NetBSD build guide

(updated for NetBSD 8.0)

This guide describes how to build bitcoind and command-line utilities on NetBSD.

This guide does not contain instructions for building the GUI.

Preparation

You will need the following modules, which can be installed via pkgsrc or pkgin:

```
autoconf
automake
boost
git
gmake
libevent
libtool
pkg-config
python37
```

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

See dependencies.md for a complete overview.

Building Bitcoin Core

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

With descriptor wallet: The descriptor wallet uses `sqlite3`. You can install it using:

```
pkgin install sqlite3
```

```
./autogen.sh
./configure --with-gui=no --without-bdb \
    CPPFLAGS="-I/usr/pkg/include" \
    LDFLAGS="-L/usr/pkg/lib" \
    BOOST_CPPFLAGS="-I/usr/pkg/include" \
    MAKE=gmake
```

With legacy wallet: BerkeleyDB is use for legacy wallet functionality.

It is recommended to use Berkeley DB 4.8. You cannot use the BerkeleyDB library from ports. You can use the installation script included in contrib/ like so:

```
./contrib/install_db4.sh `pwd`
```

from the root of the repository. Then set `BDB_PREFIX` for the next section:

```
export BDB_PREFIX="$PWD/db4"
```

```
./autogen.sh
./configure --with-gui=no CPPFLAGS="-I/usr/pkg/include" \
    LDFLAGS="-L/usr/pkg/lib" \
    BOOST_CPPFLAGS="-I/usr/pkg/include" \
    BDB_LIBS="-L${BDB_PREFIX}/lib -ldb_cxx-4.8" \
    BDB_CFLAGS="-I${BDB_PREFIX}/include" \
    MAKE=gmake
```

Without wallet:

```
./autogen.sh
./configure --with-gui=no --disable-wallet \
    CPPFLAGS="-I/usr/pkg/include" \
    LDFLAGS="-L/usr/pkg/lib" \
    BOOST_CPPFLAGS="-I/usr/pkg/include" \
    MAKE=gmake
```

Build and run the tests:

```
gmake # use "-j N" here for N parallel jobs
gmake check
```