Wallet import format

> *This page contains sample addresses and/or private keys. Do not send bitcoins to or import any sample keys; you will lose your money.*

Wallet Import Format (WIF, also known as Wallet Export Format) is a way of encoding a private ECDSA key so as to make it easier to copy.

A testing suite is available for encoding and decoding of WIF at:

http://gobittest.appspot.com/PrivateKey

Private key to WIF

1 - Take a private key

```
0C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE89827E19D72AA1D
```

2 - Add a 0x80 byte in front of it for mainnet addresses or 0xef for testnet addresses. Also add a 0x01 byte at the end if the private key will correspond to a compressed public key

```
800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE89827E19D72AA1D
```

3 - Perform SHA-256 hash on the extended key

```
8147786C4D15106333BF278D71DADAF1079EF2D2440A4DDE37D747DED5403592
```

4 - Perform SHA-256 hash on result of SHA-256 hash

```
507A5B8DFED0FC6FE8801743720CEDEC06AA5C6FCA72B07C49964492FB98A714
```

5 - Take the first 4 bytes of the second SHA-256 hash, this is the checksum

```
507A5B8D
```

6 - Add the 4 checksum bytes from point 5 at the end of the extended key from point 2

```
800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE89827E19D72AA1D507A5B8D
```

7 - Convert the result from a byte string into a base58 string using Base58Check encoding. This is the Wallet Import Format

```
5HueCGU8rMjxEXxiPuD5BDku4MkFqeZyd4dZ1jvhTVqvbTLvyTJ
```

WIF to private key

1 - Take a Wallet Import Format string

```
5HueCGU8rMjxEXxiPuD5BDku4MkFqeZyd4dZ1jvhTVqvbTLvyTJ
```

2 - Convert it to a byte string using Base58Check encoding

```
800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE89827E19D72AA1D507A5B8D
```

3 - Drop the last 4 checksum bytes from the byte string

```
800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE89827E19D72AA1D
```

4 - Drop the first byte (it should be 0x80). If the private key corresponded to a compressed public key, also drop the last byte (it should be 0x01). If it corresponded to a compressed public key, the WIF string will have started with K or L instead of 5 (or c instead of 9 on testnet). This is the private key.

```
0C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE89827E19D72AA1D
```

WIF checksum checking

1 - Take the Wallet Import Format string

```
5HueCGU8rMjxEXxiPuD5BDku4MkFqeZyd4dZ1jvhTVqvbTLvyTJ
```

2 - Convert it to a byte string using Base58Check encoding

```
800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE89827E19D72AA1D507A5B8D
```

3 - Drop the last 4 checksum bytes from the byte string

```
800C28FCA386C7A227600B2FE50B7CAE11EC86D3BF1FBE471BE89827E19D72AA1D
```

3 - Perform SHA-256 hash on the shortened string

```
8147786C4D15106333BF278D71DADAF1079EF2D2440A4DDE37D747DED5403592
```

4 - Perform SHA-256 hash on result of SHA-256 hash

```
507A5B8DFED0FC6FE8801743720CEDEC06AA5C6FCA72B07C49964492FB98A714
```

5 - Take the first 4 bytes of the second SHA-256 hash, this is the checksum

```
507A5B8D
```

6 - Make sure it is the same, as the last 4 bytes from point 2

```
507A5B8D
```

7 - If they are, and the byte string from point 2 starts with 0x80 (0xef for testnet addresses), then there is no error.

This page is a stub. Help by expanding it.

Bitcoin Core documentation

User documentation	Alert system • Bitcoin Core compatible devices • Data directory • Fallback Nodes • How to import private keys in Bitcoin Core 0.7+ • Installing Bitcoin Core • Running Bitcoin • Transaction fees • Vocabulary
Developer documentation	Accounts explained • API calls list • API reference (JSON-RPC) • Block chain download • Dump format • getblocktemplate • List of address prefixes • Protocol documentation • Script • Technical background of version 1 Bitcoin addresses • Testnet • Transaction Malleability • Wallet import format
History & theory	Common Vulnerabilities and Exposures • DOS/STONED incident • Economic majority • Full node • Original Bitcoin client • Value overflow incident

Retrieved from "https://en.bitcoin.it/w/index.php?title=Wallet_import_format&oldid=63821"
