```
BIP: 329
Layer: Applications
Title: Wallet Labels Export Format
Author: Craig Raw <craig@sparrowwallet.com>
Comments-Summary: No comments yet.
Comments-URI: https://github.com/bitcoin/bips/wiki/Comments:BIP-0329
Status: Draft
Type: Informational
Created: 2022-08-23
License: BSD-2-Clause
```

Abstract

This document specifies a format for the export of labels that may be attached to various common types of records in a wallet.

Copyright

Motivation

The export and import of funds across different Bitcoin wallet applications is well defined through standards such as BIP39, BIP32, BIP44 etc. These standards are well supported and allow users to move easily between different wallets. There is, however, no defined standard to transfer any labels the user may have applied to the transactions, addresses, public keys, inputs, outputs or xpubs in their wallet. The UTXO model that Bitcoin uses makes these labels particularly valuable as they may indicate the source of funds, whether received externally or as a result of change from a prior transaction. In both cases, care must be taken when spending to avoid undesirable leaks of private information.

Labels provide valuable guidance in this regard, and have even become mandatory when spending in several Bitcoin wallets. Allowing users to import and export their labels in a standardized way ensures that they do not experience lock-in to a particular wallet application. In addition, many wallets allow unspent outputs to be frozen or made unspendable within the wallet. Since this wallet-related metadata is similar to labels and not captured elsewhere, it is also included in this format.

Rationale

While there is currently no widely accepted format for exporting and importing labels, there are existing formats in use. SLIP-0015[1] defines a format for exporting address and output labels, but requires encryption using a private

[1] SLIP-0015

key associated with the wallet seed, and thus cannot be used independently by coordinator wallets which cannot access private keys. The Electrum wallet imports and exports address and transaction labels in a JSON format which could be used with other record types, but the format used is not self describing making record type identification difficult.

Specification

In order to be lightweight, human readable and well structured, this BIP uses a JSON format. Further, the JSON Lines format is used (also called newline-delimited JSON)[2]. This allows a document to be split, streamed, or incrementally added to, and limits the potential for formatting errors to invalidate an entire import. It is also a convenient format for command-line processing, which is often line-oriented.

Further to the JSON Lines specification, an export of labels from a wallet must be a UTF-8 encoded text file, containing one record per line consisting of a valid JSON object. Lines are separated by `\n`. Multiline values are not permitted. Each JSON object must contain 3 or 4 key/value pairs, defined as follows:

Key	Description
`type`	One of `tx`, `addr`, `pubkey`, `input`, `output` or `xpub`
`ref`	Reference to the transaction, address, public key, input, output or extended public key
`label`	The label applied to the reference
`origin`	Optional key origin information referencing the wallet associated with the label
`spendable`	One of `true` or `false`, denoting if an output should be spendable by the wallet

The reference is defined for each `type` as follows:

Type	Description	Example
`tx`	Transaction id in hexadecimal format	`f91d0a8a78462bc59398f2c5d7a84fcff491`
`addr`	Address in base58 or bech32 format	`bc1q34aq5drpuwy3wgl9lhup9892qp6svr8l`
`pubkey`	32, 33 or 65 byte public key in hexadecimal format	`0283409659355b6d1cc3c32decd5d561abaa`
`input`	Transaction id and input index separated by a colon	`f91d0a8a78462bc59398f2c5d7a84fcff491`
`output`	Transaction id and output index separated by a colon	`f91d0a8a78462bc59398f2c5d7a84fcff491`
`xpub`	Extended public key as defined by BIP32	`xpub661MyMwAqRbcFtXgS5sYJABqqG9YLmC4`

Each JSON object must contain both `type` and `ref` properties. The `label`, `origin` and `spendable` properties are optional. If the `label` or `spendable` properties are omitted, the importing wallet should not alter these values. The `origin` property should only appear where type is `tx`, and the `spendable` property only where type is `output`.

[2] jsonlines.org

If present, the optional `origin` property must contain an abbreviated output descriptor (as defined by BIP380[3]) describing a BIP32 compatible originating wallet, including all key origin information but excluding any actual keys, any child path elements, or a checksum. This property should be used to disambiguate transaction labels from different wallets contained in the same export, particularly when exporting multiple accounts derived from the same seed.

Care should be taken when exporting due to the privacy sensitive nature of the data. Encryption in transit over untrusted networks is highly recommended, and encryption at rest should also be considered. Unencrypted exports should be deleted as soon as possible. For security reasons no private key types are defined.

Importing

- An importing wallet may ignore records it does not store, and truncate labels if necessary. A suggested default for maximum label length is 255 characters, and an importing wallet should consider warning the user if truncation is applied.
- Wallets importing public key records may derive addresses from them to match against known wallet addresses.
- Wallets importing extended public keys may match them against signers, for example in a multisig setup.

Backwards Compatibility

The nature of this format makes it naturally extensible to handle other record types. However, importing wallets complying to this specification may ignore types not defined here.

Test Vectors

The following fragment represents a wallet label export:

```
{ "type": "tx", "ref": "f91d0a8a78462bc59398f2c5d7a84fcff491c26ba54c4833478b202796c8aafd", 
{ "type": "addr", "ref": "bc1q34aq5drpuwy3wgl9lhup9892qp6svr8ldzyy7c", "label": "Address" }
{ "type": "pubkey", "ref": "0283409659355b6d1cc3c32decd5d561abaac86c37a353b52895a5e6c196d6f
{ "type": "input", "ref": "f91d0a8a78462bc59398f2c5d7a84fcff491c26ba54c4833478b202796c8aafd
{ "type": "output", "ref": "f91d0a8a78462bc59398f2c5d7a84fcff491c26ba54c4833478b202796c8aaf
{ "type": "xpub", "ref": "xpub661MyMwAqRbcFtXgS5sYJABqqG9YLmC4Q1Rdap9gSE8NqtwybGhePY2gZ29ES
{ "type": "tx", "ref": "f546156d9044844e02b181026a1a407abfca62e7ea1159f87bbeaa77b4286c74", 
```

Reference Implementation

TBD

[3]BIP-0380

References