Bitcoin Core file system

Contents

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	`$HOME/.bitcoin/`
macOS	`$HOME/Library/Application Support/Bitcoin/`
Windows	`%APPDATA%\Bitcoin\` [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
`-chain=main` (default)	*path_to_datadir*`/`
`-chain=test` or `-testnet`	*path_to_datadir*`/testnet3/`
`-chain=signet` or `-signet`	*path_to_datadir*`/signet/`
`-chain=regtest` or `-regtest`	*path_to_datadir*`/regtest/`

Data directory layout

Subdirectory	File(s)	Description
`blocks/`		Blocks directory; can be specified by `-blocksdir` option (except for `blocks/index/`)
`blocks/index/`	LevelDB database	Block index; `-blocksdir` 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; *optional*, used if `-txindex=1`
`indexes/blockfilter/basic/db/`	LevelDB database	Blockfilter index LevelDB database for the basic filtertype; *optional*, used if `-blockfilterindex=basic`
`indexes/blockfilter/basic/`	`fltrNNNNN.dat`[2]	Blockfilter index filters for the basic filtertype; *optional*, used if `-blockfilterindex=basic`

Subdirectory	File(s)	Description
`indexes/coinstats/db/`	LevelDB database	Coinstats index; *optional*, used if `-coinstatsindex=1`
`wallets/`		Contains wallets; can be specified by `-walletdir` option; if `wallets/` 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 `bitcoind` or `bitcoin-qt`. File is not written to by the software and must be created manually. Path can be specified by `-conf` option

Subdirectory	File(s)	Description
`./`	`bitcoind.pid`	Stores the process ID (PID) of `bitcoind` or `bitcoin-qt` while running; created at start and deleted on shutdown; can be specified by `-pid` option
`./`	`debug.log`	Contains debug information and general logging generated by `bitcoind` or `bitcoin-qt`; can be specified by `-debuglogfile` option
`./`	`fee_estimates.dat`	Stores statistics used to estimate minimum transaction fees required for confirmation
`./`	`guisettings.ini.bak`	Backup of former GUI settings after `-resetguisettings` 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 `-asmap` option

Subdirectory	File(s)	Description
./	mempool.dat	Dump of the mempool's transactions
./	onion_v3_private_key	Cached Tor onion service private key for -listenonion option
./	i2p_private_key	Private key that corresponds to our I2P address. When -i2psam= 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)

Subdirectory	File(s)	Description
`./`	`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 `-nosettings` option. Path can be specified with `-settings` option
`./`	`.cookie`	Session RPC authentication cookie; if used, created at start and deleted on shutdown; can be specified by `-rpccookiefile` 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 logging files	Part of BDB environment; created at start and deleted on shutdown; a user *must keep it as safe* as personal wallet `wallet.dat`
`./`	`db.log`	BDB error file
`./`	`wallet.dat`	Personal wallet (a BDB database) with keys and transactions
`./`	`.walletlock`	BDB wallet lock file

SQLite database based wallets

Subdirectory	File	Description
`./`	`wallet.dat`	Personal wallet (a SQLite database) with keys and transactions
`./`	`wallet.dat-journal`	SQLite Rollback Journal file for `wallet.dat`. Usually created at start and deleted on shutdown. A user *must keep it as safe* as the `wallet.dat` 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
`banlist.dat`	Stores the addresses/subnets of banned nodes; superseded by `banlist.json` in 22.0 and completely ignored in 23.0	PR #20966, PR #22570
`blktree/`	Blockchain index; replaced by `blocks/index/` in 0.8.0	PR #2231, `8fdc94cc`
`coins/`	Unspent transaction output database; replaced by `chainstate/` in 0.8.0	PR #2231, `8fdc94cc`
`blkindex.dat`	Blockchain index BDB database; replaced by {`chainstate/`, `blocks/index/`, `blocks/revNNNNN.dat`[2]} in 0.8.0	PR #1677
`blk000?.dat`	Block data (custom format, 2 GiB per file); replaced by `blocks/blkNNNNN.dat`[2] in 0.8.0	PR #1677
`addr.dat`	Peer IP address BDB database; replaced by `peers.dat` in 0.7.0	PR #1198, `928d3a01`
`onion_private_key`	Cached Tor onion service private key for `-listenonion` option. Was used for Tor v2 services; replaced by `onion_v3_private_key` 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. `NNNNN` matches `[0-9]{5}` regex.