

Darwin10Build

Building the BSD port of OpenJDK, Java 1.7 on Max OS X 10.6.4

This content began its life at <http://confluence.concord.org/display/CCTR/Build+OpenJDK+Java+1.7.0+on+Mac+OS+X+10.5>, but should be annotated and extended as we learn more. See the [BSD-Port mailing list](#) for information.

Dependencies

SoyLatte 32-bit Java 1.6 binaries

One can use Landon Fuller's [SoyLatte 32-bit Java 1.6 binaries](#) to build the OpenJDK bsd port. After downloading the binaries, copy the whole directory to `/usr/local/soylatte16-i386-1.0.3`

Test the SoyLatte install:

```
$ /usr/local/soylattel6-i386-1.0.3/bin/java -version
java version "1.6.0_03-p3"
Java(TM) SE Runtime Environment (build 1.6.0_03-p3-landonf_19_aug_2008_14_55-b00)
Java HotSpot(TM) Server VM (build 1.6.0_03-p3-landonf_19_aug_2008_14_55-b00, mixed mode)
```

Mercurial

If you have [MacPorts](#) installed:

```
$ sudo port install mercurial +bash_completion
```

Otherwise you will need to [download](#) the latest version and install it.

Test the mercurial install:

```
$ hg --version
Mercurial Distributed SCM (version 1.1.2)

Copyright (C) 2005-2008 Matt Mackall <mpm@selenic.com> and others
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Forest Extension to Mercurial

Note: The Forest extension does not seem to be actively maintained as noted from the [ForestExtension](#) wiki page. Please use Patrick Mézard's clone of hgforest instead (just a couple of fixes to Simon's work):

```
$ hg clone http://bitbucket.org/pmezard/hgforest-crew
```

After cloning hgforest-crew add an hgext.forest item with the path to hgforest-crew/forest.py in the extensions section in your `~/.hgrc` file.

You will need to create a Mercurial configuration file called `.hgrc` in your home directory (`~/.hgrc`). A minimum version requires the following:

```
$ cat ~/.hgrc
[ui]
username = Stephen Bannasch <stephen.bannasch@gmail.com>
[extensions]
fetch=
forest=/Users/Shared/mercurial/hgforest-crew/forest.py
```

If you have [MacPorts](#) installed, use:

```
$ sudo port install hg-forest
$ cat ~/.hgrc
```

```
[extensions]
forest=
fetch=
```

If you get an error from trying to clone the tree at this point (AttributeError: 'httprepository' object has no attribute 'do_read'), you should edit the forest.py files under /opt and change the occurrences of "do_read" to "_call".

Checkout the code using hg/forest.

```
$ hg fclone http://hg.openjdk.java.net/bsd-port/bsd-port
```

Here's what my bsd-port dir looks like:

```
$ cd bsd-port/
$ ls -l
total 528
-rw-r--r--  1 stephen  staff   1503 Dec 15 12:29 ASSEMBLY_EXCEPTION
-rw-r--r--  1 stephen  staff  19241 Dec 15 12:29 LICENSE
-rw-r--r--  1 stephen  staff  16336 Dec 15 12:29 Makefile
-rw-r--r--  1 stephen  staff   1207 Dec 15 12:29 README
-rw-r--r--  1 stephen  staff   87215 Jan 25 23:04 README-builds.html
-rw-r--r--  1 stephen  staff 127532 Dec 15 12:29 THIRD_PARTY_README
drwxr-xr-x  4 stephen  staff   136 Jan 26 00:51 build
-rwxr--r--@ 1 stephen  staff   373 Jan 26 03:00 build.sh
drwxr-xr-x 11 stephen  staff   374 Jan 25 23:06 corba
drwxr-xr-x 13 stephen  staff   442 Jan 25 23:06 hotspot
drwxr-xr-x 11 stephen  staff   374 Jan 25 23:06 jaxp
drwxr-xr-x 11 stephen  staff   374 Jan 25 23:06 jaxws
drwxr-xr-x 12 stephen  staff   408 Jan 25 23:07 jdk
drwxr-xr-x 12 stephen  staff   408 Jan 25 23:07 langtools
drwxr-xr-x 18 stephen  staff   612 Dec 15 12:29 make
```

Build OpenJDK

I added the script build.sh to keep track of the correct build invocation. The version below is adapted from Kurt Millers email here. Edit the paths to fit your dir structure. This script needs to be run with the source command.

build.sh

```
#!/bin/bash
env -i PATH=/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/X11/bin \
CC=gcc-4.0 \
CXX=g++-4.0 \
LANG=C \
make \
ALLOW_DOWNLOADS=true \
ALT_BOOTDIR=/usr/local/soylatt16-i386-1.0.3 \
ALT_JDK_IMPORT_PATH=/usr/local/soylatt16-i386-1.0.3 \
ALT_FREETYPE_HEADERS_PATH=/usr/X11R6/include \
ALT_FREETYPE_LIB_PATH=/usr/X11R6/lib \
ALT_CUPS_HEADERS_PATH=/usr/include \
ANT_HOME=/usr/share/ant \
NO_DOCS=true
```

Notes:

- JIBX was required previously, but is no longer required. If you are building an older version of OpenJDK you will need to download [JIBX](#) and add the following to your build script:

```
ALT_JIBX_LIBS_PATH=/Users/jyeary/Library/Java/PrivateExtensions/jibx-1.1.5/lib \
```

- gcc 4.0 and g++ 4.0 were required for the build. The defaults on Mac OS X 10.6.4 are gcc and g++ 4.2.1.

Build OpenJDK like this:

```
$ source build.sh
```

I have managed to compile the JDK in 64-bit mode on 10.6.4. It is listed as amd64, but it is running on Intel Core Duo. Here is my build script.

build64.sh

```
#!/bin/bash
env -i PATH=/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/X11/bin \
LANG=C \
CC=gcc-4.0 \
CXX=g++-4.0 \
make \
ALT_BOOTDIR=/usr/local/soylatte16-i386-1.0.3/ \
ALT_JDK_IMPORT_PATH=/usr/local/soylatte16-i386-1.0.3/ \
ALT_FREETYPE_HEADERS_PATH=/usr/X11R6/include \
ALT_FREETYPE_LIB_PATH=/usr/X11R6/lib \
ALT_CUPS_HEADERS_PATH=/usr/include \
ALLOW_DOWNLOADS=true \
ARCH_DATA_MODEL=64 \
ANT_HOME=/usr/share/ant \
NO_DOCS=true \
HOTSPOT_BUILD_JOBS=1
```

The final result:

```
BlueLotus:bsd-port jyeary$ uname -a
Darwin BlueLotus.local 10.4.0 Darwin Kernel Version 10.4.0: Fri Apr 23 18:28:53 PDT 2010; root:xnu-1504.7.4~1/RELEASE_I386 i386
BlueLotus:bsd-port jyeary$ java -version
openjdk version "1.7.0-internal"
OpenJDK Runtime Environment (build 1.7.0-internal-jyeary_2010_07_28_17_01-b00)
OpenJDK 64-Bit Server VM (build 19.0-b03, mixed mode)
```

One person reports: "This takes about 10m on my 2.5 GHz Intel Core 2 Duo MacBook Pro." Please check with the [BSD-Port mailing list](#) if you encounter problems.

Error Recovery

Try cleaning if you get errors.

Some errors when compiling are fixed by cleaning. You can temporarily add the clean command to the end of build.sh and source it to clean the build products.

Or you can try this suggestion from Kurt Miller:

```
$ mv build build.del; rm -rf build.del &
```

FreeType

If the build fails with the message:

```
FreeType version 2.3.0 or higher is required.
```

Install the latest version of [FreeType](#) .

Additionally, It has been reported:

```
I am not sure what can be done about this. My build.sh script has "ALT_FREETYPE_HEADERS_PATH=/opt/local/include/freetype2
ALT_FREETYPE_LIB_PATH=/opt/local/lib" and these seem to be where the headers and libraries are located....
```

```
WARNING: This build does not include running javadoc.\nERROR: FreeType version 2.3.0 or higher is required. \n /bin/mkdir -p /Users/ray/Projects/OpenJDK/bsd-port
/build/bsd-i586/btbins
rm -f /Users/ray/Projects/OpenJDK/bsd-port/build/bsd-i586/btbins/freetype_versioncheck
Failed to build freetypecheck. \n
Exiting because of the above error(s). \n
make: *** [post-sanity] Error 1
```

X11 is needed for build.

If you have not installed this with Mac OS X, or have deleted it, you need to install it. A copy can be obtained from <http://xquartz.macosforge.org/trac/wiki/X112.4.0>.

Update the bsd-port directory

If updates have been made to the tree, perhaps to fix a bug, you can update your sources with:

```
cd bsd-port ; hg fpull -u
```

Other Build Failures

Warnings are Errors

[Jeff Sinclair](#) noted a specific error which occurs during HotSpot compilation because the gcc.make file has a flag to set **WARNINGS** as **ERRORS**. This can be fixed by commenting the line in the gcc.make file located in `bsd-port/hotspot/make/bsd/makefiles/gcc.make`

```
# Compiler warnings are treated as errors
#WARNINGS_ARE_ERRORS = -Werror
```

Certificates are invalid (cacerts)

If you get an error message like the one below, it is an indication that the `/jre/lib/security/cacerts` file is invalid, or has no entries.

```
javax.net.ssl.SSLException: java.lang.RuntimeException: Unexpected error:
java.security.InvalidAlgorithmParameterException: the trustAnchors parameter must be non-empty
```

You can check it with the following:

```
keytool -list -keystore /usr/local/java-1.7.0/jre/lib/security/cacerts -storepass changeit

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 43 entries

entrustclientca, Jan 9, 2003, trustedCertEntry,
Certificate fingerprint (MD5): 0C:41:2F:13:5B:A0:54:F5:96:66:2D:7E:CD:0E:03:F4
```

```
verisignclass3g2ca, Mar 25, 2004, trustedCertEntry,
Certificate fingerprint (MD5): A2:33:9B:4C:74:78:73:D4:6C:E7:C1:F3:8D:CB:5C:E9
thawtepersonalbasicca, Feb 12, 1999, trustedCertEntry,
Certificate fingerprint (MD5): E6:0B:D2:C9:CA:2D:88:DB:1A:71:0E:4B:78:EB:02:41
addtrustclassica, May 2, 2006, trustedCertEntry,
...
```

If there is nothing there, you can use the `ALT_CACERTS_FILE=` to point to a copy of the cacerts file like the one in [SoyLatte](#).

Smoketest

If the build completes do a simple test by asking the JVM to print its version info. It should look something like this:

```
$ ../build/bsd-i586/j2sdk-image/bin/java -version
openjdk version "1.7.0-internal"
OpenJDK Runtime Environment (build 1.7.0-internal-stephen_2009_01_25_23_54-b00)
OpenJDK Server VM (build 14.0-b10, mixed mode)
```

Then, see how to [switch java versions on Mac OS X](#).

Using Hudson for Continuous Build.

Hudson will perfectly fit to set a continuous build system. You could find a more complete article with screenshot on my [blog](#)

You'll need :

- an OS/X box, under Snow Leopard, 32 and 64bits mode should works
- [XCode](#)
- Mercurial with hgforest extension
- Hudson with its [Mercurial Plugin](#)

Hudson jobs.

Defined 2 **free-style software project** jobs, one for building 32 bits JVM, **openjdk-1.7-i586**, the other to build 64 bits JVM, **openjdk-1.7-x86_64**.

Each one will use self sufficient script, these will download soylatte JVMs (i386/amd64) and jaxp, jaf and jaxws2 since these are not available from the url defined in ant build scripts.

Execute shell for openjdk-1.7-i586

```
#!/bin/sh
#
if [ ! -z "$HUDSON_HOME" ]; then
  DROP_DIR=$HUDSON_HOME/DROP_DIR
else
  DROP_DIR=`pwd`/DROP_DIR`
fi

DROP_DIR=$HUDSON_HOME/DROP_DIR

if [ ! -d $DROP_DIR ]; then
  echo "creating DROP_DIR"
  mkdir -p $DROP_DIR
fi

SOYLATTE=$DROP_DIR/soylatte16-i386-1.0.3

if [ ! -d $SOYLATTE ]; then

  if [ ! -f $DROP_DIR/soylatte16-i386-1.0.3.tar.bz2 ]; then
    echo "downloading soylatte16-i386-1.0.3 into DROP_DIR"
    curl --user jrl:"I am a Licensee in good standing" http://hg.bikemonkey.org/archive/javasrc_1_6_jrl_darwin/soylatte16-i386-1.0.3.tar.bz2 -o $DROP_DIR/soylatte16-i386-1.0.3.tar.bz2
```

```

fi

pushd $DROP_DIR
tar xjf $DROP_DIR/soylattel6-i386-1.0.3.tar.bz2
popd
fi

if [ ! -f $DROP_DIR/jaxp-1_4_4.zip ]; then
echo "downloading jaxp-1_4_4.zip into DROP_DIR"
curl http://icedtea.classpath.org/download/drops/jaxp-1_4_4.zip -o $DROP_DIR/jaxp-1_4_4.zip
fi

if [ ! -f $DROP_DIR/jdk7-jaf-2010_08_19.zip ]; then
echo "downloading jdk7-jaf-2010_08_19.zip into DROP_DIR"
curl http://icedtea.classpath.org/download/drops/jdk7-jaf-2010_08_19.zip -o $DROP_DIR/jdk7-jaf-2010_08_19.zip
fi

if [ ! -f $DROP_DIR/jdk7-jaxws2_2-2010_08_19.zip ]; then
echo "downloading jdk7-jaxws2_2-2010_08_19.zip into DROP_DIR"
curl http://icedtea.classpath.org/download/drops/jdk7-jaxws2_2-2010_08_19.zip -o $DROP_DIR/jdk7-jaxws2_2-2010_08_19.zip
fi

if [ ! -d ALT_COMPILER_PATH ]; then
echo "setup compiler dirs"
mkdir ALT_COMPILER_PATH
pushd ALT_COMPILER_PATH
ln -s /usr/bin .SOURCE
ln -s .SOURCE/g++-4.0 g++
ln -s .SOURCE/gcc-4.0 gcc
popd
fi

echo "cleaning previous build"

rm -rf build

unset CLASSPATH
unset JAVA_HOME
unset LD_LIBRARY_PATH

# patching defs.make for build on 64bits machine if required
grep -q x86_64 hotspot/make/bsd/makefiles/defs.make
if [ $? -eq 1 ] ; then
echo "patching defs.make for build on 64bits machine"
patch -p0 <<EOF
--- hotspot/make/bsd/makefiles/defs.make.orig 2010-11-20 16:09:49.000000000 +0100
+++ hotspot/make/bsd/makefiles/defs.make 2010-11-20 16:10:29.000000000 +0100
@@ -90,6 +90,24 @@
endif
endif

+# x86_64 OS/X
+ifeq (\$(ARCH), x86_64)
+ ifeq (\$(ARCH_DATA_MODEL), 64)
+ ARCH_DATA_MODEL = 64
+ MAKE_ARGS += LP64=1
+ PLATFORM = bsd-amd64
+ VM_PLATFORM = bsd_amd64
+ HS_ARCH = x86
+ else
+ ARCH_DATA_MODEL = 32
+ PLATFORM = bsd-i586
+ VM_PLATFORM = bsd_i486
+ HS_ARCH = x86
+ # We have to reset ARCH to i386 since SRCARCH relies on it
+ ARCH = i386
+ endif
+endif
+
# i386

```

```

ifeq (\$(ARCH), i386)
  ifeq (\$(ARCH_DATA_MODEL), 64)
EOF
fi

# patching Platform.gmk for 32bits VM build on 64bits machine
grep -q "force i586 if 32bits VM asked" jdk/make/common/shared/Platform.gmk
if [ $? -eq 1 ] ; then
echo "patching Platform.gmk for 32bits VM on 64bits machine"
patch -p0 <<EOF1
--- jdk/make/common/shared/Platform.gmk.orig      2010-11-20 19:33:19.000000000 +0100
+++ jdk/make/common/shared/Platform.gmk          2010-11-21 00:39:18.000000000 +0100
@@ -272,6 +272,12 @@

  # Darwin builds may be 32-bit or 64-bit data model.
  ifeq (\$(SYSTEM_UNAME), Darwin)
+   ifeq (\$(ARCH), amd64)
+     # force i586 if 32bits VM asked
+     ifeq (\$(ARCH_DATA_MODEL), 32)
+       ARCH=i586
+     endif
+   endif
  ifeq (\$(ARCH), i586)
    ifeq (\$(ARCH_DATA_MODEL), 64)
      ARCH=amd64
EOF1
fi

env -i PATH=/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/X11/bin \
LANG=C \
CC=gcc-4.0 \
CXX=g++-4.0 \
make \
ALT_DROPS_DIR=$DROP_DIR \
ALT_BOOTDIR=$SOYLATTE \
ALT_JDK_IMPORT_PATH=$SOYLATTE \
ALT_FREETYPE_HEADERS_PATH=/usr/X11R6/include \
ALT_FREETYPE_LIB_PATH=/usr/X11R6/lib \
ALT_CUPS_HEADERS_PATH=/usr/include \
ALLOW_DOWNLOADS=true \
ANT_HOME=/usr/share/ant \
NO_DOCS=true \
HOTSPOT_BUILD_JOBS=2 \
ARCH_DATA_MODEL=32 \
ALT_COMPILER_PATH=$(pwd -P)/ALT_COMPILER_PATH/ \
LD_LIBRARY_PATH=

echo "testing build: ./build/bsd-i586/j2sdk-image/bin/java -version"

./build/bsd-i586/j2sdk-image/bin/java -version

```

Execute shell for openjdk-1.7-x86_64

```

#!/bin/sh
#

if [ ! -z "$HUDSON_HOME" ]; then
  DROP_DIR=$HUDSON_HOME/DROP_DIR
else
  DROP_DIR=`pwd/DROP_DIR`
fi

DROP_DIR=$HUDSON_HOME/DROP_DIR

if [ ! -d $DROP_DIR ]; then
  echo "creating DROP_DIR"
  mkdir -p DROP_DIR
fi

```

```

SOYLATTE=$DROP_DIR/soylatte16-amd64-1.0.3

if [ ! -d $SOYLATTE ]; then

    if [ ! -f $DROP_DIR/soylatte16-amd64-1.0.3.tar.bz2 ]; then
        echo "downloading soylatte16-amd64-1.0.3 into DROP_DIR"
        curl --user jrl:"I am a Licensee in good standing" http://hg.bikemonkey.org/archive/javasrc_1_6_jrl_darwin/soylatte16-amd64-1.0.3.tar.bz2 -o $DROP_DIR/soylatte16-amd64-1.0.3.tar.bz2
    fi

    pushd $DROP_DIR
    tar xjf $DROP_DIR/soylatte16-amd64-1.0.3.tar.bz2
    popd
fi

if [ ! -f $DROP_DIR/jaxp-1_4_4.zip ]; then
    echo "downloading jaxp-1_4_4.zip into DROP_DIR"
    curl http://icedtea.classpath.org/download/drops/jaxp-1_4_4.zip -o $DROP_DIR/jaxp-1_4_4.zip
fi

if [ ! -f $DROP_DIR/jdk7-jaf-2010_08_19.zip ]; then
    echo "downloading jdk7-jaf-2010_08_19.zip into DROP_DIR"
    curl http://icedtea.classpath.org/download/drops/jdk7-jaf-2010_08_19.zip -o $DROP_DIR/jdk7-jaf-2010_08_19.zip
fi

if [ ! -f $DROP_DIR/jdk7-jaxws2_2-2010_08_19.zip ]; then
    echo "downloading jdk7-jaxws2_2-2010_08_19.zip into DROP_DIR"
    curl http://icedtea.classpath.org/download/drops/jdk7-jaxws2_2-2010_08_19.zip -o $DROP_DIR/jdk7-jaxws2_2-2010_08_19.zip
fi

if [ ! -d ALT_COMPILER_PATH ]; then
    echo "setup compiler dirs"
    mkdir ALT_COMPILER_PATH
    pushd ALT_COMPILER_PATH
    ln -s /usr/bin .SOURCE
    ln -s .SOURCE/g++-4.0 g++
    ln -s .SOURCE/gcc-4.0 gcc
    popd
fi

echo "cleaning previous build"

rm -rf build

unset CLASSPATH
unset JAVA_HOME
unset LD_LIBRARY_PATH

# patching defs.make for build on 64bits machine if required
grep -q x86_64 hotspot/make/bsd/makefiles/defs.make
if [ $? -eq 1 ]; then
    echo "patching defs.make for build on 64bits machine"
    patch -p0 <<EOF
--- hotspot/make/bsd/makefiles/defs.make.orig 2010-11-20 16:09:49.000000000 +0100
+++ hotspot/make/bsd/makefiles/defs.make 2010-11-20 16:10:29.000000000 +0100
@@ -90,6 +90,24 @@
    endif
endif

+# x86_64 OS/X
+ifeq (\$(ARCH), x86_64)
+ ifeq (\$(ARCH_DATA_MODEL), 64)
+   ARCH_DATA_MODEL = 64
+   MAKE_ARGS += LP64=1
+   PLATFORM = bsd-amd64
+   VM_PLATFORM = bsd-amd64
+   HS_ARCH = x86
+ else
+   ARCH_DATA_MODEL = 32

```



```

+ PLATFORM      = bsd-i586
+ VM_PLATFORM   = bsd_i486
+ HS_ARCH       = x86
+ # We have to reset ARCH to i386 since SRCARCH relies on it
+ ARCH          = i386
+ endif
+endif
+
# i386
ifeq (\$(ARCH), i386)
  ifeq (\$(ARCH_DATA_MODEL), 64)
EOF
fi

# patching Platform.gmk for 32bits VM build on 64bits machine
grep -q "force i586 if 32bits VM asked" jdk/make/common/shared/Platform.gmk
if [ $? -eq 1 ] ; then
echo "patching Platform.gmk for 32bits VM on 64bits machine"
patch -p0 <<EOF1
--- jdk/make/common/shared/Platform.gmk.orig      2010-11-20 19:33:19.000000000 +0100
+++ jdk/make/common/shared/Platform.gmk          2010-11-21 00:39:18.000000000 +0100
@@ -272,6 +272,12 @@

# Darwin builds may be 32-bit or 64-bit data model.
ifeq (\$(SYSTEM_UNAME), Darwin)
+ ifeq (\$(ARCH), amd64)
+ # force i586 if 32bits VM asked
+ ifeq (\$(ARCH_DATA_MODEL), 32)
+ ARCH=i586
+ endif
+ endif
ifeq (\$(ARCH), i586)
  ifeq (\$(ARCH_DATA_MODEL), 64)
    ARCH=amd64
EOF1
fi

env -i PATH=/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/X11/bin \
LANG=C \
CC=gcc-4.0 \
CXX=g++-4.0 \
make \
ALT_DROPS_DIR=$DROP_DIR \
ALT_BOOTDIR=$SOYLATTE \
ALT_JDK_IMPORT_PATH=$SOYLATTE \
ALT_FREETYPE_HEADERS_PATH=/usr/X11R6/include \
ALT_FREETYPE_LIB_PATH=/usr/X11R6/lib \
ALT_CUPS_HEADERS_PATH=/usr/include \
ALLOW_DOWNLOADS=true \
ANT_HOME=/usr/share/ant \
NO_DOCS=true \
HOTSPOT_BUILD_JOBS=2 \
ARCH_DATA_MODEL=64 \
ALT_COMPILER_PATH=$(pwd -P)/ALT_COMPILER_PATH/ \
LD_LIBRARY_PATH=

echo "testing build: ./build/bsd-amd64/j2sdk-image/bin/java -version"

./build/bsd-amd64/j2sdk-image/bin/java -version

```