reserved for future use.

- **Rationale for not advertizing witnessness in invs**: we don't always use invs anymore (with 'sendheaders' BIP 130), plus it's not useful: implicitly, every transaction and block have a witness, old ones just have empty ones.

MSG_WITNESS_TX getdata requests should use the non-witness serialized hash. The peer shall respond with a tx message, and if the witness structure is nonempty, the witness serialization shall be used.

MSG_WITNESS_BLOCK requests will return a block message with transactions that have a witness using witness serialization.

Credits

Special thanks to Gregory Maxwell for originating many of the ideas in this BIP and Luke-Jr for figuring out how to deploy this as a soft fork.

Reference Implementation

https://github.com/bitcoin/bitcoin/pull/8149

Copyright