HOWTO: ARC (libobjc2) and libdispatch on Ubuntu 12.04 Server

From: Patryk Laurent
Subject: HOWTO: ARC (libobjc2) and libdispatch on Ubuntu 12.04 Server
Date: Sat, 11 Aug 2012 12:52:58 -0400

```
Greetings,

Below are some step-by-step instructions on installing libobjc2 and libdispatch
on Ubuntu 12.04 server.

Best,
Patryk




# ----------------------------------------------------------------
# Objective C 2.0 on Ubuntu (from source)
# ----------------------------------------------------------------
# PART 1: LIBOBJC2 from source (with ARC)
# PART 2: LIBDISPATCH from source
# ON UBUNTU 12.04 SERVER
# ----------------------------------------------------------------
# Patryk Laurent (address@hidden)
# Step 1 based on David Chisnall
(http://etoileos.com/news/archive/2011/08/14/1206/)
# Step 2 based on Chris Mowforth
http://chris.mowforth.com/installing-grand-central-dispatch-on-linux)
# ----------------------------------------------------------------
# August 11, 2012
# ----------------------------------------------------------------

# ==============================
# ==============================
# PART 1: new libobjc2 on Ubuntu
# ==============================
# ==============================

# -------------------------------------------------------------------
# Some examples to test Objective C and ARC once we get it installed
# -------------------------------------------------------------------

cd ~

cat > Fraction.h << EOF

#import <Foundation/NSObject.h>

@interface Fraction: NSObject {
    int numerator;
    int denominator;
}

-(void) print;
-(void) setNumerator: (int) n;
-(void) setDenominator: (int) d;
-(int) numerator;
-(int) denominator;
@end

EOF
```

```
cat > Fraction.m << EOF
#import "Fraction.h"
#import <stdio.h>

@implementation Fraction
-(void) print {
    printf( "%i/%i", numerator, denominator );
}

-(void) setNumerator: (int) n {
    numerator = n;
}

-(void) setDenominator: (int) d {
    denominator = d;
}

-(int) denominator {
    return denominator;
}

-(int) numerator {
    return numerator;
}
@end

EOF



cat > main.m << EOF

#import <stdio.h>
#import "Fraction.h"

int main( int argc, const char *argv[] ) {
    // create a new instance
    Fraction *frac = [[Fraction alloc] init];

    // set the values
    [frac setNumerator: 1];
    [frac setDenominator: 3];

    // print it
    printf( "The fraction is: " );
    [frac print];
    printf( "\n" );

    // free memory
    [frac release];

    return 0;
}

EOF

cat > mainarc.m << EOF
#import <stdio.h>
#import "Fraction.h"

int main( int argc, const char *argv[] ) {
    // create a new instance
    Fraction *frac = [[Fraction alloc] init];

    // set the values
    [frac setNumerator: 1];
    [frac setDenominator: 3];

    // print it
    printf( "The fraction is: " );
    [frac print];
    printf( "\n" );
    // free memory
    // [frac release];  // valgrind should show less leakage with -fobjc-arc

    return 0;
}

EOF


# ------------------------------------------------------------------
# INITIAL REQUIREMENTS
# ------------------------------------------------------------------
sudo apt-get -y install build-essential subversion clang libicu-dev libxml2-dev
libxml2  libgnutls-dev libssl-dev

#sudo apt-get -y install gnustep                 # If you want old runtime
#sudo apt-get -y install gnustep-make
#sudo apt-get -y install libgnustep-base-dev

sudo apt-get -y install gobjc                   # Def required for below.

# ---------------------------------------------------------------------
# TEST (may fail w/ segfault if you did not apt-get install gnustep)
# ---------------------------------------------------------------------

cd ~
gcc `gnustep-config --objc-flags` main.m  Fraction.m -o test -lobjc
-lgnustep-base
./test
clang `gnustep-config --objc-flags` main.m  Fraction.m -o test -lobjc
-lgnustep-base
./test

# ---------------------------------------------------------------------
# OK, let's install the new GNUstep from Subversion repositories!
# (based on David Chisnall http://etoileos.com/news/archive/2011/08/14/1206/)
# ---------------------------------------------------------------------
mkdir gs
cd gs

svn co svn://svn.gna.org/svn/gnustep/tools/make/trunk make
svn co http://svn.gna.org/svn/gnustep/modules/core
svn co svn://svn.gna.org/svn/gnustep/libs/libobjc2/trunk libobjc

# ---------------------------------------------------------------------
# 1) Install GNUstep Make a first time.
# ---------------------------------------------------------------------

cd ~/gs/make
 export CC=clang
 export CXX=clang++
 ./configure --enable-debug-by-default --with-layout=fhs
 make && sudo -E make install
 . /usr/local/share/GNUstep/Makefiles/GNUstep.sh
cd ..

cd ~/gs/core/base
 ./configure
 make             # On this FIRST TIME THRU, WILL SAY CAN'T BUILD NSBLOCKS for
this runtime
 sudo make install
cd ..

# ---------------------------------------------------------------------
# TEST (the resulting binary will segfault if we don't have a runtime)
# ---------------------------------------------------------------------
cd ~
clang `gnustep-config --objc-flags` main.m  Fraction.m -o test -lobjc
-lgnustep-base


# ---------------------------------------------------------------------
# 2) Build libobjc2
# -------------------------------------------------------------------
cd ~/gs/libobjc
 make -f Makefile
 sudo make -f Makefile install
cd ..

# -------------------------------------------------------------------
# 3) NOW GO BACK, RECOMPILE GNUStep MAKE (TO DETECT THE NEW OBJC RUNTIME)
# -------------------------------------------------------------------

cd ~/gs/make
 ./configure  --enable-objc-nonfragile-abi --enable-native-objc-exceptions
--with-layout=fhs --enable-debug-by-default CC=clang CXX=clang++
 make && sudo -E make install
 . /usr/local/share/GNUstep/Makefiles/GNUstep.sh
cd ..

# -------------------------------------------------------------------
# 4) AND THEN RECOMPILE CORE/BASE
# -------------------------------------------------------------------

cd ~/core/base
 ./configure --disable-mixedabi CC=clang CXX=clang++
 make            # THIS TIME THRU, NO COMPLAINTS ABOUT BLOCKS
 sudo make install
cd ..

# -------------------------------------------------------------------
# 5) FINALLY TEST AGAIN AND ENJOY OBJECTIVE C WITH ARC
# Note that I need to add GNUSTEP-CONFIG --OBJC-LIBS below.
# If you don't want ARC, omit -fobj-arc
# -------------------------------------------------------------------

cd ~
clang `gnustep-config --objc-flags` `gnustep-config --objc-libs` -fobj-arc
-fobjc-nonfragile-abi mainarc.m  Fraction.m -o test -lobjc -lgnustep-base
./test



# ============================
# ============================
# PART 2: libdispatch on Ubuntu
# ============================
# ============================




# -------------------------------------------------------------------
# Some examples to test GCD once we get it installed
# -------------------------------------------------------------------

cd ~

cat > helloGCD.c << EOF
#include <dispatch/dispatch.h>
#include <stdio.h>

int main() {
  dispatch_queue_t queue = dispatch_queue_create(NULL, NULL);

  dispatch_sync(queue, ^{
    printf("Hello, world from a dispatch queue!\n");
  });

  dispatch_release(queue);

  return 0;
}

EOF

cat > helloGCD_objc.c << EOF
```

```
#include <dispatch/dispatch.h>
#import <stdio.h>
#import "Fraction.h"

int main( int argc, const char *argv[] ) {
    dispatch_queue_t queue = dispatch_queue_create(NULL, NULL);
    Fraction *frac = [[Fraction alloc] init];

    [frac setNumerator: 1];
    [frac setDenominator: 3];

    // print it
    dispatch_sync(queue, ^{
      printf( "The fraction is: " );
      [frac print];
      printf( "\n" );
    });
    dispatch_release(queue);

    return 0;
}

EOF

# -------------------------------------------------------------------
# INSTALLING LIBDISPATCH
# (based on Chris Mowforth
http://chris.mowforth.com/installing-grand-central-dispatch-on-linux)
# -------------------------------------------------------------------

sudo apt-get install clang libblocksruntime-dev libkqueue-dev

# Visit  http://packages.ubuntu.com/oneiric/libpthread-workqueue0 for download
links
# Visit  http://packages.ubuntu.com/oneiric/libpthread-workqueue-dev for
download link

mkdir dispatch
cd dispatch
 sudo apt-get install make autoconf autogen libtool build-essential gcc-multilib
 sudo apt-get install pkg-config

 wget
http://mirror.pnl.gov/ubuntu//pool/universe/libp/libpthread-workqueue/libpthread-workqueue0_0.8.2-1_amd64.deb
 wget
http://mirror.pnl.gov/ubuntu//pool/universe/libp/libpthread-workqueue/libpthread-workqueue-dev_0.8.2-1_amd64.deb

 sudo dpkg -i libpthread-workqueue0_0.8.2-1_amd64.deb
 sudo dpkg -i libpthread-workqueue-dev_0.8.2-1_amd64.deb

 wget
http://archive.ubuntu.com/ubuntu/pool/universe/libd/libdispatch/libdispatch_0~svn197.orig.tar.gz

 tar xvfz libdispatch_0~svn197.orig.tar.gz
 cd libdispatch-0~svn197/
  export CC=clang
  export CXX=clang++
  make distclean
  ./configure
  make

  # dispatch_starfish.o: In function `_dispatch_time_mach2nano':
  # dispatch_starfish.c:(.text+0x5bc): undefined reference to
`_dispatch_host_time_data'
  # dispatch_starfish.c:(.text+0x5ea): undefined reference to
`_dispatch_get_host_time_init'
  # /usr/bin/ld: .libs/dispatch_starfish: hidden symbol
`_dispatch_host_time_data' isn't defined
  # /usr/bin/ld: final link failed: Bad value
  # clang: error: linker command failed with exit code 1 (use -v to see
invocation)
  # make[1]: *** [dispatch_starfish] Error 1
  # make[1]: Leaving directory
`/home/patryk/dispatch/libdispatch-0~svn197/testing'
```

```
  # make: *** [all-recursive] Error 1

  # -------------------------------------------------------------------
  # To fix compile, comment out build of "testing" from Makefile
  # -------------------------------------------------------------------

  make clean
  sed "s/testing/#testing/" Makefile > Makefile.new
  mv Makefile.new Makefile
  make
  sudo make install
  sudo ldconfig

# -------------------------------------------------------------------
# Testing
# -------------------------------------------------------------------
# Plain C: When not compiling with libobjc2 (just plain C) on Ubuntu you must
add -lBlocksRuntime
# -------------------------------------------------------------------

clang -o hi helloGCD.c -fblocks -ldispatch -lBlocksRuntime
./hi

clang -o hi helloGCD.c -fblocks -ldispatch
# /usr/bin/ld: /tmp/helloGCD-eXxFYY.o: undefined reference to symbol
'_NSConcreteGlobalBlock'
# /usr/bin/ld: note: '_NSConcreteGlobalBlock' is defined in DSO
/usr/lib/libBlocksRuntime.so.0
# so try adding it to the linker command line
# /usr/lib/libBlocksRuntime.so.0: could not read symbols: Invalid operation
# clang: error: linker command failed with exit code 1 (use -v to see
invocation)



# -------------------------------------------------------------------
# Compiling Objective C with ARC and blocks and libdispatch
# -------------------------------------------------------------------
# Note: do not use -lBlocksRuntime here since Apple on is not
# compatible with the libobjc2 one which has its own. (Note from
# David Chisnall)
# -------------------------------------------------------------------

clang `gnustep-config --objc-flags` `gnustep-config --objc-libs` -fobj-arc
-fobjc-nonfragile-abi -fblocks  helloGCD_objc.m Fraction.m -o test -lobjc
-lgnustep-base -ldispatch
./test
```

---

reply via email to

Patryk Laurent

[Prev in Thread] **Current Thread** [Next in Thread]

- **HOWTO: ARC (libobjc2) and libdispatch on Ubuntu 12.04 Server**, *Patryk Laurent* **<=**
  - **Re: HOWTO: ARC (libobjc2) and libdispatch on Ubuntu 12.04 Server**, *Niels Grewe*, 2012/08/12
    - **Re: HOWTO: ARC (libobjc2) and libdispatch on Ubuntu 12.04 Server**, *Patryk Laurent*, 2012/08/12
  - **Re: HOWTO: ARC (libobjc2) and libdispatch on Ubuntu 12.04 Server**, *Ivan Vučica*, 2012/08/13
    - **Re: HOWTO: ARC (libobjc2) and libdispatch on Ubuntu 12.04 Server**, *Thomas Davie*, 2012/08/13
      - **Re: HOWTO: ARC (libobjc2) and libdispatch on Ubuntu 12.04 Server**, *Niels Grewe*, 2012/08/13

- Prev by Date: **Re: TextMate @ GitHub**
- Next by Date: **Re: TextMate @ GitHub**