```
BIP: 142
Layer: Applications
Title: Address Format for Segregated Witness
Author: Johnson Lau <jl2012@xbt.hk>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0142
Status: Withdrawn
Type: Standards Track
Created: 2015-12-24
License: PD
```

Abstract

This BIP describes new types of Bitcoin address to support native segregated witness transactions with 20-byte and 32-byte program.

Motivation

To define standard payment address for native segregated witness (segwit) transactions to promote early adoption of the more efficient transaction method.

Specification

The new Bitcoin address format defined is for the Pay-to-Witness-Public-Key-Hash (P2WPKH) and Pay-to-Witness-Script-Hash (P2WSH) transaction described in segregated witness soft fork (BIP141). The scriptPubKey is an OP_0 followed by a push of 20-byte-hash (P2WPKH) or 32-byte hash (P2WSH).

The new address is encoded in a way similar to existing address formats:

```
 base58-encode:
[1-byte address version]
[1-byte witness program version]
[0x00]
[20/32-byte-hash]
[4-byte checksum]
```

For P2WPKH address, the address version is 6 (0x06) for a main-network address or 3 (0x03) for a testnet address.

For P2WSH address, the address version is 10 (0x0A) for a main-network address or 40 (0x28) for a testnet address.

The witness program version is a 1-byte value between 0 (0x00) and 16 (0x10). Only version 0 is defined in BIP141. Versions 1 to 16 are reserved for future extensions.

Following the witness program version is a 0x00 padding to make sure that each witness program version will have a unique prefix.

Following the padding is the program hash, 20 byte for a P2WPKH address and 32 byte for a P2WSH address.

The 4-byte checksum is the first four bytes of the double SHA256 hash of the serialization of the previous items.

All addresses generated with this scheme will have a constant length, with 36 digits for 20-byte and 53 digits for 32-byte. Different witness program versions will have a unique prefix, as shown in the following table:

rowspan=3 style=""|Witness program version	colspan=4 style=""|Hash size
colspan=2 style=""|20-byte (36 characters)	colspan=2 style=""|32-byte (53 characters)
Mainnet	Testnet
0	p2
1	p4
2	p6
3	p7
4	pA
5	pB
6	pD
7	pF
8	pG
9	pJ
10	pL
11	pN
12	pQ
13	pS
14	pT
15	pV
16	pX

Rationale

BIP141 defines 2 ways of encoding a "witness program", a data push of 2 to 32 bytes:

- A native witness program output is a scriptPubKey with a push of version byte followed by a push of witness program, and nothing else;
- Segwit-in-P2SH is a BIP16 P2SH redeemScript with a push of version byte followed by a push of witness program, while the scriptPubKey looks like a normal P2SH output.

Considering the BIP13 P2SH address has been defined in 2012, using segwit-in-P2SH allows most existing wallets to pay a segwit-compatible wallet without any upgrade. However, this method requires more block space and is only a short-term solution to make the transition smoother. Eventually, all users are expected to use the more efficient native witness program as the primary method of payment.

The drawbacks of Bitcoin addresses have been extensively discussed in BIP13. Since then, better payment methods have been proposed or deployed, for example:

- BIP47 Reusable Payment Codes for Hierarchical Deterministic Wallets
- BIP63 Stealth Addresses
- BIP70 Payment protocol

However, none of these are as widely adopted as the suboptimal base-58 script-PubKey template addresses, which is still a standard for the whole eco-system, from wallets, block explorers, merchants, exchanges, to end users. It is believed that the proposed P2WPKH and P2WSH address format is the easiest way for wallets and services to adopt native witness program, which is particularly important in the context of scaling the capacity of the blockchain.

While P2WPKH address is specific for simple payment to a single public key, P2WSH address allows arbitrarily complex segwit transactions, similar to the BIP13 P2SH address.

Compatibility

This proposal is not backward-compatible. However, an older implementation will report the new address type as invalid, and refuse to create a transaction.

This proposal is forward-compatible with future versions of witness programs of 20 and 32 bytes.

Example

The following public key,

```
0450863AD64A87AE8A2FE83C1AF1A8403CB53F53E486D8511DAD8A04887E5B23522CD470243453A299FA9E772
```

when encoded as a P2PKH template, would become:

```
DUP HASH160 <010966776006953D5567439E5E39F86A0D273BEE> EQUALVERIFY CHECKSIG
```

With the corresponding version 1 Bitcoin address being:

```
16UwLL9Risc3QfPqBUvKofHmBQ7wMtjvM
```

When the same public key is encoded as P2WPKH, the scriptPubKey becomes:

```
OP_0 <010966776006953D5567439E5E39F86A0D273BEE>
```

Using 0x06 as address version, followed by 0x00 as witness program version, and a 0x00 padding, the equivalent P2WPKH address is:

```
p2xtZoXeX5X8BP8JfFhQK2nD3emtjch7UeFm
```

Reference implementation

https://github.com/theuni/bitcoin/commit/ede1b57058ac8efdefe61f67395affb48f2c0d80

References

- BIP 13: Address Format for pay-to-script-hash
- BIP 16: Pay to Script Hash
- BIP 70: Payment Protocol
- BIP 141: Segregated Witness

Copyright