```
BIP: 145
Layer: API/RPC
Title: getblocktemplate Updates for Segregated Witness
Author: Luke Dashjr <luke+bip22@dashjr.org>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0145
Status: Final
Type: Standards Track
Created: 2016-01-30
License: BSD-2-Clause
         OPL
```

Abstract

This BIP describes modifications to the getblocktemplate JSON-RPC call (BIP 22) to support segregated witness as defined by BIP 141.

Specification

Block Template

The template Object is revised to include a new key:

template
Key
weightlimit

The '!' rule prefix MUST be enabled on the "segwit" rule for templates including transactions with witness data. In particular, note that even if the client's "rules" list lacks "segwit", server MAY support old miners by producing a witness-free template and omitting the '!' rule prefix for "segwit" in the template's "rules" list. If the GBT server does not support producing witness-free templates after its activation, it must also use the '!' rule prefix in the "vbavailable" list prior to activation.

Transactions Object Format The Objects listed in the response's "transactions" key is revised to include these keys:

template "transactions" element
Key
txid
weight
hash

Transactions with witness data may only be included if the template's "rules" list (see BIP 9) includes "segwit".

Sigops

For templates with "segwit" enabled as a rule, the "sigoplimit" and "sigops" keys must use the new values as calculated in BIP 141.

Block Assembly with Witness Transactions

When block assembly is done without witness transactions, no changes are made by this BIP, and it should be assembled as previously.

When witness transactions are included in the block, the primary merkle root MUST be calculated with those transactions' "txid" field instead of "hash". A secondary merkle root MUST be calculated as per BIP 141's commitment structure specification to be inserted into the generation (coinbase) transaction.

Servers MUST NOT include a commitment in the "coinbasetxn" key on the template. Clients MUST insert the commitment as an additional output at the end of the final generation (coinbase) transaction. Only if the template includes a "mutable" key (see BIP 23 Mutations) including "generation", the client MAY in that case place the commitment output in any position it chooses, provided that no later output matches the commitment pattern.

Motivation

Segregated witness substantially changes the structure of blocks, so the previous getblocktemplate specification is no longer sufficient. It additionally also adds a new way of counting resource limits, and so GBT must be extended to convey this information correctly as well.

Rationale

Why doesn't "weightlimit" simply redefine the existing "sizelimit"?

- "sizelimit" is already enforced by clients by counting the sum of bytes in transactions' "data" keys.
- Servers may wish to limit the overall size of a block, independently from the "weight" of the block.

Why is "sigoplimit" redefined instead of a new "sigopweightlimit" being added?

- The old limit was already arbitrarily defined, and could not be counted by clients on their own anyway. The concept of "sigop weight" is merely a change in the arbitrary formula used.

Why is "sigoplimit" divided by 4?

- To resemble the previous values. (FIXME: is this a good reason? maybe we shouldn't divide it?)

Why is the witness commitment required to be added to the end of the generation transaction rather than anywhere else?

- Servers which do not allow modification of the generation outputs ought to be checking this as part of the validity of submissions. By requiring a specific placement, they can simply strip the commitment and do a byte-for-byte comparison of the outputs. Placing it at the end avoids the possibility of a later output matching the pattern and overriding it.

Why shouldn't the server simply add the commitment upfront in the "coinbasetxn", and simply send the client stripped transaction data?

- It would become impossible for servers to specify only "coinbasevalue", since clients would no longer have the information required to construct the commitment.
- getblocktemplate is intended to be a *decentralised* mining protocol, and allowing clients to be blinded to the content of the block works contrary to that purpose.
- BIP 23's "transactions" mutations allow the client to modify the transaction-set on their own, which is impossible without the complete transaction data.

Reference Implementation

- libblkmaker
- Eloipool
- Bitcoin Core

See Also

- BIP 9: Version bits with timeout and delay
- BIP 22: getblocktemplate - Fundamentals
- BIP 23: getblocktemplate - Pooled Mining
- BIP 141: Segregated Witness (Consensus layer)

Copyright