

Updating Txlib based project for Xcode/LLDB debugging

Steps need to add 'Xcode' build/debug to an existing TXLIB based project: 'sample'

- Make sure the default 'make' target is building a **64-bit** version of the project: m64r/m64d
- Create a new Xcode project under the '/c' directory called 'sample-debug' (or 'dfsee')
 - Menu: File -> New project ... [Cross Platform] ... [External Build System]
Product name: 'sample-debug' (or 'dfsee')
Identifier: com.dfsee
 - Move the 'sample-debug.xcodeproj' to '/c/sample'
 - Remove the obsolete '/c/sample-debug' directory
- Open the moved projectfile in Xcode
 - Add the project code files, like sample.c
Set the 'Text settings' indent using spaces to 3, 3
 - Menu: Project -> Scheme -> edit scheme ... [> Run]
Executable: [.. ^] choose debug-executable from make, e.g. -build-/samplem64d
Debug process as: (.) root
Launch: (.) Wait for executable to be launched
 - RUN or Cmd+R should now build and debug the project (set breakpoint on main)
- Add the TX library sources for reference/debug
 - From 'Project Navigator' sample-debug context-menu:
New Group, rename after create to: 'Txlib'
Add files to "sample-debug" ...

Add all the Txlib source-files in turn, and the Txlib 'include' directory
OR
Just add the complete 'txlib' directory in one go (gives some clutter, but is easier)

Note:

The debugger will find the Txlib sources anyway, while debugging (stepping), but to set any breakpoints in code that has not been loaded yet by the debugger, it is more convenient to have them available in the navigator.

To debug something, set breakpoints as wanted/needed, and RUN (> button or Cmd-R).
The debugger will start, and show 'waiting for samplem64d to launch' is a title-status window

Now, start that executable in a Terminal (or iTerm) window, and the debugger should attach, and hit the breakpoints set, when appropriate (works with 'sudo ./samplem64d' too).

Example overall makefile for an Xcode/TXlib debug project

Located in main project directory, calling the variant makefiles

```
#overall makefile to make some or all deliverables (64bit debug built by default)

one: m64d

all: release debug m64r m64d

release:
    make -C _build_/macr

debug:
    make -C _build_/macd

m64r:
    make -C _build_/m64r

m64d:
    make -C _build_/m64d

clean:
    make -C _build_/macr clean
    make -C _build_/macd clean
    make -C _build_/m64r clean
    make -C _build_/m64d clean
```

Example variant makefiles for an Xcode/TXlib debug project

Located in variant directory (_build_/m64d), calling the master makefile.osx

```
#variant makefile, debug 64-bit
all:
    make target_env=debug target_arch=64bit tx_version=txall_ -f ../../makefile.osx | tee b.log
clean:
    make -f ../../makefile.osx clean
```

```
#variant makefile, release 64-bit
all:
    make target_env=release target_arch=64bit tx_version=txall_ -f ../../makefile.osx | tee
b.log
clean:
    make -f ../../makefile.osx clean
```

Example master makefile.osx for an Xcode/TXlib debug project

```
#
# DFSee master makefile for MAC OS X (DARWIN), using standard make and gcc
#
# JvW 27-08-2017 Initial version, derived from DFS OSX makefile, added lib dependency
# JvW 28-08-2017 Updated for sam2, and adding 64-bit variants for debugging in Xcode
#
# Usage: Include this from 2nd level subdirectory with a 'makefile' that set the
#       various environmental variables like 'target_env'

# define main component name
compo    = txt

cc        = gcc
cflags    = -Wall -Wno-invalid-source-encoding -funsigned-char

ifndef tx_version
tx_version = _txall_
endif
ifeq ($(tx_version), _txmin_)
cflags += -DTXMIN
endif

txdir     = ../../../txlib/$(tx_version)/lib
txinc     = ../../../txlib/include

cflags += -I $(txinc) -I ../../
lnkopt    = -L$(txdir)
postfix   =

ifeq ($(target_env), debug)
cflags += -g
#activate next line to remove trace from debug EXE
#cflags += -DNOTRACE
postfix = d
else
postfix = r
#select optimization for the non-debug (t and r) versions
cflags += -O2
endif

ifeq ($(target_arch), 64bit)
bintype = m64
cflags += -m64
lnkopt += -arch x86_64
else
bintype = mac
cflags += -m32
lnkopt += -arch i386
endif

gendep = makefile ../../makefile.osx \
        $(txinc)/txlib.h $(txinc)/txcon.h $(txinc)/txwin.h \
        $(txdir)/libtx$(bintype)$(postfix).a

%.o : ../../%.c $(gendep)
      $(cc) -c $(cflags) $< -o $@

exename = ../$(compo)$(bintype)$(postfix)

all : $(exename)

#list all object files for the exe
exeobjs = $(compo).o

$(exename) : $(gendep) $(exeobjs) $(txdir)/libtx$(bintype)$(postfix).a
      $(cc) $(lnkopt) -o $(exename) $(exeobjs) -ltx$(bintype)$(postfix)

clean :
      rm -f *.o b.log
```