

# Installation of IPOPT for Windows OS

## 32/64 bit, incl. MATLAB Interface

Document created by Dino Hüllmann on March 11<sup>th</sup> 2013, last update on March 13<sup>th</sup> 2013.  
Special thanks go to Tony Kelman who supported me greatly and advised me of a malicious bug.

### Step 1: MinGW installation

In case a (outdated) version of MinGW is already installed on your system you may

- replace it by the most (or a more) recent release (just delete the directory MinGW is installed to before installing the new version),
- keep the old one and install the newer release at a different location OR
- try to build Ipopt using the existing installation

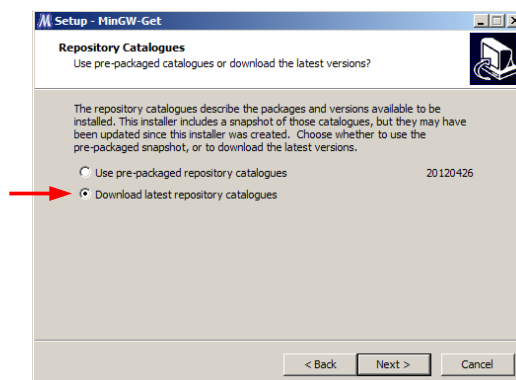
It is recommended to download the latest version of the MinGW installer from:

<http://sourceforge.net/projects/mingw/files/Installer/mingw-get-inst/>

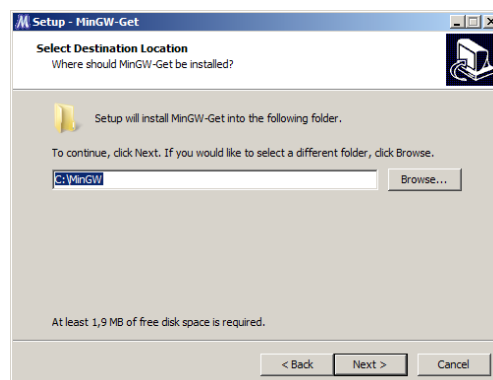
(for example: mingw-get-inst-20120426.exe)

Run the installer.

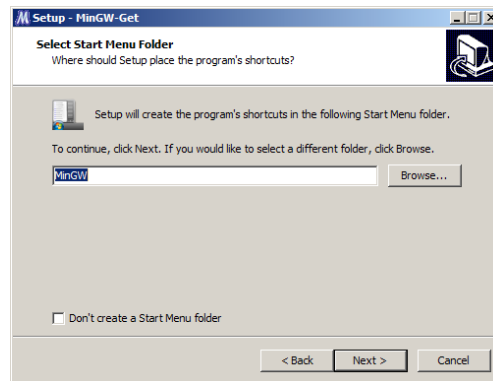
You are asked whether you want to use the pre-packaged repository catalogues or to download the latest ones. Select the latter option.



Next you have to state where MinGW should be installed. It is important that the path you choose does not contain any space or special characters. In the scope of this document it is assumed that MinGW is installed to C:\MinGW. If your installation location differs from this you have to adapt all paths stated below accordingly.



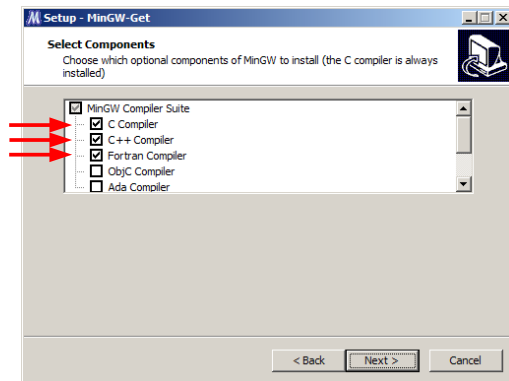
I recommend to create shortcuts in the start menu, so you may just use the default setting.



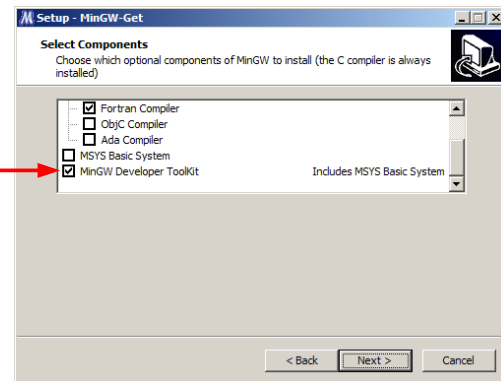
Select at least the following from the list of optional components:

C Compiler, C++ Compiler, Fortran Compiler, MinGW Developer ToolKit

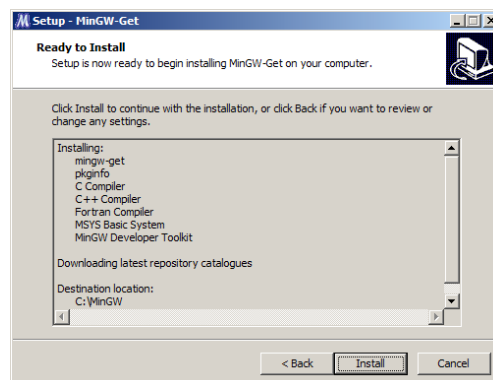
(Actually, the compilers are only necessary if you intend to create a 32 bit build, but it would not hurt to have them.)



and



Concluding, you should see an overview of your installation settings similar to that one:



After clicking “Install” some command line windows pop up in which the file downloads are processed. Depending on the speed of your internet connection this may take a few minutes.

## Step 2: MinGW-w64 installation

For 32 bit proceed with step 3.

The general remarks about previous installed versions given for MinGW apply to MinGW-w64 likewise.

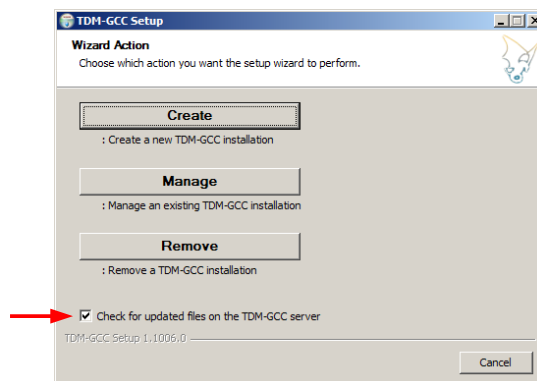
Download the latest TDM64-GCC bundle installer from:

<http://tdm-gcc.tdragon.net/download>

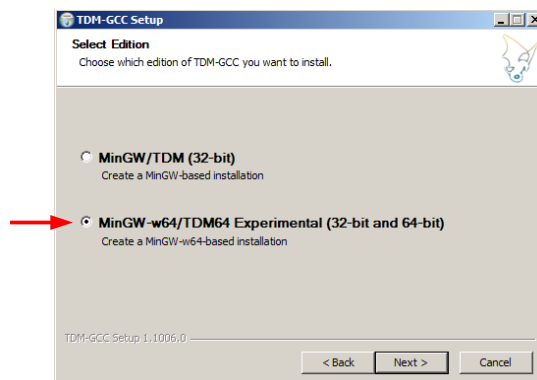
(for example: tdm64-gcc-4.7.1-3.exe)

Run the installer.

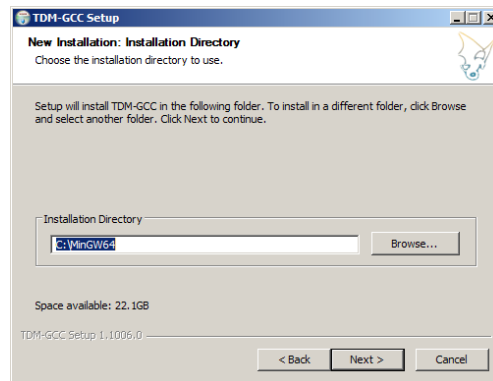
Make sure the option “Check for updated files on the TDM-GCC server” is ticked, then click “Create”.



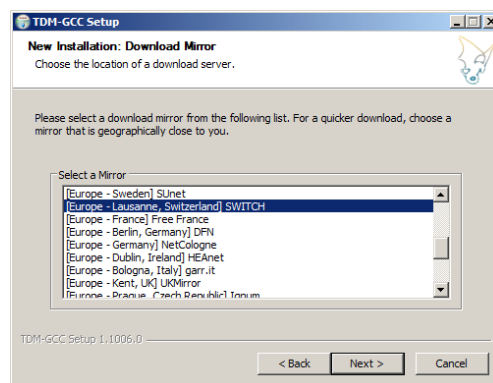
We want to create a MinGW-w64-based installation:



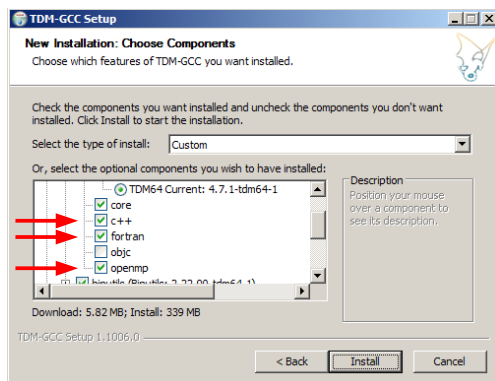
As the installation directory choose one that is different to that of MinGW. It is important that the path you choose does not contain any space or special characters. In the scope of this document it is assumed that TDM-GCC is installed to C:\MinGW64. If your installation location differs from this you have to adapt all paths stated below accordingly.



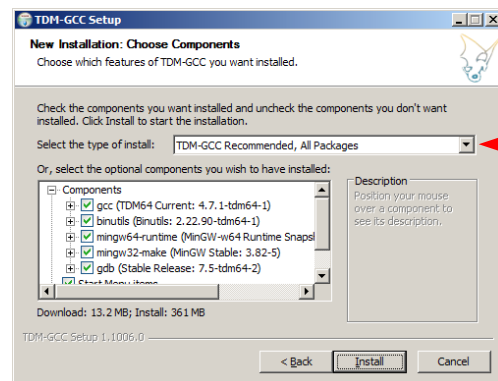
Select an appropriate download mirror from the given list:



Finally, you have to choose the components that should be installed. Either select c++, fortran and openmp manually from the gcc category or simply install all packages.



OR



### Step 3: SVN installation

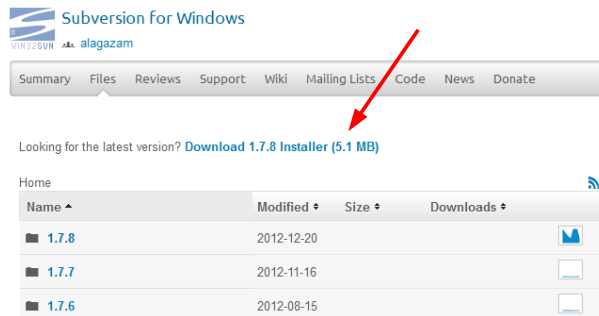
If you have SVN already installed, you can use your installation, of course.

Otherwise, I recommend to install the latest Subversion for Windows. You can get the installer from:

<http://sourceforge.net/projects/win32svn/files/>

(for example: Setup-Subversion-1.7.8.msi)

There should be a hyperlink at the top of the page:



### Step 4: Set up MSYS

Open a MinGW Shell (either using the shortcut in the start menu or run `C:\MinGW\msys\1.0\msys.bat`, but do not use the MinGW Command Prompt from MinGW64). You can insert text from the clipboard by pressing the insert key.

Execute the following command:

```
mingw-get install msys-wget msys-unzip
```

### Step 5: Tell MSYS to use MinGW64

For 32 bit continue with step 6.

Open `C:\MinGW\msys\1.0\etc\fstab` for editing, modify its content to

```
C:\MinGW64\ /mingw
```

and restart the MinGW Shell.

You can type `gcc -v` to check whether you are using the correct compiler. The output should be similar to:

```
MINGW32~
$ gcc -v
Using built-in specs.
COLLECT_GCC=C:\MinGW64\bin\gcc.exe
COLLECT_LTO_WRAPPER=c:/mingw64/bin/./libexec/gcc/x86_64-w64-mingw32/4.7.1/lto-w
rapper.exe
target: x86_64-w64-mingw32
Configured with: ./../src/gcc-4.7.1/configure --build=x86_64-w64-mingw32 --enab
le-targets=all --enable-languages=c,c++,fortran,objc,obj-c++ --enable-libgomp --
enable-lto --enable-libstdc++-debug --enable-version-specific-runtime-libs --ena
ble-fully-dynamic-string --with-gnu-ld --disable-verbose --disable-nls --disabl
e-win32-registry --prefix=/mingw64tdm --with-local-prefix=/mingw64tdm --with-pkgv
ersion=tdm64-1 --with-bugurl=http://tdm-gcc.tdragon.net/bugs
Thread model: win32
gcc version 4.7.1 (tdm64-1)
```

It is important that you are using the tdm64 version of gcc.

## Step 6: Download Ipopt

When this document was written the most recent Ipopt release was 3.10.3. Some important modifications for 64 bit support are not included in that version, thus one should use the trunk revision until a new version is released. You can check this by looking at <https://projects.coin-or.org/svn/Ipopt/releases/>.

If there is a new stable release use (replace x.y.z by the version number, e.g. 3.11.0):

```
svn co https://projects.coin-or.org/svn/Ipopt/releases/x.y.z CoinIpopt
```

Otherwise execute the following command:

```
svn co https://projects.coin-or.org/svn/Ipopt/trunk CoinIpopt
```

It may take a few minutes until everything has been downloaded.

Change into the root directory of the Ipopt distribution:

```
cd CoinIpopt
```

## Step 7: Get external components

The next steps perform downloads which may take a few minutes.

Execute the following command:

```
cd ThirdParty/Blas; ./get.Blas; cd ../Lapack; ./get.Lapack; cd ../../
```

```
MINGW32:~/CoinIpopt
$ cd ThirdParty/Blas; ./get.Blas; cd ../Lapack; ./get.Lapack; cd ../../

Running script for downloading the source code for BLAS

Downloading the source code from www.netlib.org...
--2013-03-13 09:27:50-- ftp://www.netlib.org/blas/blas.tgz
=> 'blas.tgz'
Resolving www.netlib.org... 160.36.131.121
Connecting to www.netlib.org:160.36.131.121:21... connected.
Logging in as anonymous ... Logged in!
=> SYST ... done.      => PWD ... done.
=> TYPE I ... done.    => CWD (1) /blas ... done.
=> SIZE blas.tgz ... 99927
=> PASV ... done.      => RETR blas.tgz ... done.
Length: 99927 (98K) (unauthoritative)
100%[=====] 99,927      37.7K/s   in 2.6s
2013-03-13 09:27:55 (37.7 KB/s) - 'blas.tgz' saved [99927]

Uncompressing the tarball...
Unpacking the source code...
Deleting the tar file...
Moving the source files from BLAS subdirectory
Done downloading the source code for BLAS.

Running script for downloading the source code for LAPACK

Downloading the source code from www.netlib.org...
--2013-03-13 09:27:56-- ftp://www.netlib.org/lapack/lapack-3.4.2.tgz
=> 'lapack-3.4.2.tgz'
Resolving www.netlib.org... 160.36.131.121
Connecting to www.netlib.org:160.36.131.121:21... connected.
Logging in as anonymous ... Logged in!
=> SYST ... done.      => PWD ... done.
=> TYPE I ... done.    => CWD (1) /lapack ... done.
=> SIZE lapack-3.4.2.tgz ... 6168281
=> PASV ... done.      => RETR lapack-3.4.2.tgz ... done.
Length: 6168281 (5.9M) (unauthoritative)
100%[=====] 6,168,281    46.5K/s   in 2m 10s
2013-03-13 09:30:08 (46.4 KB/s) - 'lapack-3.4.2.tgz' saved [6168281]

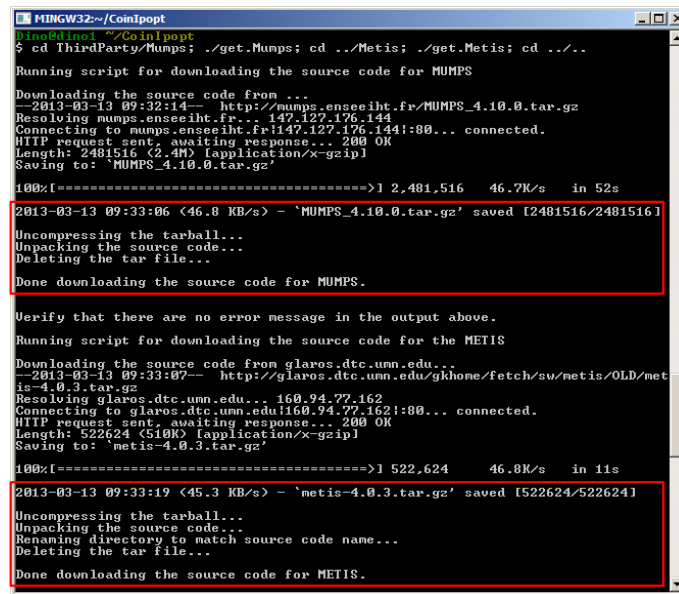
Uncompressing the tarball...
Unpacking the source code...
Deleting the tar file...
Done downloading the source code for LAPACK.
```

If you want to use Ipopt from AMPL type (not necessary for the MATLAB interface):

```
cd ThirdParty/ASL; ./get.ASL; cd ../../
```

Proceed with obtaining MUMPS and METIS (or get another solver):

```
cd ThirdParty/Mumps; ./get.Mumps; cd ../Metis; ./get.Metis; cd ../..
```



```
MINGW32~/CoinIpopt
Dino@Dino1 ~/CoinIpopt
$ cd ThirdParty/Mumps; ./get.Mumps; cd ../Metis; ./get.Metis; cd ../..

Running script for downloading the source code for MUMPS

Downloading the source code from ...
--2013-03-13 09:32:14-- http://mumps.enseeiht.fr/MUMPS_4.10.0.tar.gz
Resolving mumps.enseeiht.fr... 147.127.176.144
Connecting to mumps.enseeiht.fr:147.127.176.144:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2481516 (2.4M) [application/x-gzip]
Saving to: 'MUMPS_4.10.0.tar.gz'

100%[=====] 2,481,516 46.7K/s in 52s

2013-03-13 09:33:06 <46.8 KB/s> - 'MUMPS_4.10.0.tar.gz' saved [2481516/2481516]

Uncompressing the tarball...
Unpacking the source code...
Deleting the tar file...

Done downloading the source code for MUMPS.

Verify that there are no error message in the output above.

Running script for downloading the source code for the METIS

Downloading the source code from glaros.dtc.umn.edu...
--2013-03-13 09:33:07-- http://glaros.dtc.umn.edu/gkhome/fetch/sw/metis/OLD/metis-4.0.3.tar.gz
Resolving glaros.dtc.umn.edu... 160.94.77.162
Connecting to glaros.dtc.umn.edu:160.94.77.162:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 522624 (510K) [application/x-gzip]
Saving to: 'metis-4.0.3.tar.gz'

100%[=====] 522,624 46.8K/s in 11s

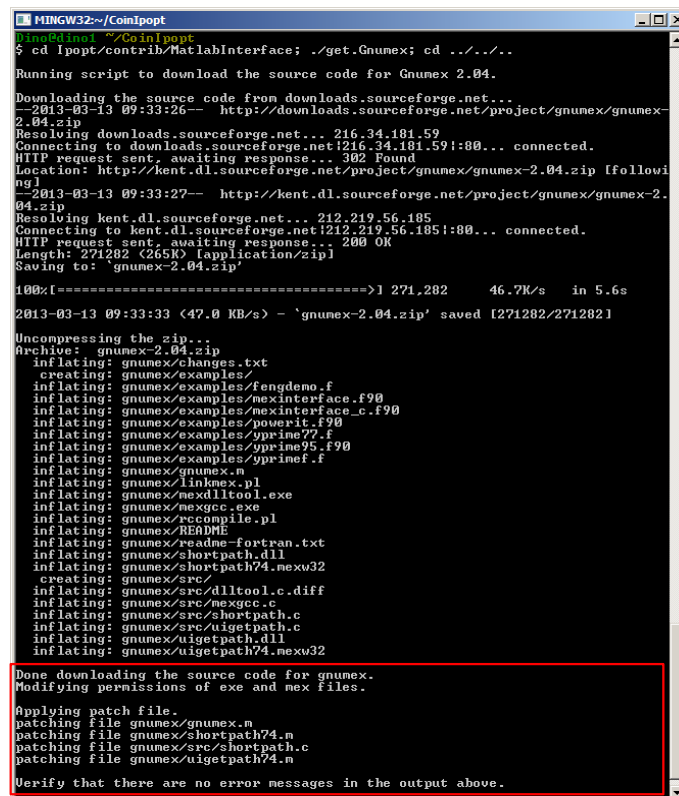
2013-03-13 09:33:19 <45.3 KB/s> - 'metis-4.0.3.tar.gz' saved [522624/522624]

Uncompressing the tarball...
Unpacking the source code...
Renaming directory to match source code name...
Deleting the tar file...

Done downloading the source code for METIS.
```

Finally, get Gnumex:

```
cd Ipopt/contrib/MatlabInterface; ./get.Gnumex; cd ../../..
```



```
MINGW32~/CoinIpopt
Dino@Dino1 ~/CoinIpopt
$ cd Ipopt/contrib/MatlabInterface; ./get.Gnumex; cd ../../..

Running script to download the source code for Gnumex 2.04.

Downloading the source code from downloads.sourceforge.net...
--2013-03-13 09:33:26-- http://downloads.sourceforge.net/project/gnunex/gnunex-2.04.zip
Resolving downloads.sourceforge.net... 216.34.181.59
Connecting to downloads.sourceforge.net:216.34.181.59:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: http://kent.dl.sourceforge.net/project/gnunex/gnunex-2.04.zip [following]
--2013-03-13 09:33:27-- http://kent.dl.sourceforge.net/project/gnunex/gnunex-2.04.zip
Resolving kent.dl.sourceforge.net... 212.219.56.185
Connecting to kent.dl.sourceforge.net:212.219.56.185:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 271282 (265K) [application/zip]
Saving to: 'gnunex-2.04.zip'

100%[=====] 271,282 46.7K/s in 5.6s

2013-03-13 09:33:33 <47.0 KB/s> - 'gnunex-2.04.zip' saved [271282/271282]

Uncompressing the zip...
Archive: gnunex-2.04.zip
  inflating: gnunex/changes.txt
    creating: gnunex/examples/
  inflating: gnunex/examples/fengdemo.f
  inflating: gnunex/examples/mexinterface.f90
  inflating: gnunex/examples/mexinterface.c.f90
  inflating: gnunex/examples/powerit.f90
  inflating: gnunex/examples/yprime77.f
  inflating: gnunex/examples/yprime95.f90
  inflating: gnunex/examples/yprime.f
  inflating: gnunex/gnunex.m
  inflating: gnunex/linkmex.pl
  inflating: gnunex/mexdlltool.exe
  inflating: gnunex/mexgcc.exe
  inflating: gnunex/rccompile.pl
  inflating: gnunex/README
  inflating: gnunex/readme-fortran.txt
  inflating: gnunex/shortpath.dll
  inflating: gnunex/shortpath74.mexu32
    creating: gnunex/src/
  inflating: gnunex/src/dlltool.c.diff
  inflating: gnunex/src/mexgcc.c
  inflating: gnunex/src/shortpath.c
  inflating: gnunex/src/uigetpath.c
  inflating: gnunex/uigetpath.dll
  inflating: gnunex/uigetpath74.mexu32

Done downloading the source code for gnunex.
Modifying permissions of exe and mex files.

Applying patch file.
patching file gnunex/gnunex.m
patching file gnunex/shortpath74.m
patching file gnunex/src/shortpath.c
patching file gnunex/uigetpath74.m

Verify that there are no error messages in the output above.
```

## Step 8: Get MATLAB's short path name

We need MATLAB's short path name. I assume MATLAB is installed at `C:\Program Files\MATLAB\R2012b`.

Open a Windows command line, for example by pressing Windows key + R and open `cmd` or use the shortcut in the start menu. Change to the drive's root by typing:

```
cd \
```

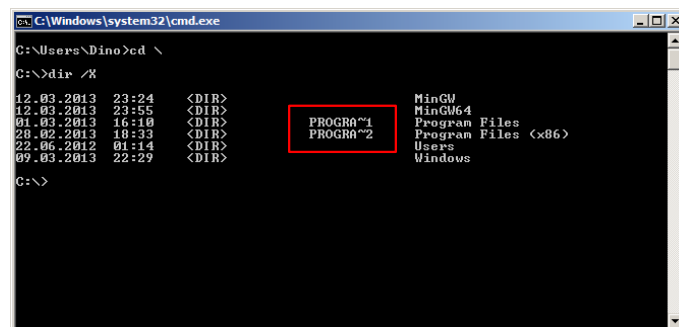
You can change to another drive by entering its letter followed by a colon, e.g.:

```
E:
```

Enter

```
dir /X
```

to list all directories with their short names:



So, the wanted short path name is `C:\PROGRA~1\MATLAB\R2012b`. Depending on the location where you have MATLAB installed you have to adapt this method accordingly. In case MATLAB is located in a sub-folder that contains space characters as well, e.g. `C:\Program Files (x86)\The MathWorks\MATLAB\R2012b`, you have to get the short path name for each of them. You can navigate into a directory by entering:

```
cd "Program Files (x86)"
```

For the last example you should end up with something like `C:\PROGRA~2\THEMAT~1\MATLAB\R2012b`.

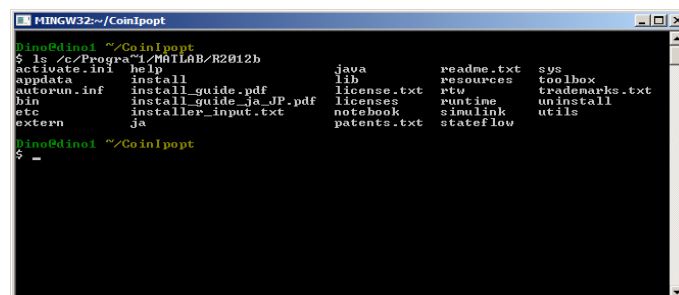
Now close the Windows command line and convert the path to UNIX style:

```
/c/Progra~1/MATLAB/R2012b
```

If you caught the right path, the output of

```
ls /c/Progra~1/MATLAB/R2012b
```

in the MinGW Shell should look like this:





### Step 9: Build Ipopt

Execute the following commands (replace the MATLAB path with yours):

```
mkdir build
cd build
../configure --with-matlab-home=/c/Progra~1/MATLAB/R2012b ADD_FFLAGS="-static-libgcc" CDEFS="-DWITHOUT_PTHREAD=1"
```

```
MINGW32~/CoinIpopt/build
DinoPdino1 ~/CoinIpopt
$ mkdir build
DinoPdino1 ~/CoinIpopt
$ cd build
DinoPdino1 ~/CoinIpopt/build
$ ./Configure --with-matlab-home=/c/Program~1/MATLAB/R2012b ADD_FFLAGS="-static
-libgcc" CDEFS="-DWITHOUT_PTHREAD=1"
```

The configuration takes a couple of minutes and should end up with:

```
configure: Main configuration of Ipopt successful
```

```
MINGW32~/CoinIopt/build
config.status: creating Iopt.pc
config.status: creating Iopt-uninstalled.pc
config.status: creating Ioptamplinterface.pc
config.status: creating Ioptamplinterface-uninstalled.pc
config.status: creating src/Common/config.h
config.status: creating src/Common/config-Iopt.h
config.status: creating examples/ScalableProblems/config.h
config.status: linking ../../Iopt/examples/hs071_cpp/hs071_main.cpp to test/hs071_nlp.cpp
config.status: linking ../../Iopt/examples/hs071_cpp/hs071_nlp.cpp to test/hs071_nlp.cpp
config.status: linking ../../Iopt/examples/hs071_cpp/hs071_nlp_h.cpp to test/hs071_nlp_h.cpp
config.status: linking ../../Iopt/examples/hs071_c/hs071_c.c to test/hs071_c.c
config.status: executing depfiles commands
configure: Creating UPRN links for data files
configure: In case of trouble, first consult the troubleshooting page at https://projects.coin-or.org/BuildTools/wiki/user-troubleshooting
configure: Configuration of Iopt successful
configure: In case of trouble, first consult the troubleshooting page at https://projects.coin-or.org/BuildTools/wiki/user-troubleshooting
configure: Main configuration of Iopt successful

Rime@edine1 ~/CoinIopt/build$
```

Now enter

make

and go for a coffee. The last output should be:

```
make[1]: Leaving directory '/home/.../CoinIpopt/build'
```

[illegible]

```
make test
```

At the end, it should say that the tests have been passed (without ASL the AMPL test cannot be performed):

```

MINGW32~/CoinIpopt/build
$ ./libgcc/x86_64-w64-mingw32/4.7.1/../../../../x86_64-w64-mingw32/bin/./lib
cc-x86_64-w64-mingw32/4.7.1/../../../../x86_64-w64-mingw32/lib-Lc-mingw64/bin/./lib
./libgcc/x86_64-w64-mingw32/4.7.1/../../../../home/Dino/CoinIpopt/build/ThirdPart
Metis/lib/liblbcnetmetis.a/home/Dino/CoinIpopt/build/ThirdPart/Lapack/lib
ThirdPart/blas/liblbcnetblas.a/home/Dino/CoinIpopt/build/ThirdPart/blas/liblbcnetblas.a
-lgfortran -lmingw32 -loldname -lmingwex -lmsvcrt -lquadmath -ladvapi32 -lshell
32 -luser32 -lkernel32 -lstdc++
chmod u+x ./run_unitTests
./run_unitTests

Running unitTests...

Testing AMPL Solver Executable...
Test passed!

Testing C++ Example...
Test passed!

Testing C Example...
Test passed!

Testing Fortran Example...
Test passed!

make[12]: Leaving directory `./home/Dino/CoinIpopt/build/Ipopt/test'
make[11]: Leaving directory `./home/Dino/CoinIpopt/build/Ipopt'

Dino@Dino1: ~/CoinIpopt/build
$

```

Finally, type

```
make install
```

```
MINGW32~/CoinIpopt/build
```

```
/opt/for addlibs_c.txt ; \
    -L"/mingw64/bin/" -libgcc/x86_64-mingw32/4.7.1 -L"/mingw64/bin/" -libgcc/x86_64-mingw32/4.7.1 -L"/mingw64/bin/" -libgcc/x86_64-mingw32/4.7.1 -L"/mingw64/bin/" -libgcc/x86_64-mingw32/4.7.1 -L"/mingw64/bin/" -libgcc/x86_64-mingw32/4.7.1 -L"/mingw64/bin/" -libgcc/x86_64-mingw32/4.7.1 -L"/mingw64/bin/" -lfortran -lmwing32 -ldlmaind -lmwex -lmvcvt -lquadmath -lm-ladeapi32 -lshe132 -lsuer32 -lkernel32 coin_dummy; do \
    addlibs='echo -n $addlibs | sed -e "s$ ?$ ! ?g";' \
done ; \
echo '$addlibs -lstdc++ -lm' /home/Dino/CoinIpopt/build/share/coin/doc
/popt/ipopt_addlibs_f.txt
make[4]: Leaving directory '/home/Dino/CoinIpopt/build/popt'
make[3]: Leaving directory '/home/Dino/CoinIpopt/build/popt'
make[2]: Leaving directory '/home/Dino/CoinIpopt/build/popt'
make[1]: Leaving directory '/home/Dino/CoinIpopt/build/popt'
make[2]: Entering directory '/home/Dino/CoinIpopt/build'
make[2]: Entering directory '/home/Dino/CoinIpopt/build'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/home/Dino/CoinIpopt/build'
make[1]: Leaving directory '/home/Dino/CoinIpopt/build'
```

```
Dino@dino1 ~/CoinIpopt/build
```

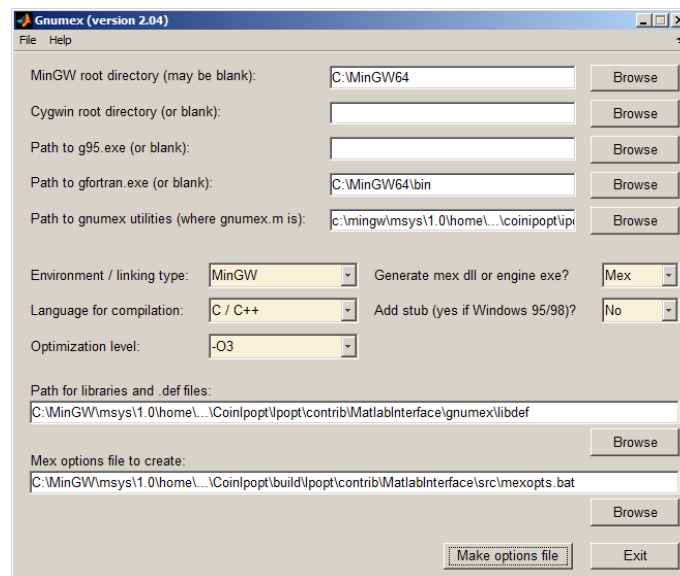
### Step 8: Build the MATLAB interface

You should be in the `~/CoinIpopt/build` directory. Type:

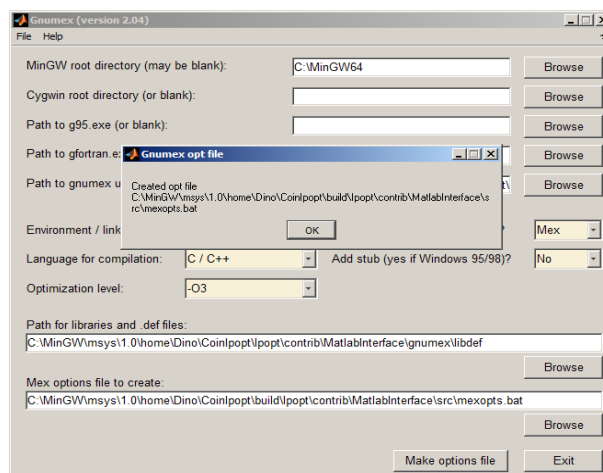
```
cd Ipopt/contrib/MatlabInterface/src/  
make gnumex
```

```
# MITIGW32~/CoinIptop/build/Iptop/contrib/MatlabInterface/src  
$ cd Iptop/contrib/MatlabInterface/src/  
$ make gnumex  
if test -d .. \&& ! test -f ../Iptop/contrib/MatlabInterface/src/..gnumex ; then  
    echo "Warning: no gnumex folder found. Run \"cd `dirname $0`\" &&  
    ./Iptop/contrib/MatlabInterface/src/..get.Gnumex\" first."  
else  
    $GNUM_COMMANDS="oldpwd-pwd; cd ../../../../../../Iptop/contrib/MatlabInterfa  
ce/src/..gnumex; gnumex<'startup'; \  
    gnumexopts.gnumex<'defaults'; gnumexopts.precompath=[pwd '<liblef']"  
;  
    gnumexopts.optfile=oldpwd '$mexopts.bat' ]; \  
case uname in  
MINGW*)  
echo Use gnumex in Matlab to create mexopts.bat file, then close t  
his new instance of Matlab : "  
c:\PROGRAM\1\MATLAB\R2012a\bin/matlab\" --wait -r \"$GNUM_COMMANDS  
> gnumexopts.mingwpath=fileparts(gnumexopts.gfortpath); gnumex<'st  
ruct2fig','gnumexopts'" \  
;;  
CYGMIN*)  
echo Use gnumex in Matlab to create mexopts.bat file, then close t  
his new instance of Matlab : "  
c:\PROGRA~1\MATLAB\R2012a\bin/matlab\" --wait -r \"$GNUM_COMMANDS gn  
umexopts.environ=3; gnumex<'struct2fig','gnumexopts'" \  
;;  
esac  
fi  
Use gnumex in Matlab to create mexopts.bat file, then close this new instance of  
Matlab.  
$ cd ~/CoinIptop/build/Iptop/contrib/MatlabInterface/src
```

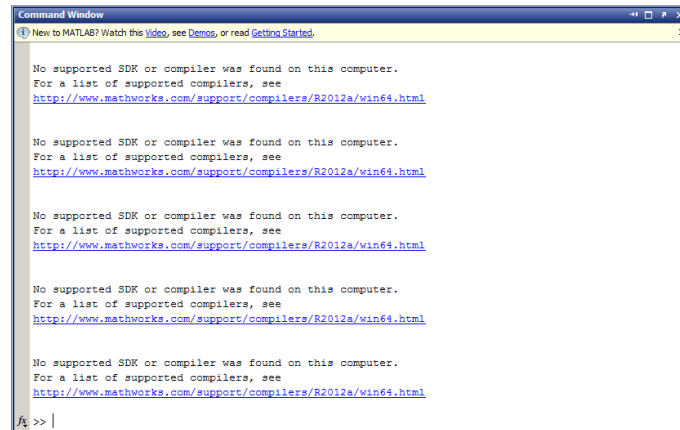
Wait until MATLAB is ready and the Gnumex window is shown:



Click “Make options file” and wait until it says that the file has been created:



During that process MATLAB may show some warnings/errors but anyway, we can build the mex file:



Click "Exit" and close MATLAB.

Type

make mexopts

and open C:\MinGW\msys\1.0\home\...\CoinIpot\build\Ipot\contrib\MatlabInterface\src\mexopts.bat for editing. Add -static to line 38:

set GM\_ADD\_LIBS=-static -llibmx -llibmex -llibmat -LC:/MinGW/...

```
1 @echo off
2 rem C:\MinGW\msys\1.0\home\Dino\CoinIpot\build\Ipot\contrib\MatlabInterface\src\mexopts.bat
3 rem Generated by gnumex.m script in c:\mingw\msys\1.0\home\dino\COINIP-1\ipopt\contrib\MATLAB-1\gnumex
4 rem gnumex version: 2.04
5 rem Compile and link options used for building MEX etc files with
6 rem the Mingw/Cygwin tools. Options here are:
7 rem Gnumex, version 2.04
8 rem MinGW linking
9 rem Mex (*.dll) creation
10 rem Libraries regenerated now
11 rem Language: C / C++
12 rem Optimization level: -O3 (full optimization)
13 rem Matlab version 7.14
14 rem
15 set MATLAB=c:\PROGRA-1\MATLAB\R2012A-1
16 set GM_PERLPATH=c:\PROGRA-1\MATLAB\R2012A-1\sys\perl\win32\bin\perl.exe
17 set GM_UTIL_PATH=c:\mingw\msys\1.0\home\dino\COINIP-1\ipopt\contrib\MATLAB-1\gnumex
18 set PATH=C:\MinGW64\bin;%PATH%
19 set PATH=%PATH%;C:\Cygwin\usr\local\gfortran\libexec\gcc\i686-pc-cygwin\4.3.0
20 set LIBRARY_PATH=C:\MinGW64\lib
21 set G95_LIBRARY_PATH=C:\MinGW64\lib
22 rem
23 rem precompiled library directory and library files
24 set GM_QLIB_NAME=C:\MinGW\msys\1.0\home\Dino\COINIP-1\Ipot\contrib\MATLAB-1\gnumex\libdef
25 rem
26 rem directory for .def-files
27 set GM_DEF_PATH=C:\MinGW\msys\1.0\home\Dino\COINIP-1\Ipot\contrib\MATLAB-1\gnumex\libdef
28 rem
29 rem Type of file to compile (mex or engine)
30 set GM_MEXTYPE=mex
31 rem
32 rem Language for compilation
33 set GM_MEXLANG=cxx
34 rem
35 rem File for exporting mexFunction symbol
36 set GM_MEXDEF=C:\MinGW\msys\1.0\home\Dino\COINIP-1\Ipot\contrib\MATLAB-1\gnumex\libdef\mex.def
37 rem
38 set GM_ADD_LIBS=-static -llibmx -llibmex -llibmat -LC:/MinGW/msys/1.0/home/Dino/CoinIpot/build/lib -lipopt
-LC:/MinGW/msys/1.0/home/Dino/CoinIpot/build/lib -lcoinmumps -lpthread
-Lc:/mingw64/bin/./lib/gcc/x86_64-w64-mingw32/4.7.1 -Lc:/mingw64/bin/./lib/gcc
-Lc:/mingw64/bin/./lib/gcc/x86_64-w64-mingw32/4.7.1/./../x86_64-w64-mingw32/lib/./lib
```

To finish, enter

make install

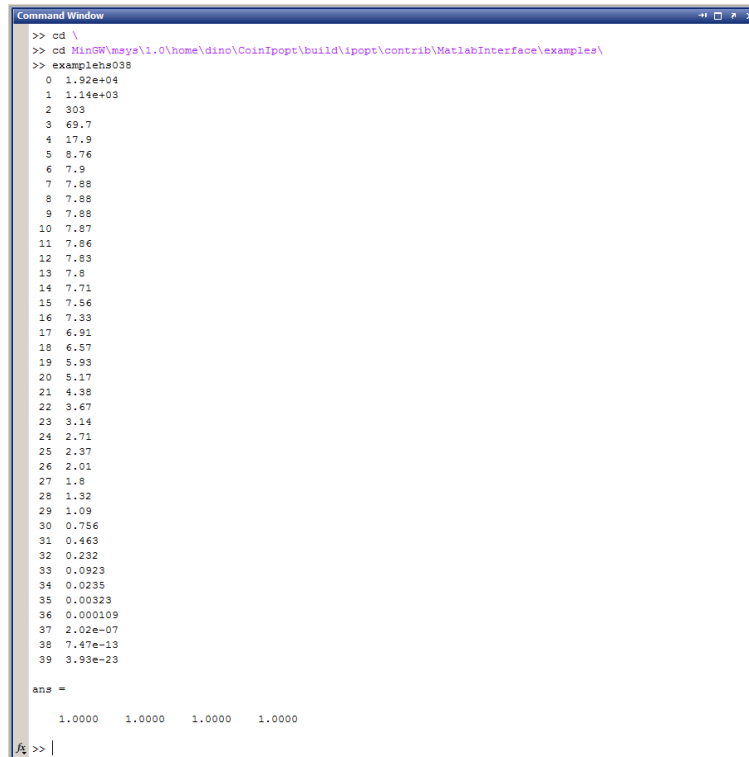
Now you should have a file named ipopt.mexw32 or ipopt.mexw64, respectively, in the src folder.

## Step 9: Test the MATLAB interface

Copy the mex file from the `src` directory into the `...\MatlabInterface\examples` folder:

```
cp ipopt.mexw64 ../examples/
```

Open a new MATLAB instance, change into that path and run one of the included m-files:



```
Command Window
>> cd \
>> cd MinGW\sys\1.0\home\dino\CoinIpopt\build\ipopt\contrib\MatlabInterface\examples\
>> exampleha038
0 1.92e+04
1 1.14e+03
2 303
3 69.7
4 17.9
5 8.76
6 7.9
7 7.88
8 7.88
9 7.88
10 7.87
11 7.86
12 7.83
13 7.8
14 7.71
15 7.56
16 7.33
17 6.91
18 6.57
19 5.93
20 5.17
21 4.38
22 3.67
23 3.14
24 2.71
25 2.37
26 2.01
27 1.8
28 1.32
29 1.09
30 0.756
31 0.463
32 0.232
33 0.0923
34 0.0235
35 0.00323
36 0.000109
37 2.02e-07
38 7.47e-13
39 3.93e-23

ans =
    1.0000    1.0000    1.0000    1.0000

>> |
```

## Sources

<http://www.coin-or.org/Ipopt/documentation/>  
<https://projects.coin-or.org/Ipopt/wiki/MatlabInterface>  
<http://list.coin-or.org/pipermail/ipopt/2013-March/003276.html>