

```

#
# configure.ac
#
# Authors:
#   Eric Butler <eric@codebutler.com>
#   Nick Kossifidis <mickflemm@gmail.com>
#
# This file is part of Firesheep.
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program. If not, see <http://www.gnu.org/licenses/>.

AC_INIT([Firesheep], [0.1], [eric@codebutler.com],
        [firesheep], [http://codebutler.github.com/firesheep])
AC_PREREQ([2.61])
AM_INIT_AUTOMAKE([foreign])

AC_PROG_CXX

# OSX doesn't come with pkg-config, but PKG_CHECK_MODULES has to be defined to
# something to avoid errors.
m4_define_default([PKG_CHECK_MODULES], [AC_MSG_FAILURE([pkg-config was not
found]])])

AC_CANONICAL_HOST
case "$host_os" in
    cygwin)
        TARGET=WIN32
        ;;
    darwin*)
        CFLAGS="-isysroot /Developer/SDKs/MacOSX10.5.sdk -arch i386 -
mmacosx-version-min=10.5"
        TARGET=OSX
        ;;
    linux*)
        TARGET=LINUX
        ;;
esac

AM_CONDITIONAL(PLATFORM_WIN32, test x$TARGET = xWIN32)
AM_CONDITIONAL(PLATFORM_OSX, test x$TARGET = xOSX)
AM_CONDITIONAL(PLATFORM_LINUX, test x$TARGET = xLINUX)

# BEGIN MOZILLA STUFF (XULRUNNER SDK)
# Mozilla doesn't provide pkg-config files for OSX, hence all this mess.

if test x$TARGET = xOSX; then
    AC_ARG_WITH([xulrunner-sdk],
        [AS_HELP_STRING([--with-xulrunner-sdk=<path>], [path to xulrunner
sdk])],
        [xulrunner_sdk="$withval"],
        [xulrunner_sdk=""])

```

```

if test x$TARGET = xWIN32; then
    xulrunner_sdk=`readlink -f $xulrunner_sdk`
fi

AS_IF([test -d "$xulrunner_sdk"], [], [AC_MSG_FAILURE([Specify --with-
xulrunner-sdk=<path>.] )])

MOZ_SDK=$xulrunner_sdk
AC_SUBST(MOZ_SDK)

MOZ_PATH="{xulrunner_sdk}/bin"
AC_SUBST(MOZ_PATH)

AC_PATH_PROG([XPIDL], [xpidl], [no], [$MOZ_PATH:$PATH])
if test "x$XPIDL" = "xno"; then
    AC_MSG_ERROR([xpidl compiler not found])
fi
AC_SUBST(XPIDL)

XPIDL_FLAGS="-I {xulrunner_sdk}/idl"
AC_SUBST(XPIDL_FLAGS)

MOZ_CFLAGS="-fshort-wchar -I${MOZ_SDK}/include"
AC_SUBST(MOZ_CFLAGS)

MOZ_LIBS="-L${MOZ_SDK}/lib -L${MOZ_SDK}/bin -Wl,-
executable_path,{MOZ_SDK}/bin/xulrunner-bin -lxpcomglue_s -lxpcom -lplds4 -lplc4
-lnspr4 -lpthread -ldl"
AC_SUBST(MOZ_LIBS)
else
    PKG_CHECK_MODULES(MOZ, [libxul >= 1.9.2])
    AC_SUBST(MOZ_CFLAGS)
    AC_SUBST(MOZ_LIBS)

    MOZ_SDK=`pkg-config libxul --variable=sdkdir`
    AC_SUBST(MOZ_SDK)

    MOZ_PATH="{MOZ_SDK}/bin"
    AC_SUBST(MOZ_PATH)

    XPIDL_FLAGS="-I {MOZ_SDK}/idl"
    AC_SUBST(XPIDL_FLAGS)

    AC_PATH_PROG([XPIDL], [xpidl], [no], [$MOZ_PATH:$PATH])
    if test "x$XPIDL" = "xno"; then
        AC_MSG_ERROR([xpidl compiler not found])
    fi
    AC_SUBST(XPIDL)
fi

# END MOZILLA STUFF

# BEGIN PCAP LIBS
AC_PATH_PROG([PCAP_CONFIG], [pcap-config], [no], [$PATH])
if test "x$PCAP_CONFIG" = "xno"; then
    AC_MSG_ERROR([pcap-config not found (libpcap not installed ?)])
fi

PCAP_LIBS=`pcap-config --libs`
AC_SUBST(PCAP_LIBS)
PCAP_CFLAGS=`pcap-config --cflags`
AC_SUBST(PCAP_FLAGS)

```

```

# END PCAP LIBS

# BEGIN BOOST LIBS
# Specific version ?
#BOOST_REQUIRE([1.40])
BOOST_REQUIRE
AC_SUBST(BOOST_CPPFLAGS)
BOOST_FORMAT
AC_SUBST(BOOST_FORMAT_LIBS)
BOOST_STRING_ALGO
AC_SUBST(BOOST_STRING_ALGO_LIBS)
# END BOOST LIBS

# BEGIN LINUX
# Use generic CPU arch for xpi/platform
# on Linux
case "$build_cpu" in
    i[[3456]]86*)
        build_arch="x86"
        build_target="-mtune=generic -m32"
        ;;
    amd64*|x86_64*)
        build_arch="x86_64"
        build_target="-mtune=generic -m64"
        ;;
esac

BUILD_ARCH=$build_arch
AC_SUBST(BUILD_ARCH)

BUILD_TARGET=$build_target
AC_SUBST(BUILD_TARGET)

if test x$TARGET = xLINUX; then
    PKG_CHECK_MODULES(HAL, [hal])
    AC_SUBST(HAL_CFLAGS)
    AC_SUBST(HAL_LIBS)
fi
# END LINUX

AC_SUBST(CFLAGS)

AC_CONFIG_FILES([
Makefile
backend/Makefile
mozpopen/Makefile
])

AC_OUTPUT

```