

Recovering Coins Across Multiple Address Types

October 22, 2026 · Document #405 · bip39-pharse.com



Photo by [Susan Holt Simpson](#) on [Unsplash](#)

Learn how one BIP-39 seed derives Legacy, Nested SegWit, Native SegWit, and Taproot addresses, and how to recover balances across different paths.

Contents

The Four Standard Derivation Schemes	2
Why Software Scans Show Incomplete Balances	2
Methods for Complete Balance Recovery	3
Consolidating Scattered Funds	3
FAQ	3
Sources	4
Sample sequences of BIP39 words	4
Reading a BIP39 phrase	5
See also	6

A BIP-39 recovery phrase does not contain address records, transaction histories, or coin balances. It generates a single 512-bit binary seed, from which a master extended private key is computed using HMAC-SHA512. From that master key, wallet software calculates a vast hierarchy of keys defined by **derivation paths**.

Over the lifespan of Bitcoin, address encoding standards have evolved. Users who have held funds for several years and switched wallet software often generated addresses under distinct derivation schemes without realizing it. If a user later imports their seed into an application that inspects only one derivation scheme by default, the wallet may report a zero or partial balance. The missing coins remain on the blockchain, tied to

different branches of the exact same master key.

The Four Standard Derivation Schemes

Bitcoin addresses fall into distinct categories, each defined by an associated Bitcoin Improvement Proposal (BIP) that specifies the derivation path structure `m / purpose' / coin_type' / account' / change / address_index`.

Address Type	Standard	Derivation Path	Address Prefix	Output Script
Legacy	BIP-44	<code>m/44' / 0' / 0'</code>	1	P2PKH
Nested SegWit	BIP-49	<code>m/49' / 0' / 0'</code>	3	P2SH-P2WPKH
Native SegWit	BIP-84	<code>m/84' / 0' / 0'</code>	bc1q	P2WPKH
Taproot	BIP-86	<code>m/86' / 0' / 0'</code>	bc1p	P2TR

For example, using the standard test vector phrase `abandon abandon abandon abandon abandon abandon abandon about`, the first receive address (`change = 0`, `index = 0`) produces completely different public outputs depending on the path:

- BIP-44 path `m/44' / 0' / 0' / 0 / 0` derives a legacy P2PKH address beginning with 1.
- BIP-49 path `m/49' / 0' / 0' / 0 / 0` derives a nested SegWit P2SH address beginning with 3.
- BIP-84 path `m/84' / 0' / 0' / 0 / 0` derives a native SegWit Bech32 address beginning with bc1q.
- BIP-86 path `m/86' / 0' / 0' / 0 / 0` derives a Taproot Bech32m address beginning with bc1p.

Because each script type requires different cryptographic locking scripts on the blockchain, wallet software scanning for native SegWit outputs at `m/84' / . . .` does not inspect outputs stored at legacy addresses on `m/44' / . . .` unless explicitly instructed to do so.

Why Software Scans Show Incomplete Balances

When a seed phrase is entered into recovery software, the wallet must query an indexing server or a local node to check for unspent transaction outputs (UTXOs). Wallets make specific assumptions during this discovery process:

- **Default Purpose Paths:** Modern software often defaults strictly to BIP-84 (`m/84' / 0' / 0'`) or BIP-86 (`m/86' / 0' / 0'`). If historical transactions were received on BIP-44 addresses, the software typically does not query those public keys, leaving the user with an incomplete or zero displayed balance.
- **Account Index Divergence:** Multi-account derivation specifies an `account'` level (for instance, `m/84' / 0' / 1'`). A wallet scanning only `account 0'` misses funds assigned to `account 1'`.
- **Gap Limits:** Standards mandate a lookahead window, typically set to 20 addresses. The wallet derives consecutive addresses on a path. If 20 unused addresses appear in sequence, scanning halts. If a user previously skipped 20 unused addresses or received funds on an address at index 25, automatic discovery may fail to register the balance even if the derivation path is correct.

Methods for Complete Balance Recovery

To recover all funds derived from a single seed phrase, choose between multi-path automated scanning tools or manual account instantiation.

Method 1: Using Multi-Account Scanning Wallets

Advanced desktop wallet coordinators (such as Sparrow Wallet or Electrum) allow granular control over derivation branches.

- 1 Install an open-source coordinator that allows custom derivations and supports offline air-gapped operation.
- 2 Create a new wallet file and select the hardware device or input the BIP-39 mnemonic.
- 3 Create an initial account selecting Native SegWit ($m/84'/0'/0'$). Name it accordingly.
- 4 Add another account within the same wallet or create a second wallet file using the same seed, selecting Legacy ($m/44'/0'/0'$).
- 5 Repeat the process for Nested SegWit ($m/49'/0'/0'$) and Taproot ($m/86'/0'/0'$).
- 6 Verify the transaction history of each independent account. If transactions stopped before an expected balance appeared, increase the gap limit in the wallet preferences from 20 to 50 or 100.

Method 2: Comprehensive Hardware Companion Suites

Certain hardware wallet management suites (such as Trezor Suite) scan multiple standard script types during initialization or account addition. When adding a Bitcoin account, the interface checks for balances across BIP-44, BIP-49, BIP-84, and BIP-86 paths, presenting detected balances under separate account labels. If a historical account is not detected automatically, use the account creation settings to manually check legacy or SegWit branches.

Consolidating Scattered Funds

Managing multiple derivation paths on the same recovery phrase introduces friction during long-term storage and inheritance planning. Once all balances are located across the four types, consider consolidating the UTXOs into a single modern script type, such as native SegWit or Taproot.

To consolidate safely:

- 1 Ensure all accounts have synchronized and all active UTXOs are accounted for.
- 2 Generate a fresh receive address on your preferred modern path (typically Native SegWit $m/84'/0'/0'/0'/x$ or Taproot $m/86'/0'/0'/0'/x$).
- 3 Construct separate sweep transactions from each older account (BIP-44 and BIP-49) to that new address.
- 4 Check network fee conditions before broadcasting to avoid excessive fee burning on legacy inputs.

FAQ

Can one BIP-39 phrase hold funds on different address types simultaneously?

Yes. The recovery phrase computes a master key that can derive every valid Bitcoin path. Your master private key controls the legacy, nested SegWit, native SegWit, and Taproot addresses simultaneously.

Does restoring my seed into a new wallet automatically move my coins?

No. Restoring a seed simply recalculates your private keys so the software can sign transactions. Your coins remain static on the blockchain under their respective output scripts until you explicitly broadcast a transaction.

Why does my restored wallet show an empty transaction history?

Your wallet software is likely checking a different derivation path than the one used when you received your funds, or your active addresses lie beyond the software's default address gap limit.

Sources

- [BIP-0044: Multi-Account Hierarchy for Deterministic Wallets \(github.com\)](#)
- [BIP-0049: Derivation scheme for P2WPKH-nested-in-P2SH based accounts \(github.com\)](#)
- [BIP-0084: Derivation scheme for P2WPKH based accounts \(github.com\)](#)

Continue reading on bip39-phrased.com: [BIP 44 – Multi-Account Hierarchy for Deterministic Wallets](#)

More on this topic: [Recovery](#)

Sample sequences of BIP39 words

Randomly generated sequences of words from the BIP39 English wordlist. None of them passes the BIP39 checksum, so no wallet will accept them: they show the format only. Never store funds with a phrase you found in a document.

Example 1					
1	polar	2	vehicle	3	math
4	forum	5	ribbon	6	divorce
7	purchase	8	vacuum	9	title
10	jungle	11	same	12	viable

Example 2					
1	tribe	2	walnut	3	beauty
4	bicycle	5	toe	6	kitchen
7	filter	8	great	9	dove
10	juice	11	junk	12	throw

Example 3					
1	pigeon	2	bicycle	3	stomach
4	perfect	5	nasty	6	fame
7	clarify	8	now	9	material
10	donkey	11	cabbage	12	melody

Example 4					
1	lens	2	spread	3	season
4	nurse	5	dumb	6	plastic
7	extend	8	mail	9	come
10	coach	11	airport	12	alcohol

Example 5					
1	travel	2	pass	3	near
4	supreme	5	clog	6	measure
7	plunge	8	phone	9	abstract
10	trend	11	cycle	12	heavy

Example 6

1	ranch	2	negative	3	tattoo	4	appear	5	trend	6	frequent
7	brass	8	fold	9	bundle	10	fresh	11	derive	12	point

Example 7

1	start	2	patrol	3	glimpse	4	vicious	5	antenna	6	elbow
7	drum	8	club	9	manual	10	bid	11	expose	12	hawk

Example 8

1	any	2	fluid	3	home	4	hope	5	ripple	6	upon
7	glad	8	zebra	9	explain	10	penalty	11	anchor	12	media

Example 9

1	slush	2	medal	3	bleak	4	village	5	laugh	6	traffic
7	zoo	8	uncover	9	odor	10	alpha	11	machine	12	label

Example 10

1	beach	2	faint	3	crack	4	render	5	piece	6	name
7	fuel	8	interest	9	antenna	10	strong	11	erosion	12	demand

Reading a BIP39 phrase

Every word stands for an 11-bit number: its position in the 2048-word list, counted from 0. Here is Example 1 decoded word by word. Each word links to its page on bip39-phrase.com, where the list is numbered from 1.

#	Word	Index (0-2047)	Word no.	11 bits
1	polar	1339	1340	10100111011
2	vehicle	1935	1936	11110001111
3	math	1096	1097	10001001000
4	forum	733	734	01011011101
5	ribbon	1480	1481	10111001000
6	divorce	512	513	01000000000
7	purchase	1393	1394	10101110001
8	vacuum	1924	1925	11110000100
9	title	1815	1816	11100010111
10	jungle	968	969	01111001000
11	same	1527	1528	10111110111
12	viable	1946	1947	11110011010

The 12 words carry 132 bits: 128 bits of entropy followed by a 4-bit checksum. The checksum must equal the first 4 bits of the SHA-256 hash of the entropy.

Entropy (hex): `a77e3e242ddb90802b8f84e2ef22fbf9`

This sequence ends in the checksum bits 1010, but SHA-256 of its entropy starts with 0100, so wallets reject it as an invalid phrase. That is the point of the checksum: about 15 of every 16 random or mistyped sequences fail it.

See also

- [Coins Sent to Mistyped or Wrongly Restored Wallets](#) · October 15, 2026
- [What a Change Address Is and Why Wallets Create One](#) · October 15, 2026
- [Air-Gapped Signing with PSBT and QR Codes](#) · June 2, 2026
- [BIP-39 SEED PHRASE The Principles of Cryptographic Backup and Wallet Recovery](#) · June 10, 2025

Educational document. The example sequences are random and fail the BIP39 checksum: they are not wallets and must never hold funds.
Security notes: bip39-pharse.com/security/