

Bitcoin Core file system

Contents

- [Data directory location](#)
- [Data directory layout](#)
- [Multi-wallet environment](#)
 - [Berkeley DB database based wallets](#)
 - [SQLite database based wallets](#)
- [GUI settings](#)
- [Legacy subdirectories and files](#)
- [Notes](#)

Data directory location

The data directory is the default location where the Bitcoin Core files are stored.

1. The default data directory paths for supported platforms are:

Platform	Data directory path
Linux	<code>\$HOME/.bitcoin/</code>
macOS	<code>\$HOME/Library/Application Support/Bitcoin/</code>
Windows	<code>%APPDATA%\Bitcoin\</code> [1]

2. A custom data directory path can be specified with the `-datadir` option.
3. All content of the data directory, except for `bitcoin.conf` file, is chain-specific. This means the actual data directory paths for non-mainnet cases differ:

Chain option	Data directory path
<code>-chain=main</code> (default)	<code>path_to_datadir/</code>
<code>-chain=test</code> OR <code>-testnet</code>	<code>path_to_datadir/testnet3/</code>
<code>-chain=signet</code> OR <code>-signet</code>	<code>path_to_datadir/signet/</code>
<code>-chain=regtest</code> OR <code>-regtest</code>	<code>path_to_datadir/regtest/</code>

Data directory layout

Subdirectory	File(s)	Description
<code>blocks/</code>		Blocks directory; can be specified by <code>-blocksdir</code> option (except for <code>blocks/index/</code>)

blocks/index/	LevelDB database	Block index; <code>-blocksdir</code> option does not affect this path
blocks/	blkNNNNN.dat [2]	Actual Bitcoin blocks (in network format, dumped in raw on disk, 128 MiB per file)
blocks/	revNNNNN.dat [2]	Block undo data (custom format)
chainstate/	LevelDB database	Blockchain state (a compact representation of all currently unspent transaction outputs (UTXOs) and metadata about the transactions they are from)
indexes/txindex/	LevelDB database	Transaction index; <i>optional</i> , used if <code>-txindex=1</code>
indexes/blockfilter/basic/db/	LevelDB database	Blockfilter index LevelDB database for the basic filtertype; <i>optional</i> , used if <code>-blockfilterindex=basic</code>
indexes/blockfilter/basic/	fltrNNNNN.dat [2]	Blockfilter index filters for the basic filtertype; <i>optional</i> , used if <code>-blockfilterindex=basic</code>
indexes/coinstats/db/	LevelDB database	Coinstats index; <i>optional</i> , used if <code>-coinstatsindex=1</code>
wallets/		Contains wallets ; can be specified by <code>-walletdir</code> option; if <code>wallets/</code> subdirectory does not exist, wallets reside in the data directory
./	anchors.dat	Anchor IP address database, created on shutdown and deleted at startup. Anchors are last known outgoing block-relay-only peers that are tried to re-connect to on startup
./	banlist.json	Stores the addresses/subnets of banned nodes.
./	bitcoin.conf	User-defined configuration settings for <code>bitcoind</code> or <code>bitcoin-qt</code> . File is not written to by the software and must be created manually. Path can be specified by <code>-conf</code> option
./	bitcoind.pid	Stores the process ID (PID) of <code>bitcoind</code> or <code>bitcoin-qt</code> while running; created at start and deleted on shutdown; can be specified by <code>-pid</code> option
./	debug.log	Contains debug information and general logging generated by <code>bitcoind</code> or <code>bitcoin-qt</code> ; can be specified by <code>-debuglogfile</code> option
./	fee_estimates.dat	Stores statistics used to estimate minimum transaction fees required for confirmation
./	guisettings.ini.bak	Backup of former GUI settings after <code>-resetguisettings</code> option is used

./	ip_asn.map	IP addresses to Autonomous System Numbers (ASNs) mapping used for bucketing of the peers; path can be specified with the <code>-asmap</code> option
./	mempool.dat	Dump of the mempool's transactions
./	onion_v3_private_key	Cached Tor onion service private key for <code>-listenonion</code> option
./	i2p_private_key	Private key that corresponds to our I2P address. When <code>-i2psam=</code> is specified the contents of this file is used to identify ourselves for making outgoing connections to I2P peers and possibly accepting incoming ones. Automatically generated if it does not exist.
./	peers.dat	Peer IP address database (custom format)
./	settings.json	Read-write settings set through GUI or RPC interfaces, augmenting manual settings from bitcoin.conf . File is created automatically if read-write settings storage is not disabled with <code>-nosettings</code> option. Path can be specified with <code>-settings</code> option
./	.cookie	Session RPC authentication cookie; if used, created at start and deleted on shutdown; can be specified by <code>-rpccookiefile</code> option
./	.lock	Data directory lock file

Multi-wallet environment

Wallets are Berkeley DB (BDB) or SQLite databases.

1. Each user-defined wallet named "wallet_name" resides in the `wallets/wallet_name/` subdirectory.
2. The default (unnamed) wallet resides in `wallets/` subdirectory; if the latter does not exist, the wallet resides in the data directory.
3. A wallet database path can be specified with the `-wallet` option.
4. `wallet.dat` files must not be shared across different node instances, as that can result in key-reuse and double-spends due the lack of synchronization between instances.
5. Any copy or backup of the wallet should be done through a `backupwallet` call in order to update and lock the wallet, preventing any file corruption caused by updates during the copy.

Berkeley DB database based wallets

Subdirectory	File(s)	Description
database/	BDB	Part of BDB environment; created at start and deleted on shutdown; a user

	logging files	<i>must keep it as safe as personal wallet</i> <code>wallet.dat</code>
<code>./</code>	<code>db.log</code>	BDB error file
<code>./</code>	<code>wallet.dat</code>	Personal wallet (a BDB database) with keys and transactions
<code>./</code>	<code>.walletlock</code>	BDB wallet lock file

SQLite database based wallets

Subdirectory	File	Description
<code>./</code>	<code>wallet.dat</code>	Personal wallet (a SQLite database) with keys and transactions
<code>./</code>	<code>wallet.dat-journal</code>	SQLite Rollback Journal file for <code>wallet.dat</code> . Usually created at start and deleted on shutdown. A user <i>must keep it as safe as the</i> <code>wallet.dat</code> file.

GUI settings

`bitcoin-qt` uses [QSettings](#) class; this implies platform-specific [locations where application settings are stored](#).

Legacy subdirectories and files

These subdirectories and files are no longer used by Bitcoin Core:

Path	Description	Repository notes
<code>banlist.dat</code>	Stores the addresses/subnets of banned nodes; superseded by <code>banlist.json</code> in 22.0 and completely ignored in 23.0	PR #20966 , PR #22570
<code>blktree/</code>	Blockchain index; replaced by <code>blocks/index/</code> in 0.8.0	PR #2231 , 8fdc94cc
<code>coins/</code>	Unspent transaction output database; replaced by <code>chainstate/</code> in 0.8.0	PR #2231 , 8fdc94cc
<code>blkindex.dat</code>	Blockchain index BDB database; replaced by <code>{chainstate/, blocks/index/, blocks/revNNNNN.dat^[2]}</code> in 0.8.0	PR #1677
<code>blk000?.dat</code>	Block data (custom format, 2 GiB per file); replaced by <code>blocks/blkNNNNN.dat^[2]</code> in 0.8.0	PR #1677
<code>addr.dat</code>	Peer IP address BDB database; replaced by <code>peers.dat</code> in 0.7.0	PR #1198 , 928d3a01
<code>onion_private_key</code>	Cached Tor onion service private key for <code>-listenonion</code> option. Was used for Tor v2 services; replaced by <code>onion_v3_private_key</code> in 0.21.0	PR #19954

Notes

1. The `/` (slash, U+002F) is used as the platform-independent path component separator in this document.
2. `NNNN` matches `[0-9]{5}` regex.