FreeBSD Build Guide

Updated for FreeBSD 12.3

This guide describes how to build bitcoind, command-line utilities, and GUI on FreeBSD.

Preparation

1. Install Required Dependencies

Run the following as root to install the base dependencies for building.

```
pkg install autoconf automake boost-libs git gmake libevent libtool pkgconf
```

See dependencies.md for a complete overview.

2. Clone Bitcoin Repo

Now that `git` and all the required dependencies are installed, let's clone the Bitcoin Core repository to a directory. All build scripts and commands will run from this directory.

```
git clone https://github.com/bitcoin/bitcoin.git
```

3. Install Optional Dependencies

Wallet Dependencies It is not necessary to build wallet functionality to run either `bitcoind` or `bitcoin-qt`.

Descriptor Wallet Support

`sqlite3` is required to support descriptor wallets. Skip if you don't intend to use descriptor wallets.

```
pkg install sqlite3
```

Legacy Wallet Support

`db5` is only required to support legacy wallets. Skip if you don't intend to use legacy wallets.

```
pkg install db5
```

GUI Dependencies Qt5

Bitcoin Core includes a GUI built with the cross-platform Qt Framework. To compile the GUI, we need to install `qt5`. Skip if you don't intend to use the GUI.

```
pkg install qt5
```

libqrencode

The GUI can encode addresses in a QR Code. To build in QR support for the GUI, install `libqrencode`. Skip if not using the GUI or don't want QR code functionality.

```
pkg install libqrencode
```

Notifications ZeroMQ

Bitcoin Core can provide notifications via ZeroMQ. If the package is installed, support will be compiled in.

```
pkg install libzmq4
```

Test Suite Dependencies There is an included test suite that is useful for testing code changes when developing. To run the test suite (recommended), you will need to have Python 3 installed:

```
pkg install python3
```

Building Bitcoin Core

1. Configuration

There are many ways to configure Bitcoin Core, here are a few common examples:

Descriptor Wallet and GUI: This explicitly enables the GUI and disables legacy wallet support, assuming `sqlite` and `qt` are installed.

```
./autogen.sh
./configure --without-bdb --with-gui=yes MAKE=gmake
```

Descriptor & Legacy Wallet. No GUI: This enables support for both wallet types and disables the GUI, assuming `sqlite3` and `db5` are both installed.

```
./autogen.sh
./configure --with-gui=no --with-incompatible-bdb \
    BDB_LIBS="-ldb_cxx-5" \
    BDB_CFLAGS="-I/usr/local/include/db5" \
    MAKE=gmake
```

No Wallet or GUI

```
./autogen.sh
./configure --without-wallet --with-gui=no MAKE=gmake
```

2. Compile

Important: Use `gmake` (the non-GNU `make` will exit with an error).

```
gmake # use "-j N" for N parallel jobs
gmake check # Run tests if Python 3 is available
```