Bitcoin BIP110 Technical Specifications

Version: 1.3
Updated: 7 September 2026

1. Scope

This document defines the chain identity, node requirements, amount format, block-header transition, replay-protection requirements, custody invariants, and production acceptance checks needed for exchange integration of the chain identified by the Bitcoin BIP110 project.

This specification is intentionally operational. Consensus behavior remains defined by the Bitcoin Knots reference implementation and its source code.

2. Asset identity

Field	Value
Exchange listing name	Bitcoin BIP110
Exchange listing ticker	BIP110
Date of issue (chain split)	8 August 2026
Max supply (nominal)	21,000,000 BIP110
Circulating supply at issuance (nominal)	20,067,600 BIP110
Asset class	Native UTXO coin
Origin ledger	Bitcoin
Decimal places	8
Base units per coin	100,000,000
Genesis block	Bitcoin genesis block
Genesis hash	`000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f`

The ticker is a project declaration. Exchanges may use a different internal or public symbol if required to avoid a collision.

For integration metadata, the date of issue is the date of the first BIP110 branch block at height 961632. The nominal circulating supply at issuance is calculated from the protocol subsidy schedule through the last common height 961631. It does not subtract coins that were lost, burned, or otherwise provably unspendable.

3. Fork identity

Field	Value
Last common height	961631
Last common hash	`00000000000000000000807f9dc917442a67910426d79ebb2f8aa2149327ce8a`
First BIP110 branch height	961632
BIP110 hash at 961632	`00000000000000000000169eb6f811ddbd0daf343af7b62180cdb13e7c78dbc16`
Last SHA256d height on BIP110 branch	961639
BIP110 hash at 961639	`00000000000000000001bbc439e13f749dca850d32c7a2834165338713027e65`
First BLAKE2b height	961640
First BLAKE2b hash	`0000000000000050c1e5f69672f459293be14f46e5a494e7a8c8541396f18eeb`
First BLAKE2b date	30 August 2026

Mandatory identity rule

An exchange service MUST treat the chain as Bitcoin BIP110 only when all configured fork anchors match the expected hashes.

Recommended startup checks:

```
bitcoin-cli getblockhash 961632
bitcoin-cli getblockhash 961639
bitcoin-cli getblockhash 961640
```

A mismatch MUST disable deposit crediting and withdrawal broadcast.

4. Reference software

Field	Value
Reference implementation	Bitcoin Knots
Minimum reference release for this specification	29.4.1.knots20260508
Release tag	`v29.4.1.knots20260508`
Official release directory	`https://bitcoinknots.org/files/29.x/29.4.1.knots20260508/`
Release checksum file	`SHA256SUMS`
Release signature file	`SHA256SUMS.asc`

Production deployments SHOULD verify release hashes and signatures before installation.

5. Explorer and integration endpoints

Field	Value
Project website	`https://bitcoinbip110.org/`
Source repository	`https://github.com/bitcoinknots/bitcoin`
Block explorer	`https://mempool.guide/`
Transaction URL template	`https://mempool.guide/tx/{txid}`
Address URL template	`https://mempool.guide/address/{address}`
Circulating supply URL	No dedicated endpoint currently published

6. Branding assets

Field	Value
Preferred logo	`https://bitcoinbip110.org/downloads/logo-orange-on-transparent-bg.svg`
Secondary logo	`https://bitcoinbip110.org/downloads/logo-black-on-transparent-bg.svg`
Dark-background logo	`https://bitcoinbip110.org/downloads/logo-black-on-transparent-bg-with-contour.svg`
Preferred usage	Orange logo is the first choice
Secondary usage	Black logo is the second choice when the orange logo cannot be used
Dark-background usage	Use the black logo with white contour when the black logo is presented on a dark background

The Resource Directory uses the contoured black SVG as the dark-background example.

8. Mainnet network parameters

Field	Value
RPC chain name	`main`
Default P2P port	8333
Default RPC port	8332
P2P message start	`f9 be b4 d9`
P2PKH prefix	`0x00`
P2SH prefix	`0x05`
Private-key prefix	`0x80`
Bech32 HRP	`bc`

BIP32 public prefix	`0488b21e`
BIP32 private prefix	`0488ade4`

Chain-identity warning

The values above overlap Bitcoin BTC mainnet. An exchange MUST NOT identify the chain solely by:

- `chain=main`;
- genesis block;
- P2P magic;
- default ports;
- Base58 prefix;
- Bech32 HRP;
- address appearance.

The fork-anchor hashes are the required identity mechanism for this specification.

8. Proof of work and block headers

Field	Value
Proof of work through 961639	SHA256d
Proof of work from 961640	BLAKE2b
BLAKE2b activation height	961640
BLAKE2b permanence	Permanent
Legacy header size	80 bytes
Header size from 961640	164 bytes
Header generation	Version 2
Target spacing	600 seconds
Legacy target timespan parameter	1,209,600 seconds
Mainnet BLAKE2b target shift	22

Raw block-header consumers MUST support the version 2 format from height 961640.

The reference node exposes additional fields in `getblockheader` and `getblock`, including:

- `txcount`
- `header_version`
- `nonce2`
- `nonce3`
- `extranonce`

- `time_offset`
- `header_flags`
- `xor_key`
- `xor_key_mask_clear_bits`
- `mm_rhs`

An exchange that delegates validation to the reference node can consume normalized RPC output. An exchange that independently hashes headers MUST implement the reference BLAKE2b header rules.

9. Temporary reduced-data consensus window

Field	Value
Activation height	961640
Expiry clock	Previous block median time past
Expiry time	1 September 2027 00:00 UTC
Temporary maximum block weight	800,000 weight units
Approximate serialized size	About 300 kB
BLAKE2b after expiry	Remains active

Inputs spending coins created before the fork are exempt from the new script and data restrictions during the temporary window, as described by the developer guidance.

Exchange systems SHOULD use the reference node for consensus validation rather than duplicating these temporary rules in an application-layer indexer.

10. Monetary parameters

Field	Value
Base units per coin	100,000,000
Subsidy halving interval	210,000 blocks
Coinbase maturity	100 blocks
Current subsidy era at the fork	3.125 coins per block before fees
Next halving height	1,050,000
Nominal issued supply at split	20,067,600 coins

Accounting systems MUST use integer base units internally.

11. Replay model

9.1 Ordinary signatures

A transaction spending a pre-fork output with an ordinary signature may remain valid on both chains. The same signed transaction can therefore be rebroadcast where the transaction and its inputs are valid.

9.2 SIGHASH_UNIFIED

Field	Value
Name	`SIGHASH_UNIFIED`
Hash-type bit	`0x20`
Tagged-hash name	`UnifiedSighash`
Opt-in granularity	Per signature
Replay direction	Protected from this chain to a chain that does not implement the algorithm
Reverse protection	None for ordinary signatures
Covered script families	Bare/P2SH, SegWit v0, Taproot key path, Tapscript
Upstream test vectors	166

The unified-sighash design document currently labels itself draft, while the released reference implementation activates the rule with the BLAKE2b deployment.

9.3 Signer data requirement

A signer implementing `SIGHASH_UNIFIED` needs the value and scriptPubKey of every spent output for the transaction inputs. PSBT can carry the required previous-output data.

9.4 Exchange custody rule

Normal BIP110 withdrawal coin selection MUST exclude UTXOs whose cross-chain separation state is unknown.

Recommended internal state:

```
SHARED
BIP110_ONLY
BTC_ONLY
UNKNOWN
```

Normal customer withdrawals SHOULD use `BIP110_ONLY` outputs.

12. Coin-separation procedure

For a shared pre-fork UTXO:

1. Create a BIP110-only destination from a dedicated BIP110 wallet.

2. Construct a send-to-self transaction on BIP110.
3. Sign applicable inputs with `SIGHASH_UNIFIED`.
4. Broadcast through the verified BIP110 node.
5. Wait for the exchange's confirmation threshold.
6. Verify confirmation on BIP110.
7. Verify that the BTC-side source output remains unspent.
8. Mark the new output `BIP110_ONLY`.

An exchange SHOULD complete separation before enabling normal withdrawals from recovered fork inventory.

13. Wallet architecture requirements

BTC and BIP110 systems SHOULD use:

- separate data directories;
- separate node processes;
- separate wallet databases;
- separate seeds, xpubs, or HSM namespaces;
- separate deposit-address mappings;
- separate withdrawal queues;
- separate broadcast paths;
- separate asset IDs in accounting and risk systems.

Address encodings overlap. An address string MUST NOT be used as the network discriminator.

14. Deposit specification

A BIP110 deposit processor MUST:

1. verify the fork-anchor hashes;
2. verify that the node is synchronized;
3. observe the transaction through the BIP110 node or a BIP110-validated indexer;
4. map the destination using the BIP110 address database;
5. apply the exchange's confirmation threshold;
6. enforce idempotent crediting;
7. stop crediting when chain identity fails.

Deposit UI MUST explicitly identify the network as Bitcoin BIP110 / BIP110.

15. Withdrawal specification

A BIP110 withdrawal processor MUST:

1. verify the asset ID is BIP110;
2. verify fork anchors before transaction construction and broadcast;

3. select only approved BIP110 inventory;
4. reject `SHARED` or `UNKNOWN` UTXOs from standard withdrawals;
5. use a BIP110-specific signer policy;
6. decode and validate the transaction through the BIP110 node before broadcast;
7. broadcast through BIP110 infrastructure only;
8. monitor confirmation on the BIP110 chain.

16. Customer fork-credit accounting

The protocol creates corresponding on-chain outputs for keys that controlled pre-fork UTXOs. It does not create exchange customer database balances.

The exchange MUST define its own fork-credit policy.

The technical reference state is the shared UTXO set after height 961631. Internal customer accounting around the split requires exchange-specific treatment of pending deposits, pending withdrawals, internal transfers, lending, margin, and external custody.

Before enabling credited balances for withdrawal:

```
verified BIP110 custody assets >= credited BIP110 customer liabilities + operational reserves
```

17. Node health requirements

Recommended precondition for deposit and withdrawal service:

```
checkpoint_961632 == expected
checkpoint_961639 == expected
checkpoint_961640 == expected
initialblockdownload == false
wallet_or_indexer_height == node_height within configured tolerance
rpc_health == healthy
signer_health == healthy
```

Any failed mandatory condition SHOULD place the BIP110 asset into a fail-closed state.

18. RPC verification profile

Minimum recommended startup profile:

```
bitcoin-cli getblockchaininfo
bitcoin-cli getdeploymentinfo
bitcoin-cli getblockhash 961632
bitcoin-cli getblockhash 961639
bitcoin-cli getblockhash 961640
```

Expected fork hashes are defined in Section 3.

During the reduced-data window, `getdeploymentinfo` should report BLAKE2b active at height 961640 and the reduced-data deployment active at the same height.

19. Testing profile

An exchange integration SHOULD pass these tests before production:

Chain identity

- correct BIP110 node accepted;
- BTC node rejected by fork-anchor check;
- unsynchronized node rejected.

Amounts

- exact 8-decimal parsing;
- integer base-unit accounting;
- maximum and dust boundary behavior defined by exchange policy.

Signing

- ordinary signing path does not get used accidentally for unsplit inventory;
- `SIGHASH_UNIFIED` test vectors pass in custom signers;
- multi-input PSBT contains every required previous output;
- supported script families pass end-to-end signing.

Headers and indexers

- height 961639 parsed as legacy header;
- height 961640 parsed as version 2 BLAKE2b header;
- custom block indexer follows the same chain as the reference node.

Accounting

- customer snapshot reproducible;
- replayed transactions handled;
- duplicate credit jobs idempotent;
- assets reconcile against liabilities.

Failure handling

- checkpoint mismatch pauses service;
- node outage pauses service;
- signer outage pauses withdrawal creation;
- reorg handling does not double credit.

20. Regtest support for engineering tests

The reference software exposes regtest-only arguments including:

```
-testactivationheight=blake2b@<height>
-rdtsexpiry=<unix_time>
-blake2b_headline=<headline>
```

These can be used to exercise the transition in controlled integration tests without depending on mainnet timing.

References

Bitcoin BIP110 Resource Directory: https://bitcoinbip110.org/directory/#overview
Bitcoin BIP110 Developer Reference: https://bitcoinbip110.org/developers/
Bitcoin BIP110 Exchange Integration: https://bitcoinbip110.org/exchanges/
Bitcoin BIP110 Technicals: https://bitcoinbip110.org/technicals/
Bitcoin Knots source: https://github.com/bitcoinknots/bitcoin