

Cplex su Mac OS X: Guida Rapida

di Alessandro Beltramin

8 marzo 2015

Prerequisiti

Scaricare ed installare i seguenti software:

- Cplex
- CMake : tool multiplatforma per la gestione della compilazione (<http://www.cmake.org/download/>)
- Xcode : ambiente di sviluppo integrato per Mac OS (<https://developer.apple.com/xcode/downloads/>)

Note

Guida rapida per poter programmare in C / C++ su Mac OS X, utilizzando le librerie di Cplex.

L'utilizzo di un IDE come Xcode permette l'autocompletamento e la compilazione dinamica del codice, evidenziando eventuali errori in tempo reale.

CMake evita di dover configurare manualmente i parametri di compilazione: si occupa della configurazione dell'IDE o, in alternativa, di generare automaticamente il *Makefile* necessario alla compilazione.

La prima installazione e configurazione di tutto il necessario richiede circa 3 ore.

Installazione e configurazione Cplex

1. Per assegnare i permessi di esecuzione al file di installazione ed avviare l'installazione:

```
chmod +x ./CPLEX_OPT_STUD_12.6.1_FOR_OSX.bin  
./CPLEX_OPT_STUD_12.6.1_FOR_OSX.bin
```

2. Al termine, tenere a mente il percorso scelto in fase di installazione ed eventualmente creare un link per poter eseguire Cplex da qualunque posizione:

```
sudo ln -s /Users/#UTENTE#/Applications/IBM/ILOG/CPLEX_Studio1261/cplex/bin  
/x86-64_osx/cplex /usr/bin/cplex
```

3. Per verificare che la compilazione avvenga correttamente, cercare la sottodirectory *examples/-platform/libformat* in base alla propria piattaforma e tipo di compilatore. Aprire una finestra di terminale su quel percorso, in cui dovrebbe esserci un file chiamato *Makefile*. Provare ad eseguire un esempio in C++ con i comandi:

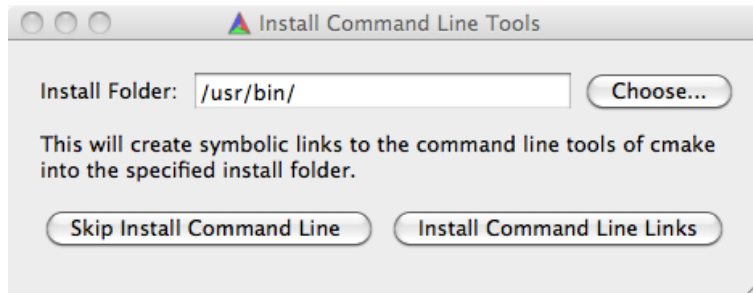
```
make ilolpex1  
ilolpex1 -r
```

Configurazione CMake

1. Al termine dell'installazione di CMake, eseguire l'interfaccia grafica di CMake con i permessi di root :

```
cd /Applications/CMake.app/Contents/bin
sudo cmake-gui
```

2. Per creare automaticamente i link per l'utilizzo da riga di comando, andare in Tools -> "Install For Command Line Usage" :



3. Aggiustare opportunamente i percorsi e copiare il modulo *FindCPLX.cmake* in:
/Applications/CMake.app/Contents/share/cmake-3.2/Modules

Configurazione ambiente di sviluppo

Si può procedere allo sviluppo di una applicazione, creando tutti i files sorgenti necessari (ad es. main.cpp, ufl.cpp, ufl.h). E' necessario inoltre copiare il file *CMakeLists.txt* nella directory principale di progetto, per specificare di quali files CMake dovrà gestire la compilazione (da modificare opportunamente).

L'albero delle directory di progetto potrebbe essere:

- src/
 - main.cpp
 - ufl.h
 - ufl.cpp
- CMakeLists.txt

Compilazione tramite XCode

1. Nella directory principale di progetto, eseguire:

```
mkdir buildXcode && cd buildXcode
cmake .. -G "Xcode"
```
2. Nella cartella *buildXcode* vengono creati automaticamente i files di progetto: aprire *UFL.xcodeproj* con XCode, selezionare come build "*UFL -> My Mac 64-bit*".

Compilazione tramite Makefile

1. Nella directory principale di progetto, eseguire:

```
mkdir buildMake && cd buildMake
cmake ..
```
2. Per compilare ed eseguire l'applicazione (ad es. *ufl*):

```
make && ./ufl ../data/B1.1
```

File allegati

FindCplex.cmake

```
1 SET(CPLEX_ROOT_DIR "" CACHE PATH "CPLEX root directory")
2
3 FIND_PATH(CPLEX_INCLUDE_DIR
4     ilcplex/cplex.h
5     PATHS "/Users/#USER#/Applications/IBM/ILOG/CPLEX_Studio1261/cplex/include"
6     HINTS ${CPLEX_ROOT_DIR}/include )
7
8 FIND_LIBRARY(CPLEX_LIBRARY
9     libcplex.a
10    PATHS "/Users/#USER#/Applications/IBM/ILOG/CPLEX_Studio1261/cplex/lib/x86-64
11         _osx/static_pic"
12    HINTS ${CPLEX_ROOT_DIR}/lib )
13
14 INCLUDE(FindPackageHandleStandardArgs)
15 FIND_PACKAGE_HANDLE_STANDARD_ARGS(CPLEX DEFAULT_MSG CPLEX_LIBRARY CPLEX_INCLUDE_DIR)
16 FIND_PATH(CPLEX_BIN_DIR
17     cplex
18     PATHS "/Users/#USER#/Applications/IBM/ILOG/CPLEX_Studio1261/cplex/bin/x86-64
19         _osx" )
20
21 IF(CPLEX_FOUND)
22     SET(CPLEX_INCLUDE_DIRS ${CPLEX_INCLUDE_DIR})
23     SET(CPLEX_LIBRARIES ${CPLEX_LIBRARY})
24     IF(CMAKE_SYSTEM_NAME STREQUAL "Linux")
25         SET(CPLEX_LIBRARIES "${CPLEX_LIBRARIES};m;pthread")
26     ENDIF(CMAKE_SYSTEM_NAME STREQUAL "Linux")
27 ENDIF(CPLEX_FOUND)
28
29 MARK_AS_ADVANCED(CPLEX_LIBRARY CPLEX_INCLUDE_DIR CPLEX_BIN_DIR)
```

CMakeLists.txt

```
1 cmake_minimum_required(VERSION 2.8)
2
3 PROJECT (UFL)
4
5 FIND_PACKAGE(CPLEX REQUIRED)
6
7 INCLUDE_DIRECTORIES( include ${CPLEX_INCLUDE_DIRS} )
8
9 add_executable(ufl src/ufl.cpp src/ufl.h src/main.cpp)
10
11 TARGET_LINK_LIBRARIES( ufl ${CPLEX_LIBRARIES} )
```