```
BIP: 84
Layer: Applications
Title: Derivation scheme for P2WPKH based accounts
Author: Pavol Rusnak <stick@satoshilabs.com>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0084
Status: Final
Type: Informational
Created: 2017-12-28
License: CC0-1.0
```

Abstract

This BIP defines the derivation scheme for HD wallets using the P2WPKH (BIP 173) serialization format for segregated witness transactions.

Motivation

With the usage of P2WPKH transactions it is necessary to have a common derivation scheme. It allows the user to use different HD wallets with the same masterseed and/or a single account seamlessly.

Thus the user needs to create dedicated segregated witness accounts, which ensures that only wallets compatible with this BIP will detect the accounts and handle them appropriately.

Considerations

We use the same rationale as described in Considerations section of BIP 49.

Specifications

This BIP defines the two needed steps to derive multiple deterministic addresses based on a BIP 32 root account.

Public key derivation

To derive a public key from the root account, this BIP uses the same account-structure as defined in BIP 44 and BIP 49, but only uses a different purpose value to indicate the different transaction serialization method.

`m / purpose' / coin_type' / account' / change / address_index`

For the `purpose`-path level it uses `84'`. The rest of the levels are used as defined in BIP44 or BIP49.

Address derivation

To derive the P2WPKH address from the above calculated public key, we use the encapsulation defined in BIP 141:

```
   witness:
scriptSig:    (empty)
scriptPubKey: 0 <20-byte-key-hash>
(0x0014{20-byte-key-hash})
```

Extended Key Version

When serializing extended keys, this scheme uses alternate version bytes. Extended public keys use `0x04b24746` to produce a "zpub" prefix, and private keys use `0x04b2430c` to produce a "zprv" prefix. Testnet uses `0x045f1cf6` "vpub" and `0x045f18bc` "vprv."

Additional registered version bytes are listed in SLIP-0132.

Backwards Compatibility

This BIP is not backwards compatible by design as described under [#considerations]. An incompatible wallet will not discover accounts at all and the user will notice that something is wrong.

Test vectors

```
mnemonic = abandon abandon abandon abandon abandon abandon abandon abandon abandon abandon
rootpriv = zprvAWgYBBk7JR8Gjrh4UJQ2uJdG1r3WNRRfURiABBE3RvMXYSrRJL62XuezvGdPvG6GFBZduosCc1Y
rootpub  = zpub6jftahH18ngZxLmXaKw3GSZzZsszmt9WqedkyZdezFtWRFBZqsQH5hyUmb4pCEeZGmVfQuP5bed

// Account 0, root = m/84'/0'/0'
xpriv = zprvAdG4iTXWBoARxkkzNpNh8r6Qag3irQB8PzEMkAFeTRXxHpbF9z4QgEvBRmfvqWvGp42t42nvgGpNgY
xpub  = zpub6rFR7y4Q2AijBEqTUquhVz398htDFrtymD9xYYfG1m4wAcvPhXNfE3EfH1r1ADqtfSdVCToUG868Rv

// Account 0, first receiving address = m/84'/0'/0'/0/0
privkey = KyZpNDKnfs94vbrwhJneDi77V6jF64PWPF8x5cdJb8ifgg2DUc9d
pubkey  = 0330d54fd0dd420a6e5f8d3624f5f3482cae350f79d5f0753bf5beef9c2d91af3c
address = bc1qcr8te4kr609gcawutmrza0j4xv80jy8z306fyu

// Account 0, second receiving address = m/84'/0'/0'/0/1
privkey = Kxpf5b8p3qX56DKEe5NqWbNUP9MnqoRFzZwHRtsFqhzuvUJsYZCy
pubkey  = 03e775fd51f0dfb8cd865d9ff1cca2a158cf651fe997fdc9fee9c1d3b5e995ea77
address = bc1qnjg0jd8228aq7egyzacy8cys3knf9xvrerkf9g

// Account 0, first change address = m/84'/0'/0'/1/0
privkey = KxuoxufJL5csa1Wieb2kp29VNdn92Us8CoaUG3aGtPtcF3AzeXvF
pubkey  = 03025324888e429ab8e3dbaf1f7802648b9cd01e9b418485c5fa4c1b9b5700e1a6
```

```
address = bc1q8c6fshw2dlwun7ekn9qwf37cu2rn755upcp6el
```

Reference

- BIP32 - Hierarchical Deterministic Wallets
- BIP43 - Purpose Field for Deterministic Wallets
- BIP44 - Multi-Account Hierarchy for Deterministic Wallets
- BIP49 - Derivation scheme for P2WPKH-nested-in-P2SH based accounts
- BIP141 - Segregated Witness (Consensus layer)
- BIP173 - Base32 address format for native v0-16 witness outputs