Note: the compilation and library combination has been tested only as a 32 bit application (even though working on a 64 bit machine).

Prerequisites:

1. Visual studio 2010 (The project description files provided are for VS 2010, for other VS versions you might have to manually set up the dependencies). Note: The QT add in does not exist for VS express. You will have to compile the project manually when using the VS Express studio (not described here)
2. The boost libraries:
    a. Get them at http://sourceforge.net/projects/boost/files/boost/1.53.0/
    b. and unzip them somewhere you will find them again. No building is required.
3. QT and the qt add in for visual studio 2010: http://qt-project.org/downloads
    a. Choose qt4.something (e.g.4.84) download for VS2010. QT 5 is not recommended for this project. There where changes between qt4/5 which would make it necessary to modify the source files/set additional macros to compile the project with qt5.
    b. Download tha QT 4 VS addin (In the "other" section) http://qt-project.org/downloads#qt-other
    c. To test that VS is set up correctly with QT create a new visual studio project of the type "QT Application" and build it. An empty window should show.

Building DEC 2013

1. Download the whole project at https://github.com/bertholet/DEC_2013
2. Open the DEC.sln file (visual studio 2010 recommended)
3. set "Application" as your start up project:
    a. Rightclick and select "set as start up project"
4. Adding the boost folder to the additional include libraries of the DECCore subproject:
    a. Rightclick on DECore->Properties-> Confiuration Properties->C/C++->General -> Additional Include Directories
    b. Select Configuration: All Configurations
    c. Add the path to the boost folder containing the folders boost, doc, lib etc
    d. Try to build the DECCore subproject by rightclicking it, and clicking build, DECCore should

compile.

5. Getting/Compiling sparse solver libraries, two wrappers are implemented

    a. Either get libparadiso library, http://www.pardiso-project.org/

    Don't forget that Pardiso also needs a license file, which has to be located in your home directory; you should have received a license key when downloading pardiso; create an empty file pardiso.lic in your home folder and paste the license key in there.

    Go to the SolverConfig.h file in the Solver subproject and modify it such that the PardisoSolver Wrapper will be used.

    b. or compile the Suitesparse solver suite to library files, which takes considerably longer and is described in the seperate ReadMe ....

6. The Solver Subproject should compile now.

7. put the solver specific lib, dll files in the Debug and Release folder which are generated by VS and are located in your project directory; typically this will be

    a. for Pardiso: libpardiso412-WIN-X86.dll,libpardiso412-WIN-X86.lib

    b. SuiteSparse: AMD.lib,CAMD.lib,CCOLAMD.lib,CHOLMOD.lib,COLAMD.lib,libblas.dll,libblas.lib,libgcc_s_dw2-1.dll,libgfortran-3.dll ,liblapack.dll, liblapack.lib, liblapacke.dll, liblapacke.lib, libmetis.lib,SuiteSparse_config.lib,UMFPACK.lib

8. Go to the "Application" Project property page (Right click applications->Properties)

    a. If you are using only the PardisoSolver: Configuration properties->Linker->Input: Remove all referenced Suitesparse libraries

    b. If you are using only suiteSparse: remove the pardiso library

    c. Navigate to configuration properties -> Debugging and add PATH=../$(Configuration);$(PATH) to the Environmnet. If the PATH variable is already specified, just add ../$(Configuration); to the PATH variable. Make sure you have "for all configurations" selected

9. click run. If libiomp5md.dll is missing, you might have to get the intel/fortran c++ libraries, they are needed by the pardiso Solver. Get them by using the installer and restart your machine http://software.intel.com/en-us/articles/redistributable-libraries-of-the-intel-c-and-fortran-compiler-for-windows