



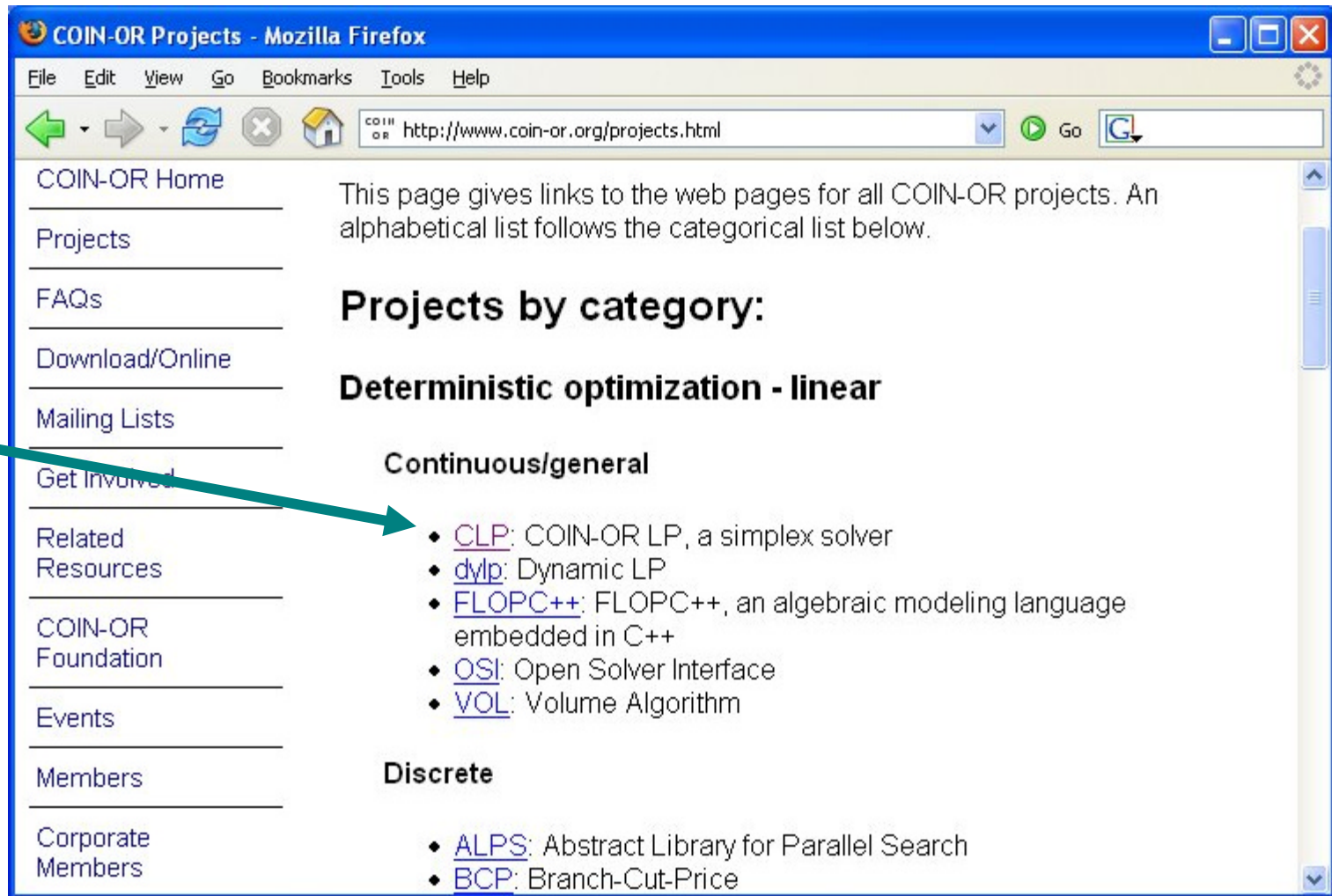
Web Site (www.coin-or.org)

Getting around the Web Site to
Build and Use COIN-OR

www.coin-or.org Home Page



Projects Page



COIN-OR Projects - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://www.coin-or.org/projects.html

COIN-OR Home

Projects

FAQs

Download/Online

Mailing Lists

Get Involved

Related Resources

COIN-OR Foundation

Events

Members

Corporate Members

This page gives links to the web pages for all COIN-OR projects. An alphabetical list follows the categorical list below.

Projects by category:

Deterministic optimization - linear

Continuous/general

- [CLP](#): COIN-OR LP, a simplex solver
- [dylp](#): Dynamic LP
- [FLOPC++](#): FLOPC++, an algebraic modeling language embedded in C++
- [OSI](#): Open Solver Interface
- [VOL](#): Volume Algorithm

Discrete

- [ALPS](#): Abstract Library for Parallel Search
- [BCP](#): Branch-Cut-Price

Clp Trac Page

COIN-OR Linear Programming Solver - Trac - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://projects.coin-or.org/Clp

trac
Integrated SCM & Project Management

[Login](#) [Settings](#) [Help/Guide](#) [About Trac](#) [Register](#)

[Search](#)

[Wiki](#) [Timeline](#) [Roadmap](#) [Browse Source](#) [View Tickets](#) [Search](#)

[Start Page](#) [Title Index](#) [Recent Changes](#) [Page History](#)

Welcome to the Clp home page

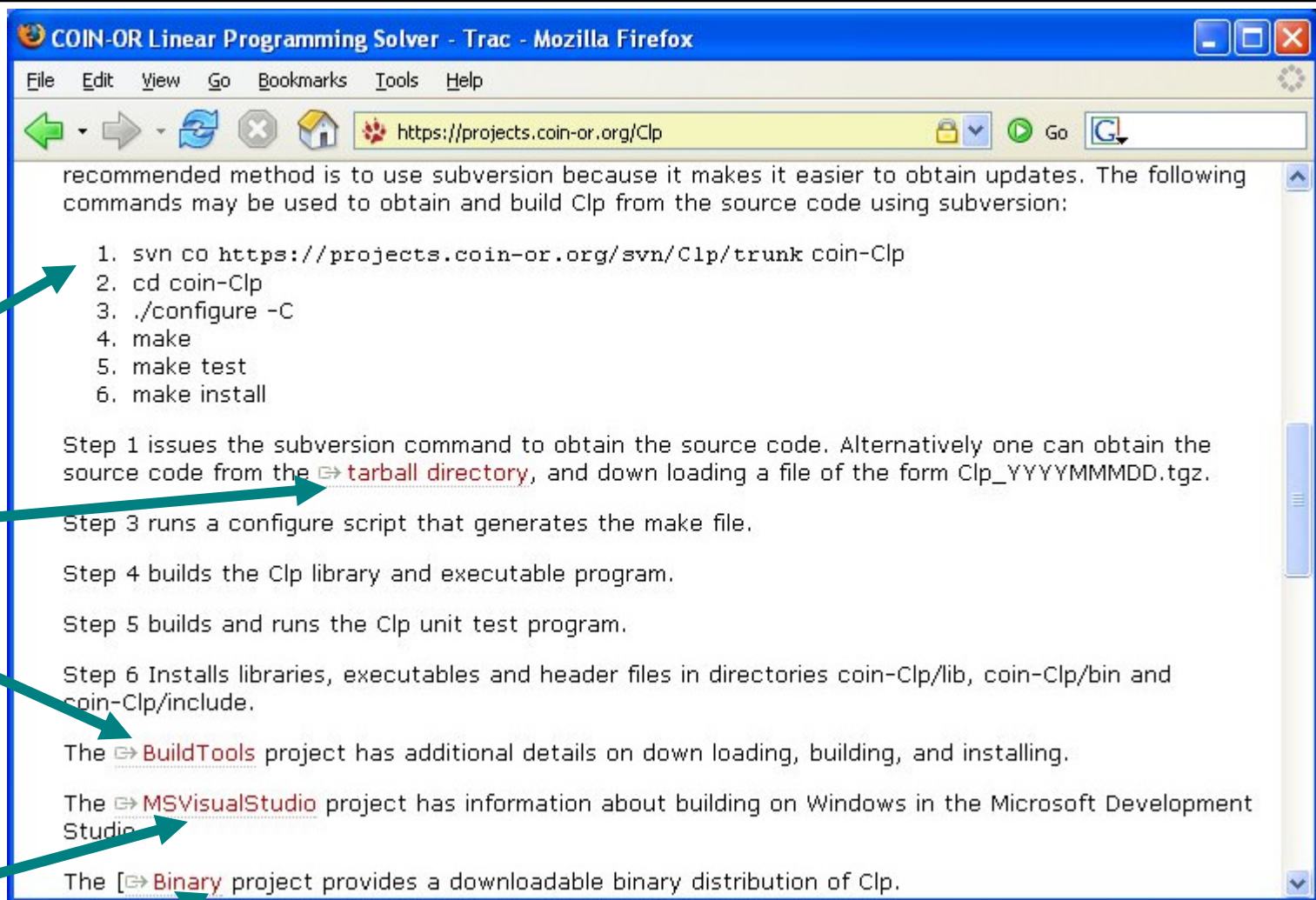
Note that these project webpages are based on Wiki, which allows webusers to modify the content to correct typos, add information, or share their experience and tips with other users. You are welcome to contribute to these project webpages. To edit these pages or submit a ticket you must first [register and login](#).

Introduction

Clp (**C**oin-or **l**inear **p**rogramming) is an open-source linear programming solver written in C++. It is primarily meant to be used as a callable library, but a basic, stand-alone executable version is also available.. It is designed to find solutions of mathematical optimization problems of the form

$$\begin{aligned} \min & c^t x \\ \text{such that:} & \\ & \text{rowlower_bound} \leq A x \leq \text{rowupper_bound} \\ & \text{columnlower_bound} \leq x \leq \text{columnupper_bound} \end{aligned}$$

Clp Build Instructions on Trac Page



recommended method is to use subversion because it makes it easier to obtain updates. The following commands may be used to obtain and build Clp from the source code using subversion:

1. `svn co https://projects.coin-or.org/svn/Clp/trunk coin-Clp`
2. `cd coin-Clp`
3. `./configure -C`
4. `make`
5. `make test`
6. `make install`

Step 1 issues the subversion command to obtain the source code. Alternatively one can obtain the source code from the [tarball directory](#), and downloading a file of the form `Clp_YYYYMMDD.tgz`.

Step 3 runs a configure script that generates the make file.

Step 4 builds the Clp library and executable program.

Step 5 builds and runs the Clp unit test program.

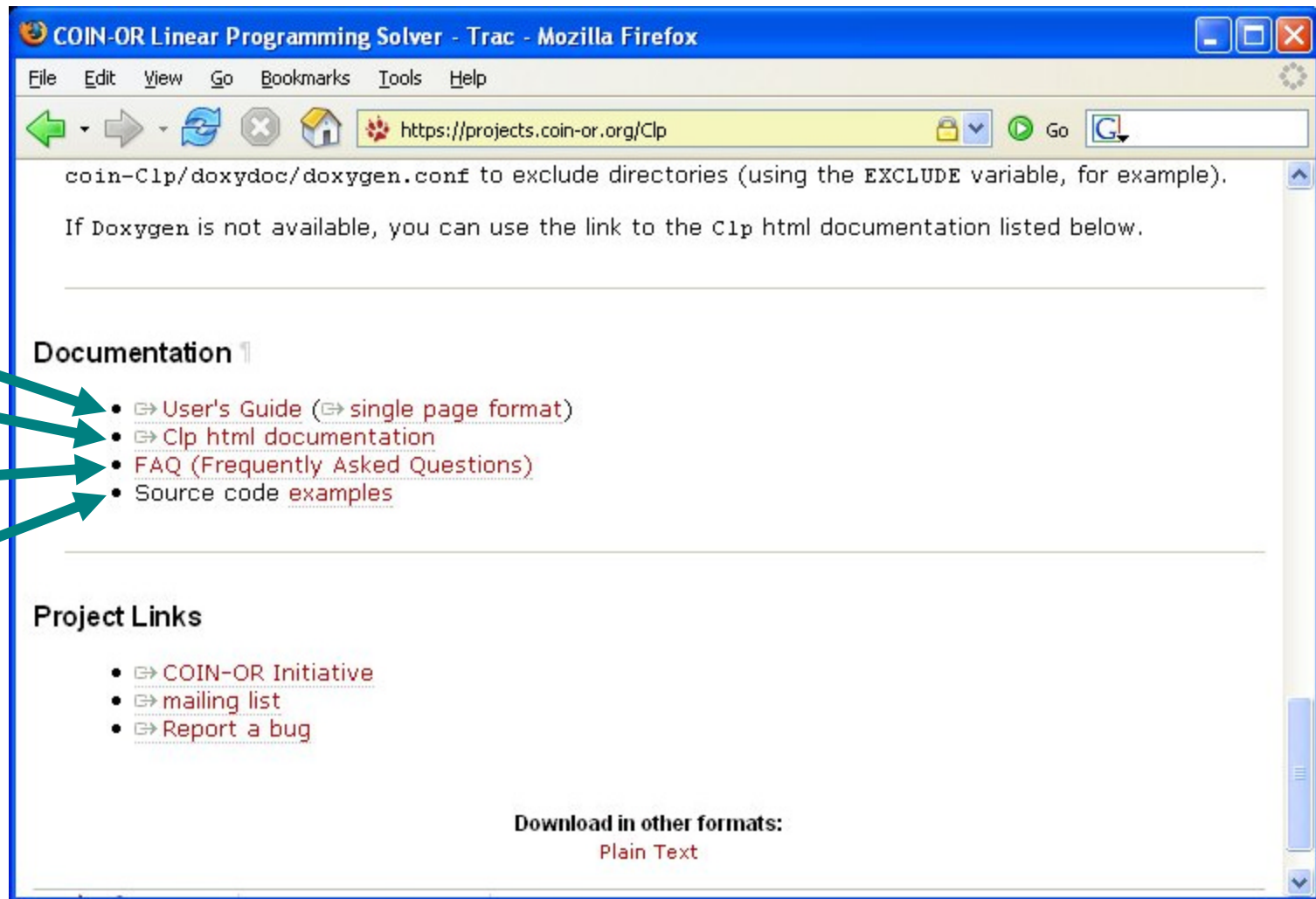
Step 6 installs libraries, executables and header files in directories `coin-Clp/lib`, `coin-Clp/bin` and `coin-Clp/include`.

The [BuildTools](#) project has additional details on downloading, building, and installing.

The [MSVisualStudio](#) project has information about building on Windows in the Microsoft Development Studio.

The [Binary](#) project provides a downloadable binary distribution of Clp.

Clp Documentation on Trac Page





Build Clp

Unix

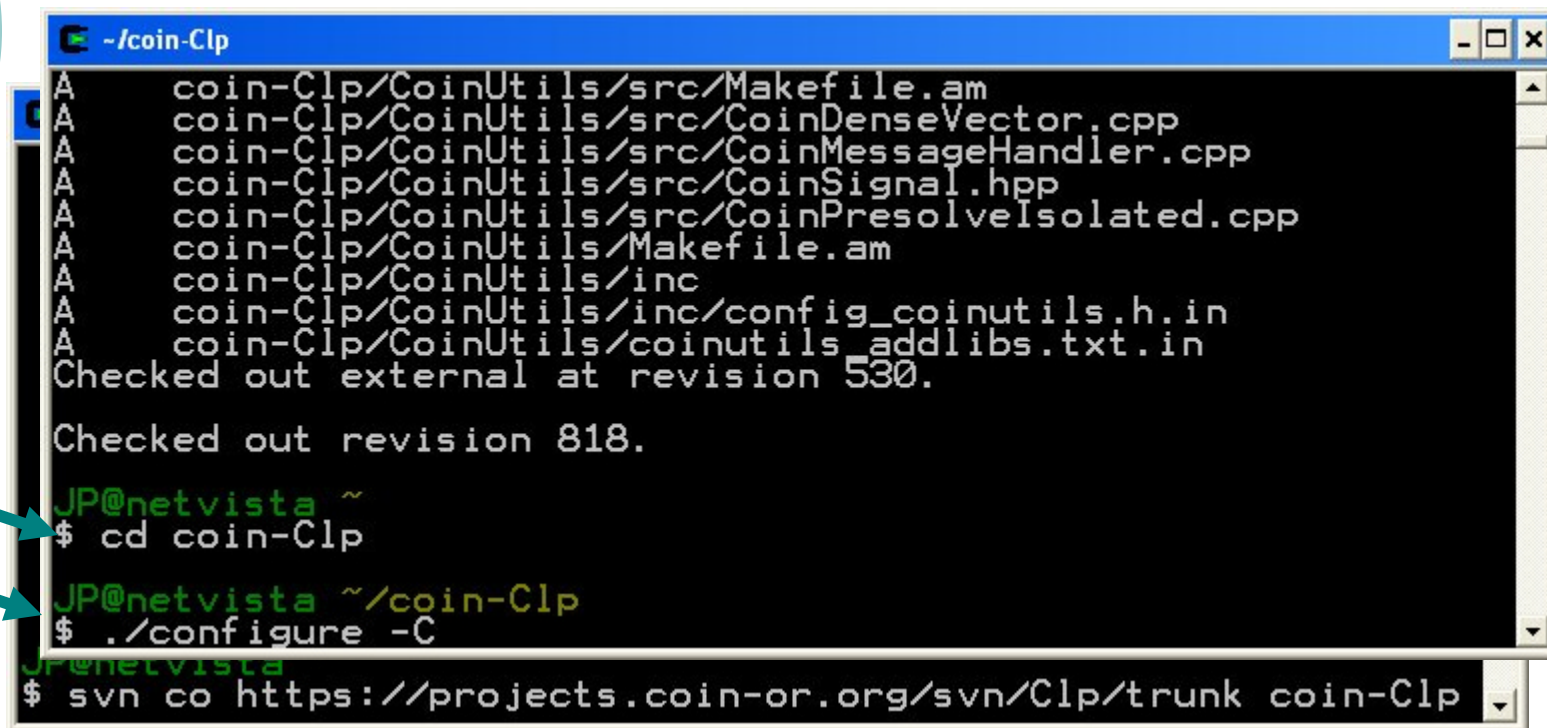
Build Clp on Unix



A terminal window with a blue title bar and standard window controls. The terminal text shows a user prompt and a command to checkout the Clp project from a Subversion repository. A teal arrow points to the command line.

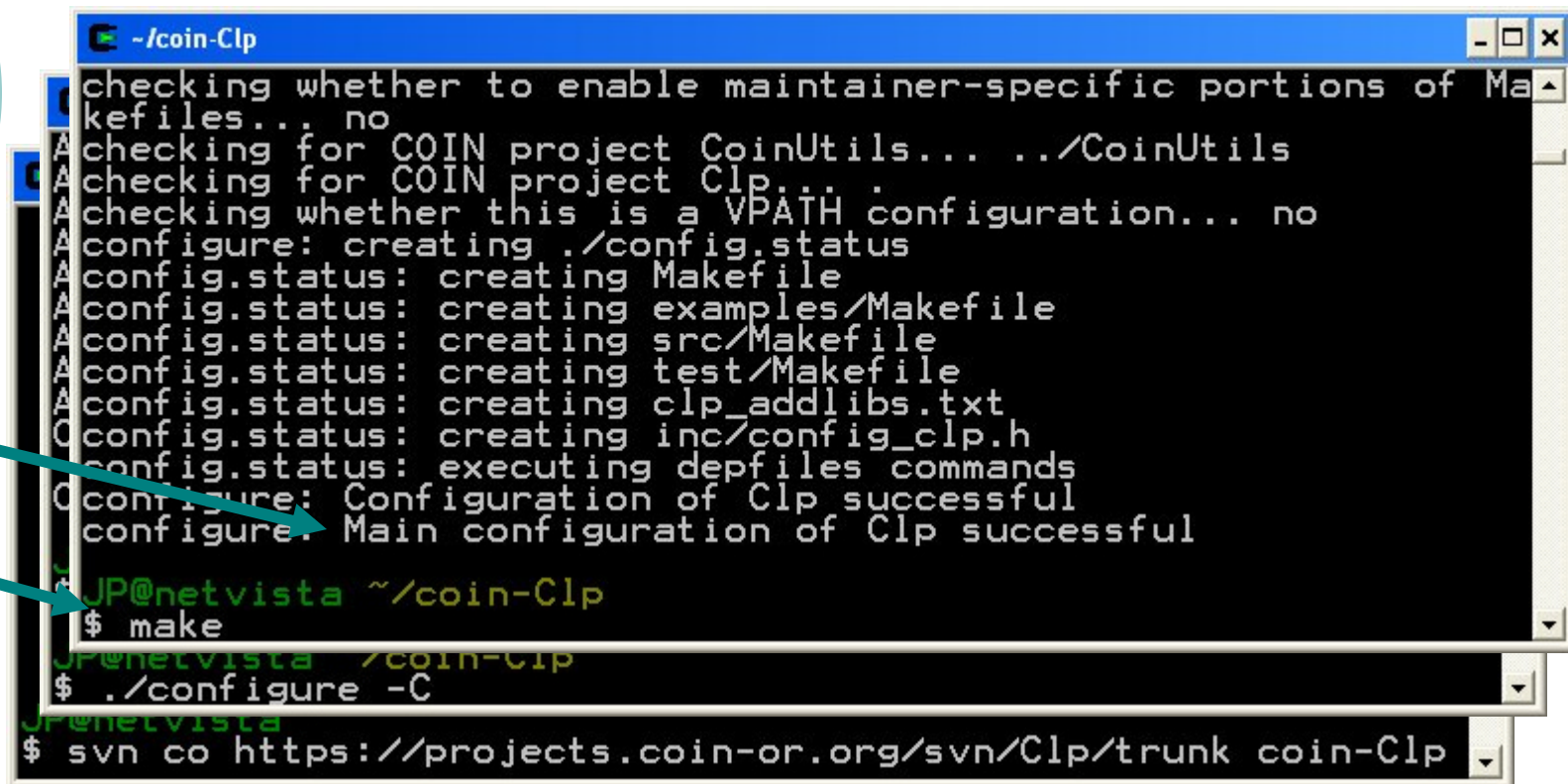
```
JP@netvista ~  
$ svn co https://projects.coin-or.org/svn/Clp/trunk coin-Clp
```

Build Clp on Unix



```
~/coin-Clp
A coin-Clp/CoinUtils/src/Makefile.am
A coin-Clp/CoinUtils/src/CoinDenseVector.cpp
A coin-Clp/CoinUtils/src/CoinMessageHandler.cpp
A coin-Clp/CoinUtils/src/CoinSignal.hpp
A coin-Clp/CoinUtils/src/CoinPresolveIsolated.cpp
A coin-Clp/CoinUtils/Makefile.am
A coin-Clp/CoinUtils/inc
A coin-Clp/CoinUtils/inc/config_coinutils.h.in
A coin-Clp/CoinUtils/coinutils_addlibs.txt.in
Checked out external at revision 530.
Checked out revision 818.
JP@netvista ~
$ cd coin-Clp
JP@netvista ~/coin-Clp
$ ./configure -C
JP@netvista ~
$ svn co https://projects.coin-or.org/svn/Clp/trunk coin-Clp
```


Build Clp on Unix



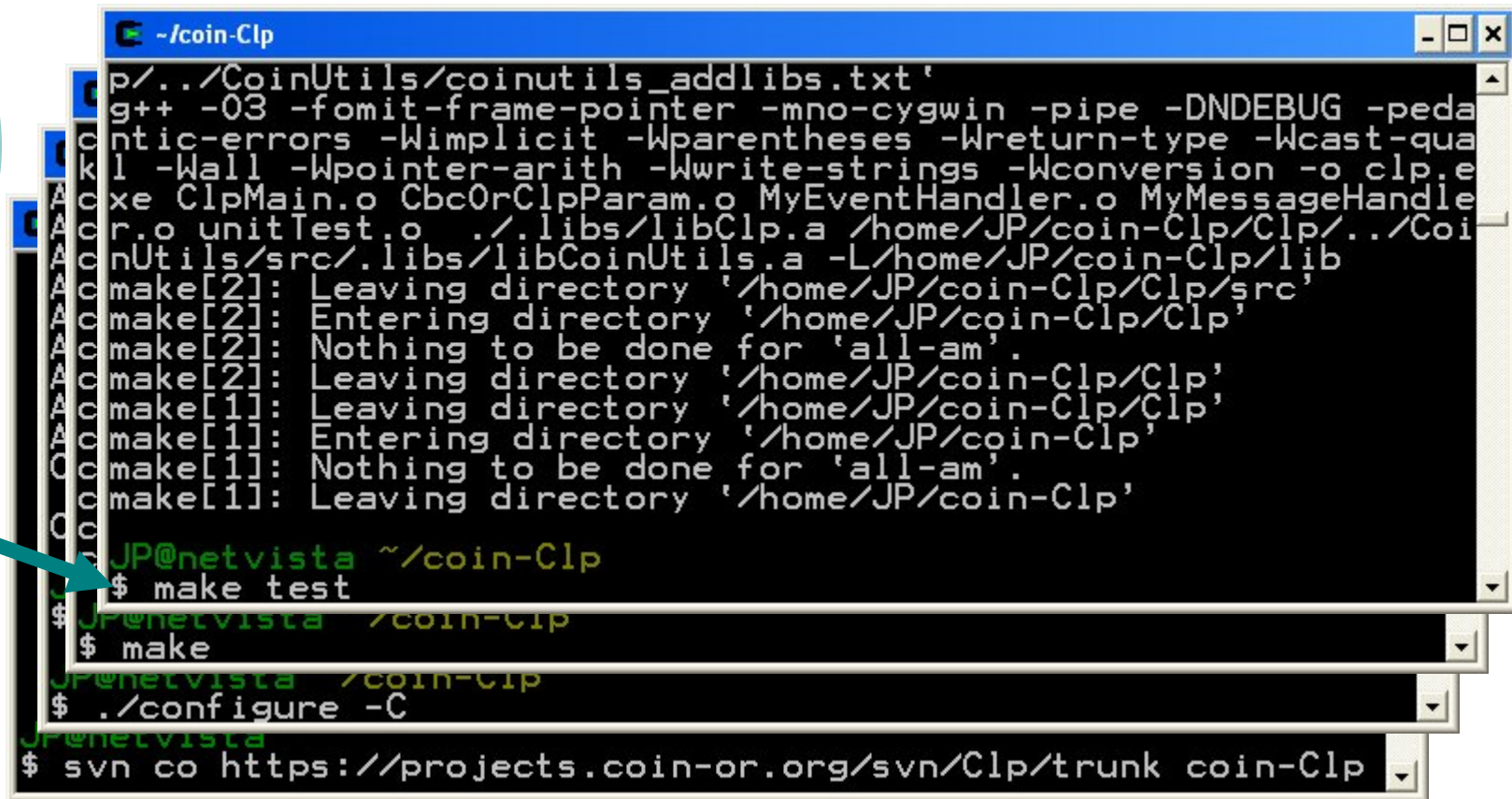
```
~/coin-Clp
checking whether to enable maintainer-specific portions of Makefiles... no
checking for COIN project CoinUtils... ../CoinUtils
checking for COIN project Clp... 
checking whether this is a VPATH configuration... no
configure: creating ./config.status
config.status: creating Makefile
config.status: creating examples/Makefile
config.status: creating src/Makefile
config.status: creating test/Makefile
config.status: creating clp_addlibs.txt
config.status: creating inc/config_clp.h
config.status: executing depfiles commands
configuration: Configuration of Clp successful
configuration: Main configuration of Clp successful

JP@netvista ~/coin-Clp
$ make

JP@netvista ~/coin-Clp
$ ./configure -C

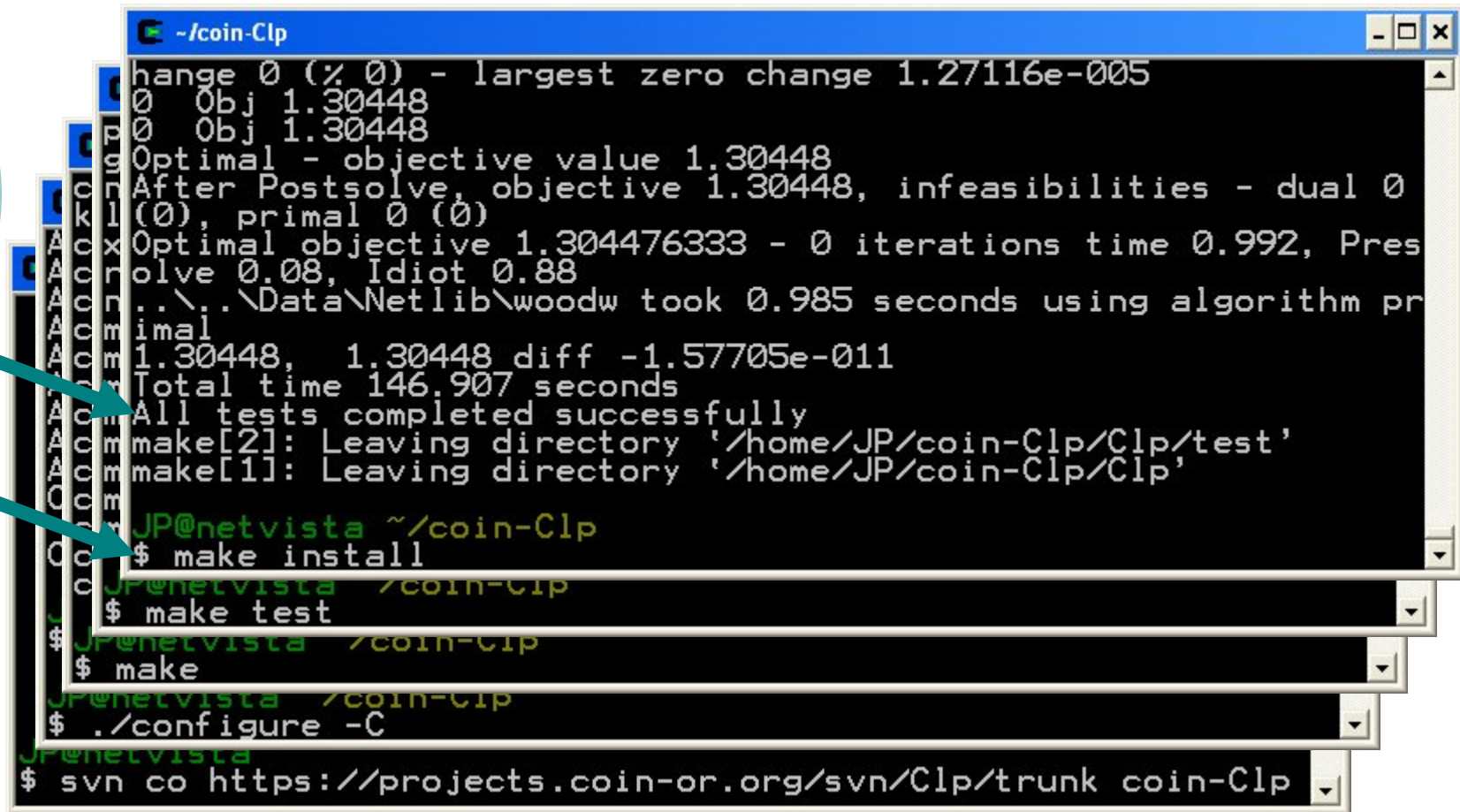
JP@netvista
$ svn co https://projects.coin-or.org/svn/Clp/trunk coin-Clp
```

Build Clp on Unix



```
~/coin-Clp
p/../../CoinUtils/coinutils_addlibs.txt'
g++ -O3 -fomit-frame-pointer -mno-cygwin -pipe -DNDEBUG -peda
ntic-errors -Wimplicit -Wparentheses -Wreturn-type -Wcast-qua
kl -Wall -Wpointer-arith -Wwrite-strings -Wconversion -o clp.e
xe ClpMain.o CbcOrClpParam.o MyEventHandler.o MyMessageHandle
r.o unitTest.o ../libs/libClp.a /home/JP/coin-Clp/Clp/../../Coi
nUtils/src/./libs/libCoinUtils.a -L/home/JP/coin-Clp/lib
make[2]: Leaving directory '/home/JP/coin-Clp/Clp/src'
make[2]: Entering directory '/home/JP/coin-Clp/Clp'
make[2]: Nothing to be done for 'all-am'.
make[2]: Leaving directory '/home/JP/coin-Clp/Clp'
make[1]: Leaving directory '/home/JP/coin-Clp/Clp'
make[1]: Entering directory '/home/JP/coin-Clp'
make[1]: Nothing to be done for 'all-am'.
make[1]: Leaving directory '/home/JP/coin-Clp'
JP@netvista ~/coin-Clp
$ make test
JP@netvista ~/coin-Clp
$ make
JP@netvista ~/coin-Clp
$ ./configure -C
JP@netvista
$ svn co https://projects.coin-or.org/svn/Clp/trunk coin-Clp
```

Build Clp on Unix



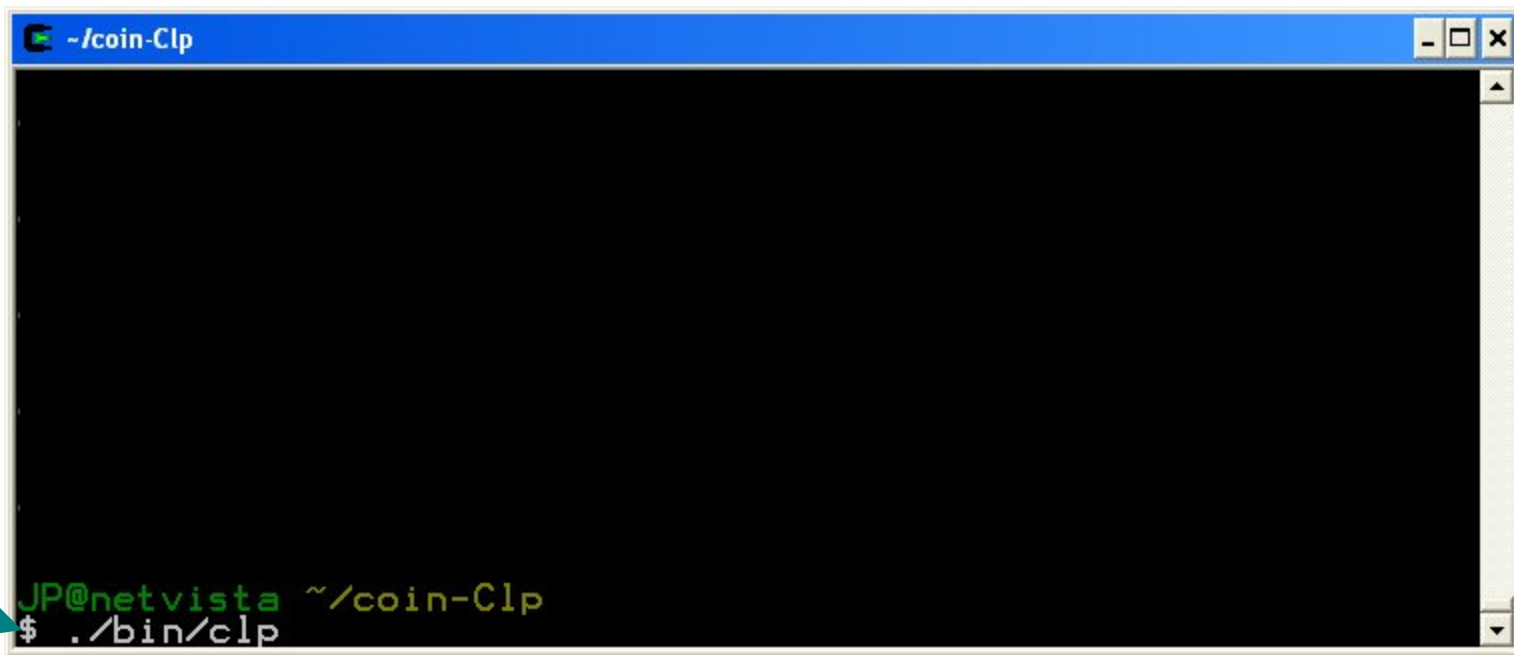
```
change 0 (% 0) - largest zero change 1.27116e-005
0 Obj 1.30448
0 Obj 1.30448
Optimal - objective value 1.30448
After Postsolve, objective 1.30448, infeasibilities - dual 0
(0), primal 0 (0)
Optimal objective 1.304476333 - 0 iterations time 0.992, Pres
olve 0.08, Idiot 0.88
...Data\Netlib\woodw took 0.985 seconds using algorithm pr
imal
1.30448, 1.30448 diff -1.57705e-011
Total time 146.907 seconds
All tests completed successfully
make[2]: Leaving directory '/home/JP/coin-Clp/Clp/test'
make[1]: Leaving directory '/home/JP/coin-Clp/Clp'
JP@netvista ~/coin-Clp
$ make install
JP@netvista ~/coin-Clp
$ make test
JP@netvista ~/coin-Clp
$ make
JP@netvista ~/coin-Clp
$ ./configure -C
JP@netvista
$ svn co https://projects.coin-or.org/svn/Clp/trunk coin-Clp
```



Run Clp executable

Experiment with Clp options

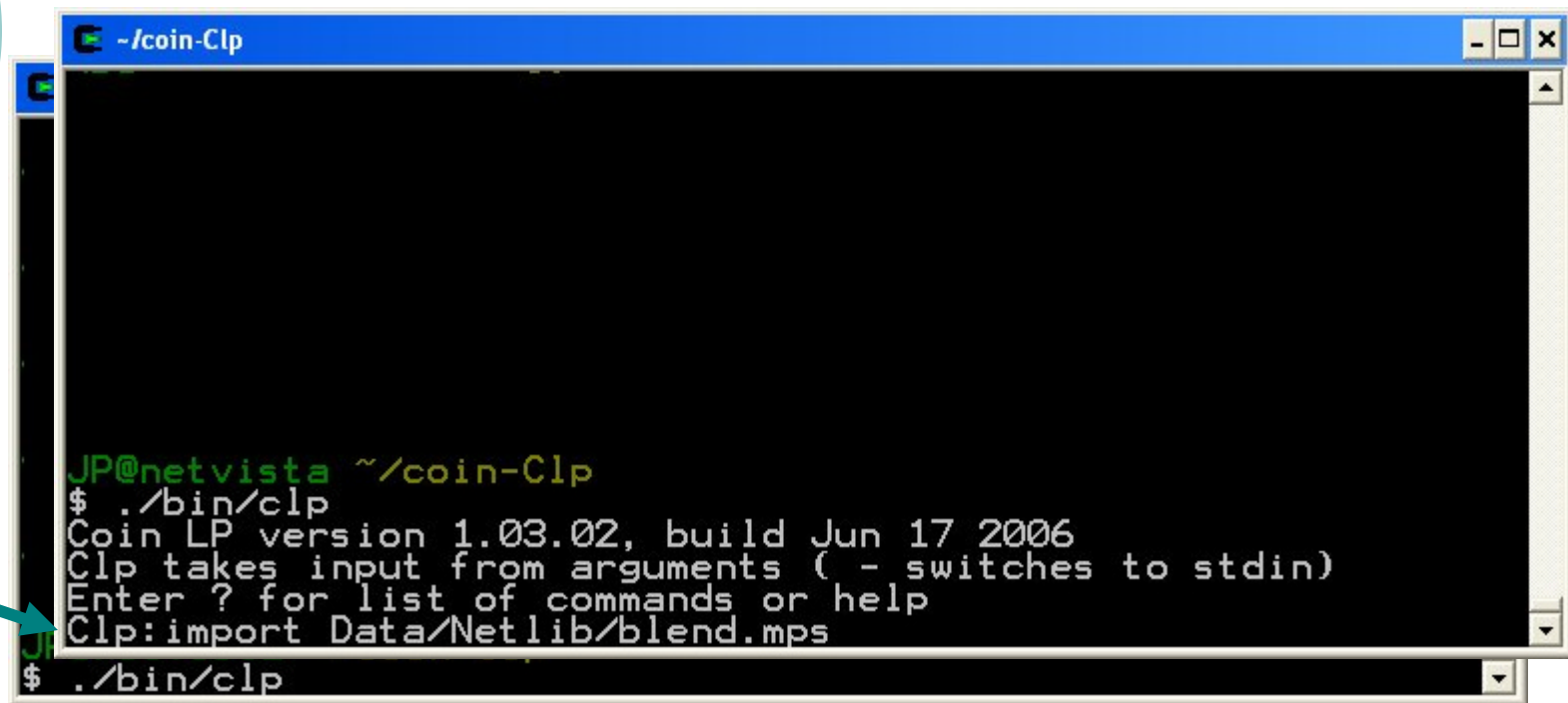
Experimenting with clp executable



A terminal window titled `~/coin-Clp` is shown. The window has a blue title bar and standard window controls. The terminal content is black with green text. At the bottom, the prompt `JP@netvista ~/coin-Clp` is followed by the command `$ ./bin/clp`. A teal arrow points to the command line.

```
JP@netvista ~/coin-Clp
$ ./bin/clp
```

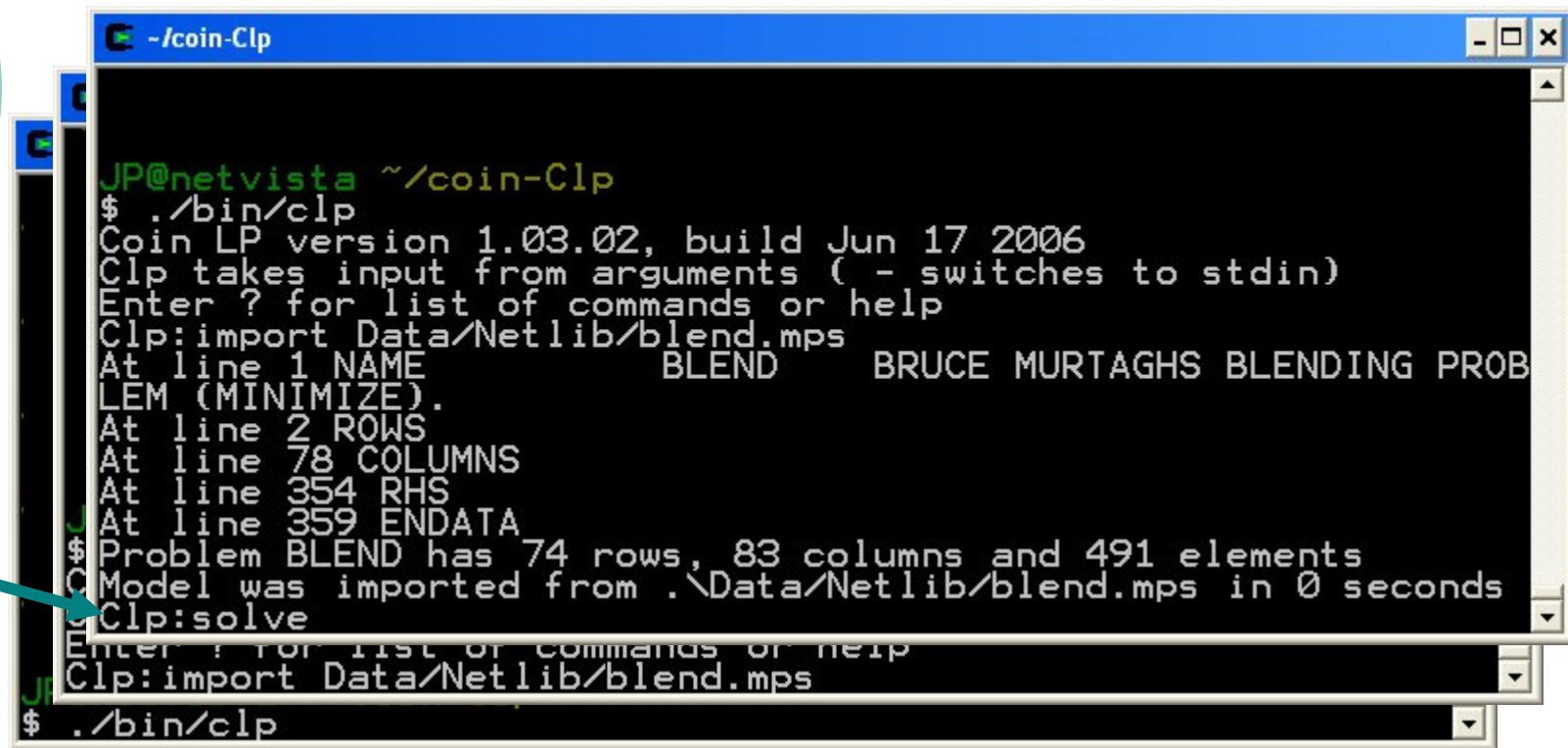
Experimenting with clp executable



A terminal window titled `~/coin-Clp` is shown. The prompt is `JP@netvista ~/coin-Clp`. The user enters `$ ./bin/clp`, which outputs: `Coin LP version 1.03.02, build Jun 17 2006`, `Clp takes input from arguments ( - switches to stdin)`, and `Enter ? for list of commands or help`. The user then enters `Clp:import Data/Netlib/blend.mps`. A teal arrow points to the prompt `JP` on the left side of the terminal window. Below the terminal window, the prompt `$ ./bin/clp` is visible.

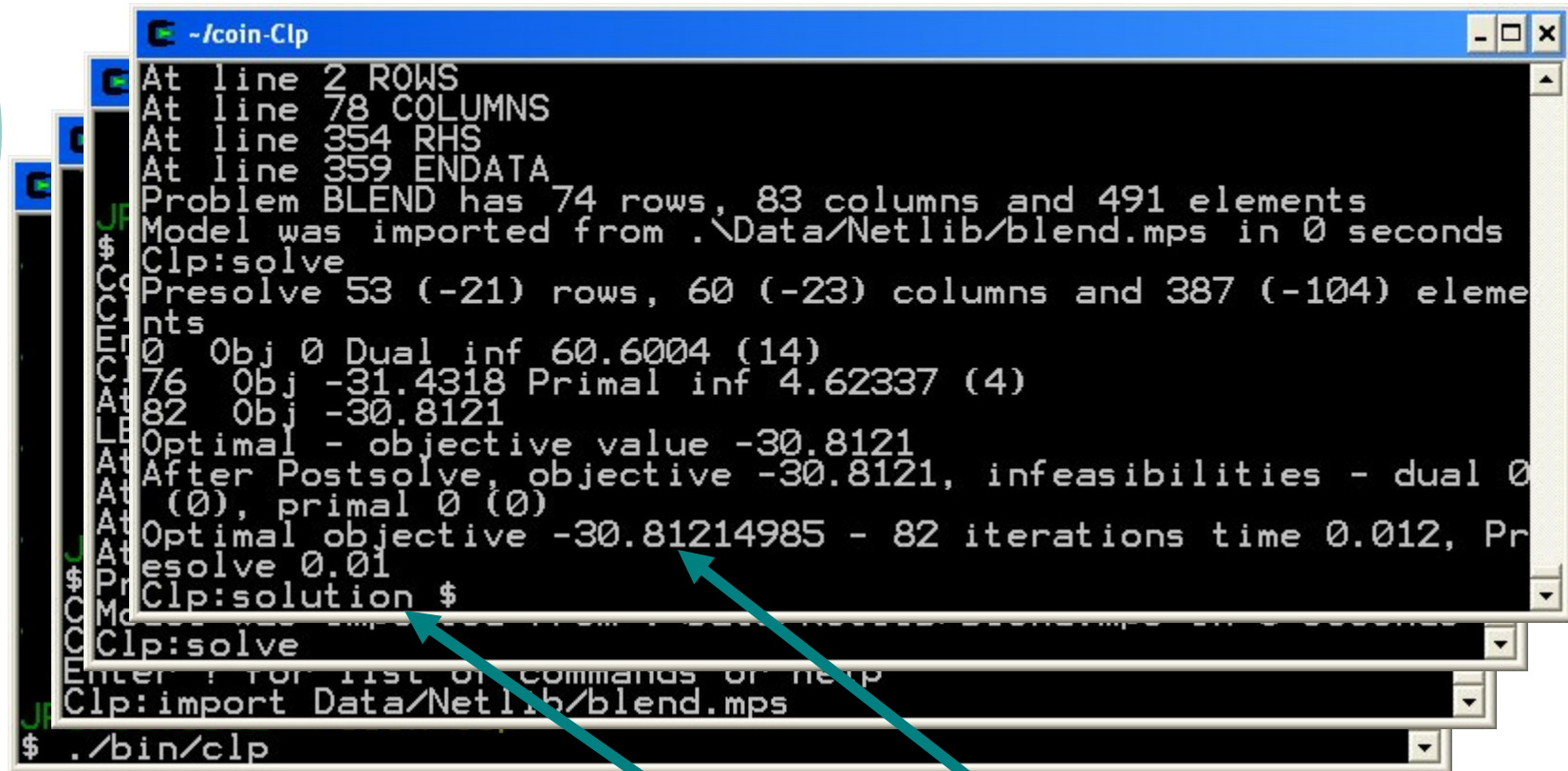
```
JP@netvista ~/coin-Clp
$ ./bin/clp
Coin LP version 1.03.02, build Jun 17 2006
Clp takes input from arguments ( - switches to stdin)
Enter ? for list of commands or help
Clp:import Data/Netlib/blend.mps
JP
$ ./bin/clp
```

Experimenting with clp executable



```
JP@netvista ~/coin-Clp
$ ./bin/clp
Coin LP version 1.03.02, build Jun 17 2006
Clp takes input from arguments ( - switches to stdin)
Enter ? for list of commands or help
Clp:import Data/Netlib/blend.mps
At line 1 NAME          BLEND      BRUCE MURTAGHS BLENDING PROB
LEM (MINIMIZE).
At line 2 ROWS
At line 78 COLUMNS
At line 354 RHS
At line 359 ENDATA
$ Problem BLEND has 74 rows, 83 columns and 491 elements
Clp:Model was imported from ./Data/Netlib/blend.mps in 0 seconds
Clp:solve
Enter ? for list of commands or help
Clp:import Data/Netlib/blend.mps
JP$ ./bin/clp
```


Experimenting with clp executable



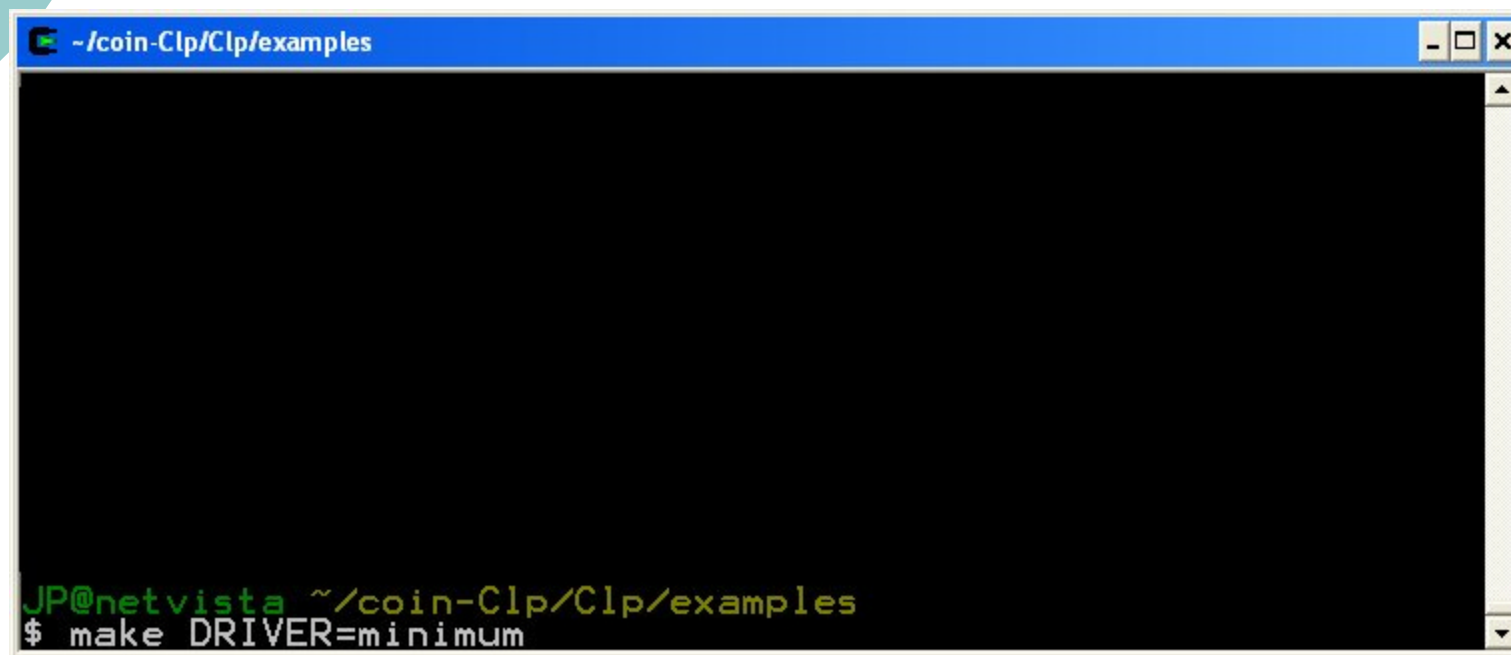
```
At line 2 ROWS
At line 78 COLUMNS
At line 354 RHS
At line 359 ENDATA
Problem BLEND has 74 rows, 83 columns and 491 elements
Model was imported from .\Data\Netlib\blend.mps in 0 seconds
Clp:solve
Presolve 53 (-21) rows, 60 (-23) columns and 387 (-104) elements
0 Obj 0 Dual inf 60.6004 (14)
76 Obj -31.4318 Primal inf 4.62337 (4)
82 Obj -30.8121
Optimal - objective value -30.8121
After Postsolve, objective -30.8121, infeasibilities - dual 0
(0), primal 0 (0)
Optimal objective -30.81214985 - 82 iterations time 0.012, Pr
esolve 0.01
Clp:solution $
Clp:solve
Enter ? for list of commands or help
Clp:import Data\Netlib\blend.mps
$ ./bin/clp
```



Building Examples

`coin-Clp/Clp/examples/minimum.cpp`
Unix

Build and run coin-Clp/Clp/examples/minimum.cpp

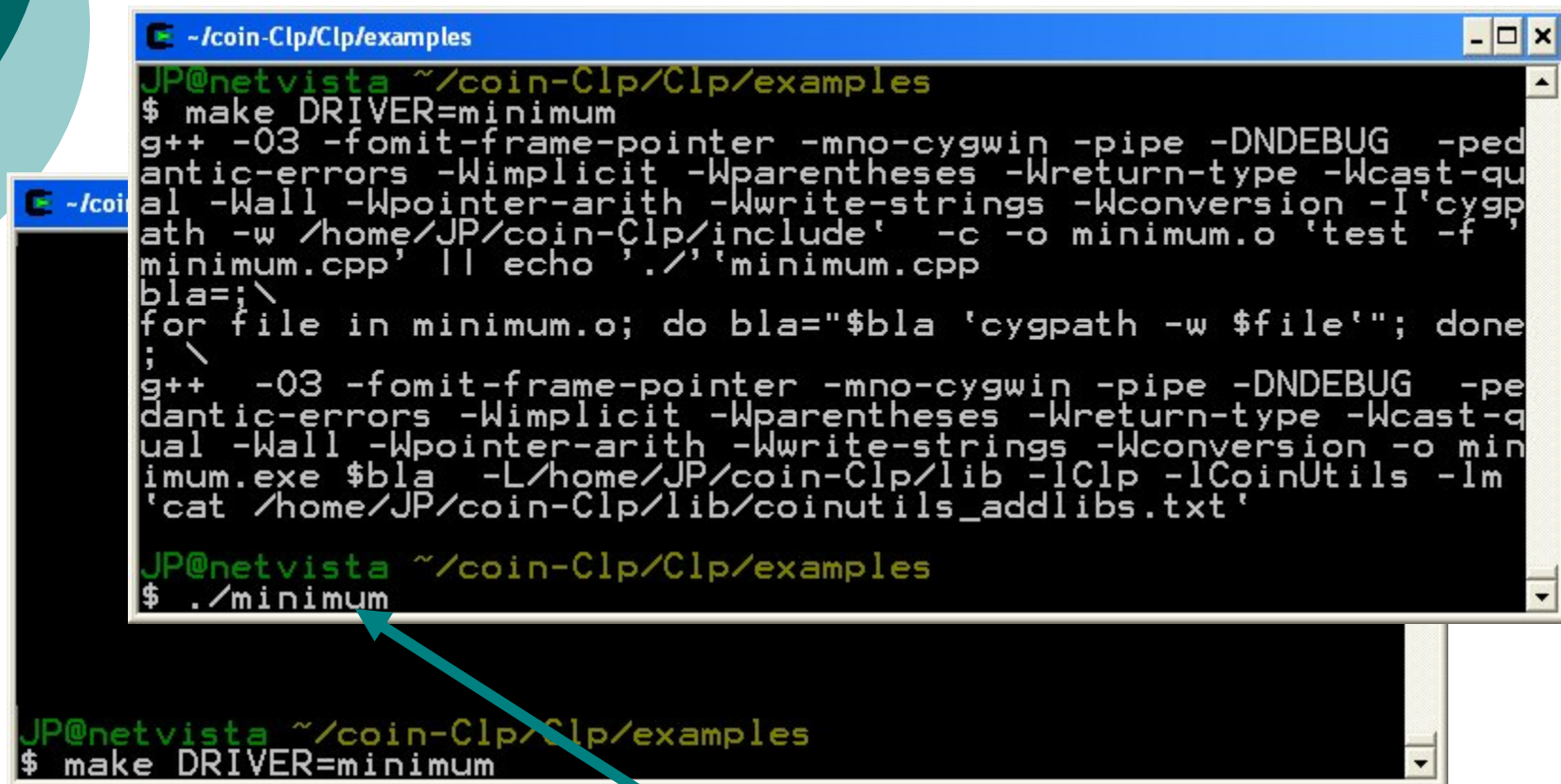


A terminal window with a blue title bar containing the text `~/coin-Clp/Clp/examples`. The terminal has a black background with green text. The prompt is `JP@netvista ~/coin-Clp/Clp/examples` and the command entered is `$ make DRIVER=minimum`. A green arrow points from the text 'EURO XXI' below to the `make` command.

```
~/coin-Clp/Clp/examples
JP@netvista ~/coin-Clp/Clp/examples
$ make DRIVER=minimum
```

EURO XXI

Build and run coin-Clp/Clp/examples/minimum.cpp



The image shows a terminal window with a blue title bar labeled `-/coin-Clp/Clp/examples`. The prompt is `JP@netvista ~/coin-Clp/Clp/examples`. The user enters `$ make DRIVER=minimum`. The terminal displays the following commands and output:

```
$ make DRIVER=minimum
g++ -O3 -fomit-frame-pointer -mno-cygwin -pipe -DNDEBUG -pedantic-errors -Wimplicit -Wparentheses -Wreturn-type -Wcast-qual -Wall -Wpointer-arith -Wwrite-strings -Wconversion -I'cygpath -w /home/JP/coin-Clp/include' -c -o minimum.o 'test -f minimum.cpp' || echo './'minimum.cpp
bla=;\
for file in minimum.o; do bla="$bla 'cygpath -w $file'"; done
; \
g++ -O3 -fomit-frame-pointer -mno-cygwin -pipe -DNDEBUG -pedantic-errors -Wimplicit -Wparentheses -Wreturn-type -Wcast-qual -Wall -Wpointer-arith -Wwrite-strings -Wconversion -o minimum.exe $bla -L/home/JP/coin-Clp/lib -lClp -lCoinUtils -lm 'cat /home/JP/coin-Clp/lib/coinutils_addlibs.txt'
```

The prompt returns to `JP@netvista ~/coin-Clp/Clp/examples`. The user enters `$ ./minimum`. A red arrow points from this command to a second terminal window below it. The second window shows the prompt `JP@netvista ~/coin-Clp/Clp/examples` and the command `$ make DRIVER=minimum`.

Build and run coin-Clp/Clp/examples/minimum.cpp

```
JP@netvista ~/coin-Clp/Clp/examples
$ ./minimum
Coin0001I At line 15 NAME P0033
Coin0001I At line 16 ROWS
Coin0001I At line 34 COLUMNS
Coin0001I At line 109 RHS
Coin0001I At line 118 BOUNDS
Coin0001I At line 152 ENDATA
Coin0002I Problem P0033 has 16 rows, 33 columns and 98 elements
Clp00027I Model was imported from ../../Data/Sample/p0033.mps
in 0.016 seconds
Clp0006I 0 Obj 0 Primal inf 18.4925 (10) Dual inf 5.59875e+0
11 (32)
Clp0006I 29 Obj 2520.57
Clp0000I Optimal - objective value 2520.57

JP@netvista ~/coin-Clp/Clp/examples
$ make DRIVER=minimum
g++ -O3 -std=c++11 -Wall -Wpointer-arith -Wwrite-strings -Wconversion -o minimum.exe $bla -L/home/JP/coin-Clp/lib -lClp -lCoinUtils -lm
'cat /home/JP/coin-Clp/lib/coinutils_addlibs.txt'

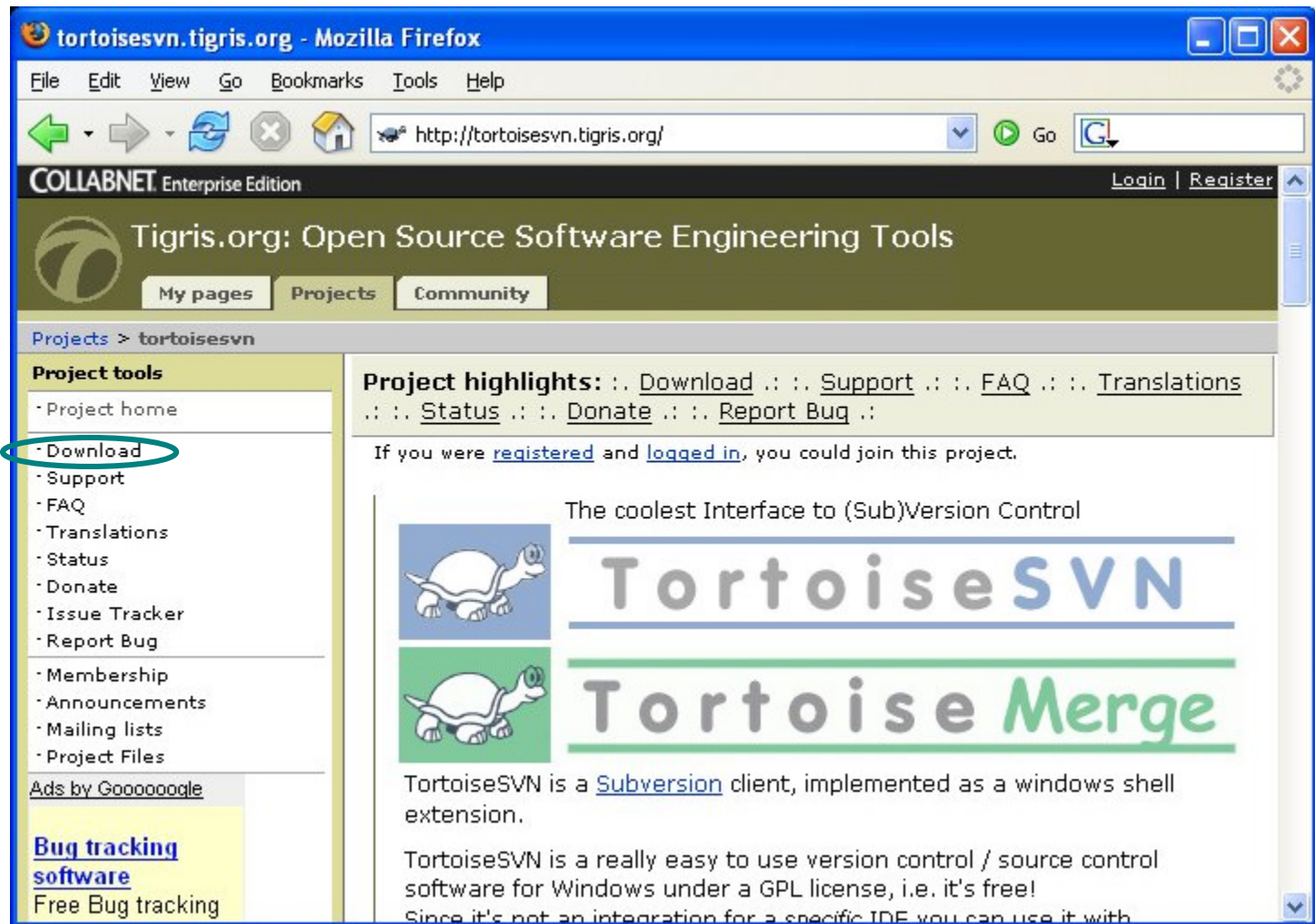
JP@netvista ~/coin-Clp/Clp/examples
$ ./minimum
```



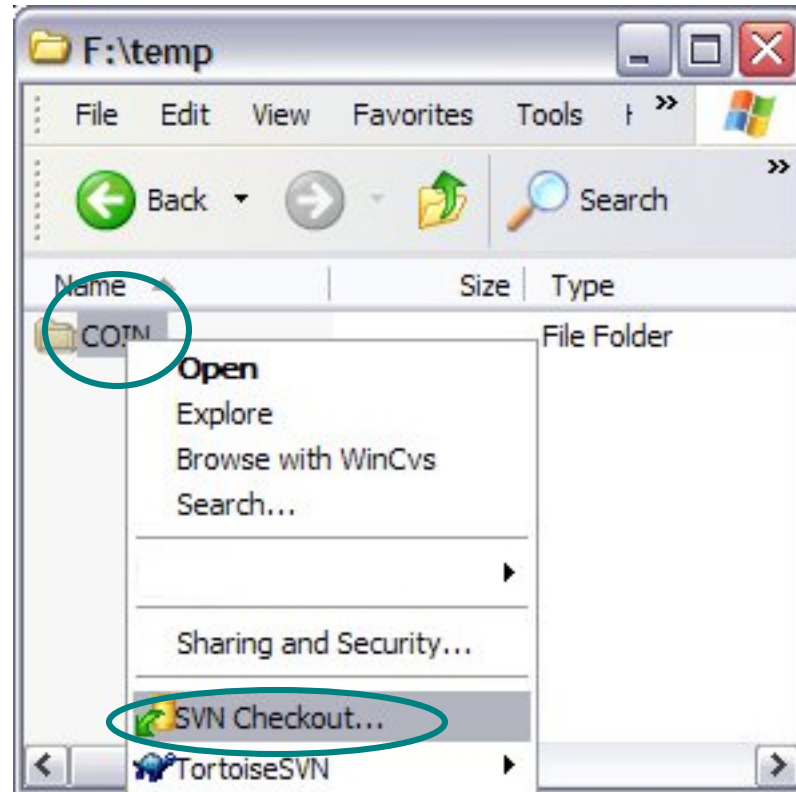
Windows: Obtaining Source

Using TortoiseSVN Client

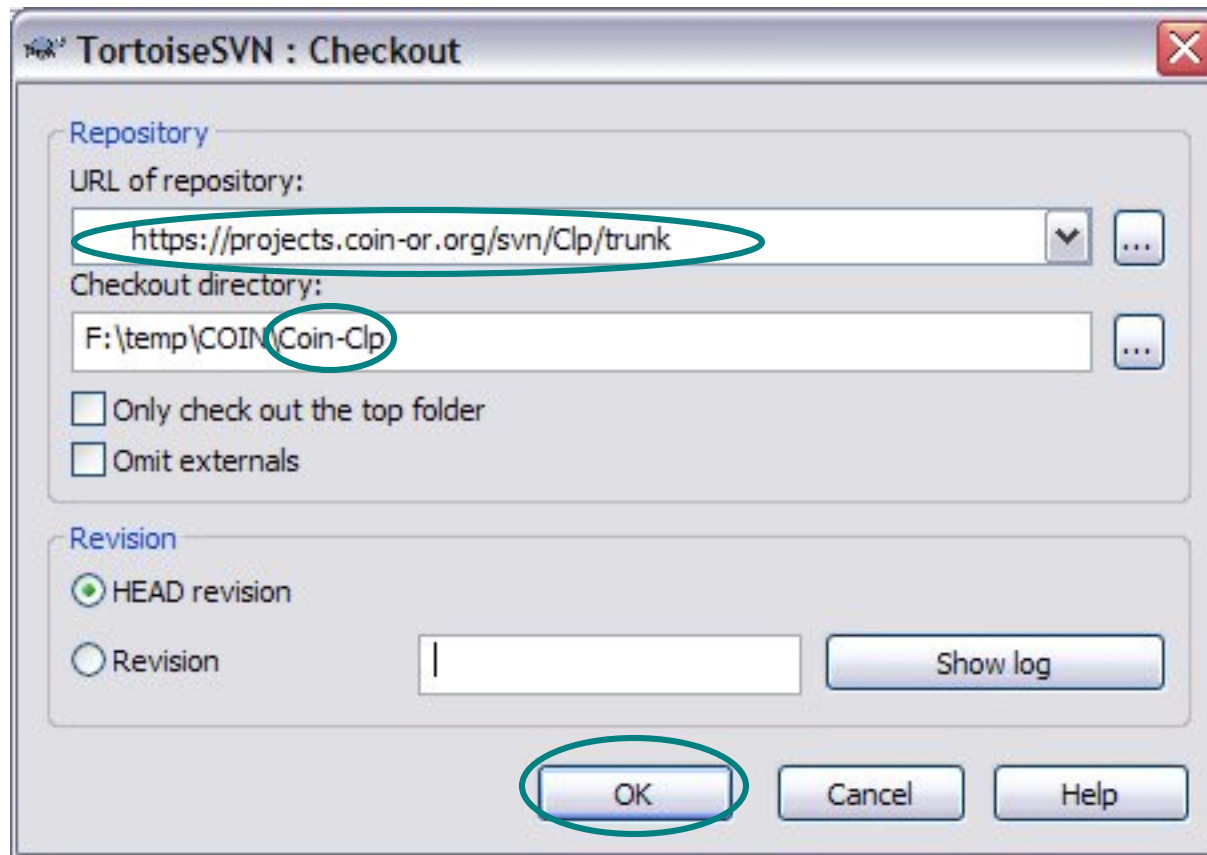
A Subversion Windows Client



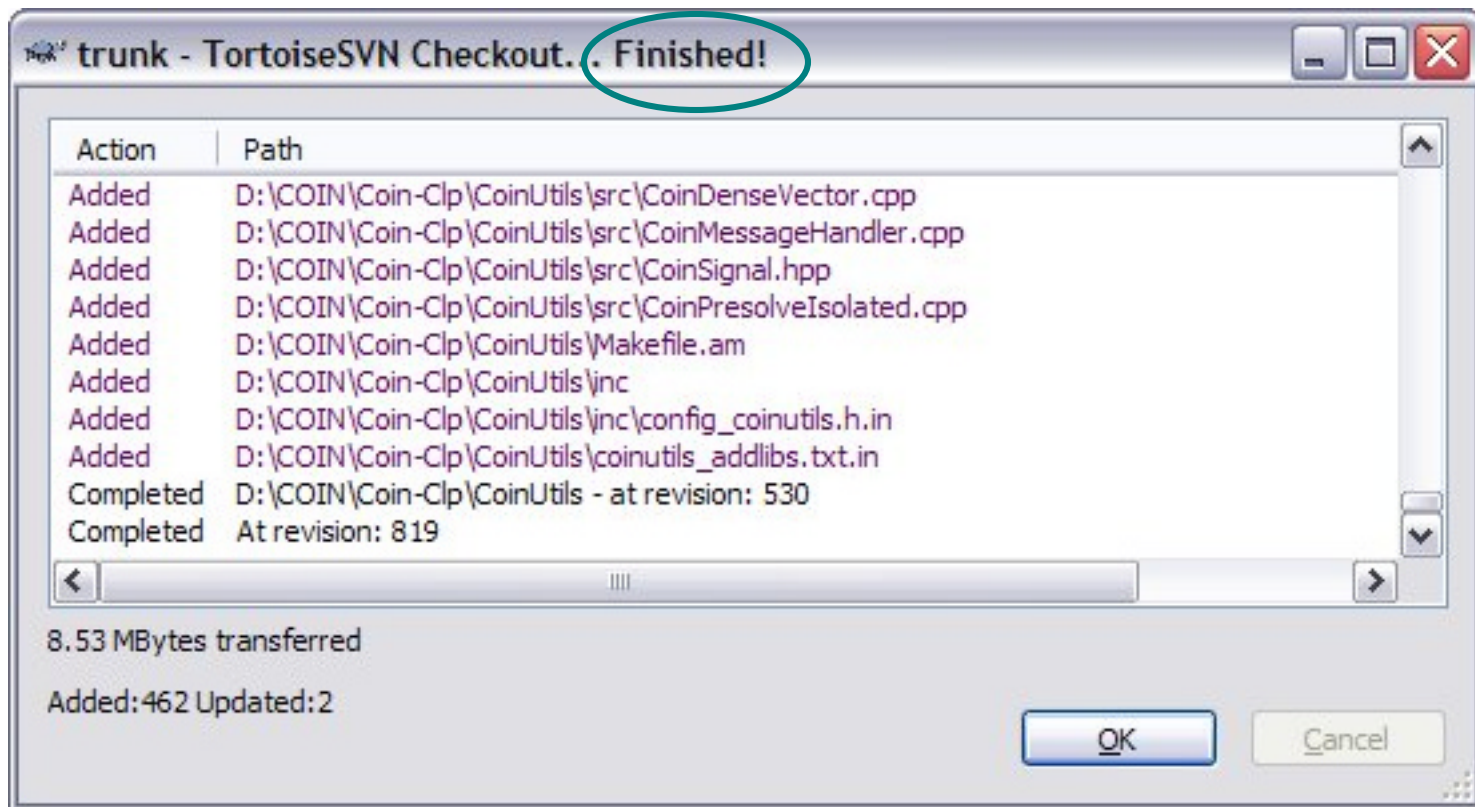
TortiseSVN: Checkout



TortoiseSVN: Checkout



TortiseSVN: Checkout





Build Clp

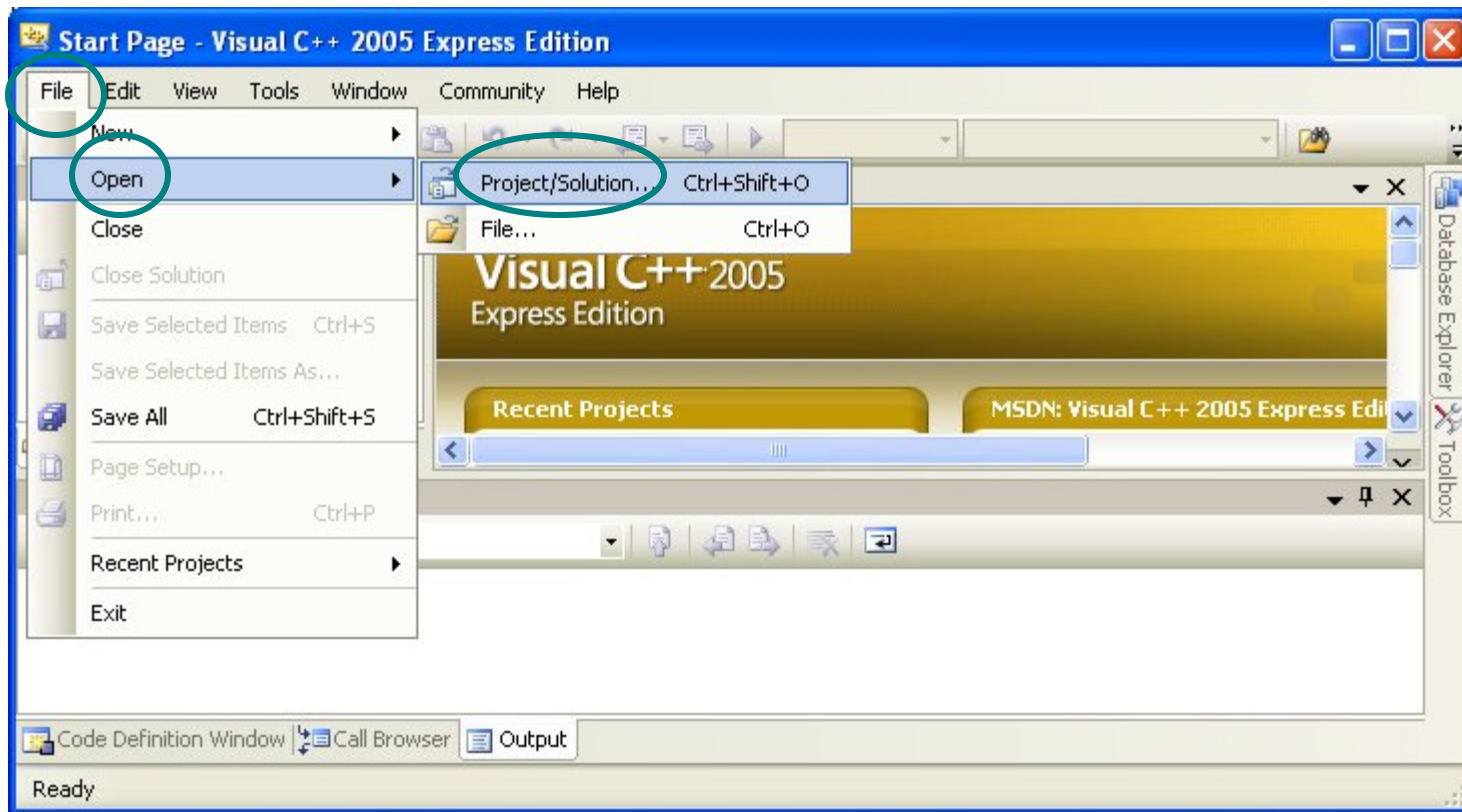
Windows



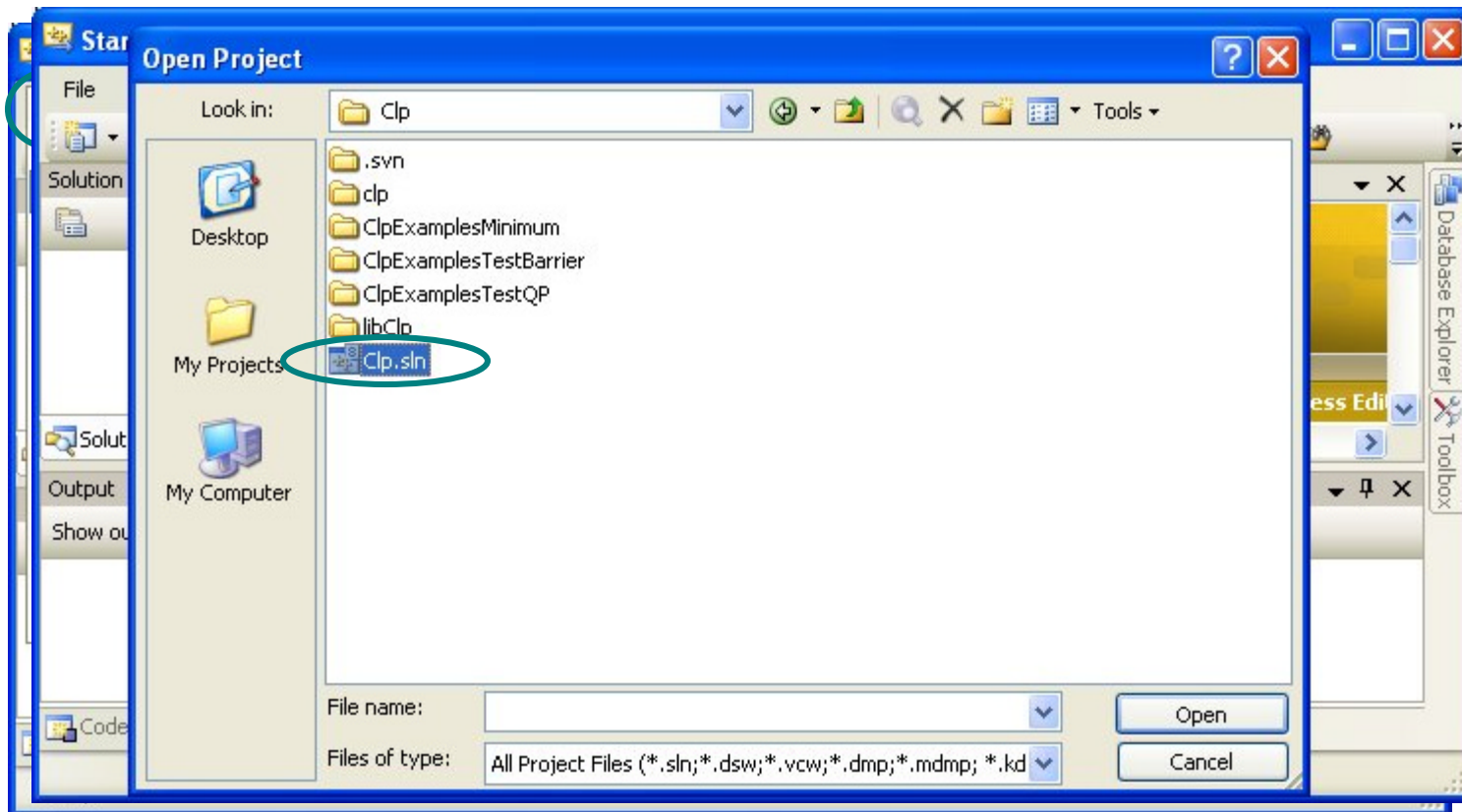
Microsoft Visual Studio 2005

- Visual C++ 2005 Express Edition
- “Free, lightweight, easy-to-use, and easy-to-learn tools for the hobbyist, novice, and student developer.”
- <http://msdn.microsoft.com/vstudio/express>
- <http://msdn.microsoft.com/vstudio/express/visualc>
- Must register to activate

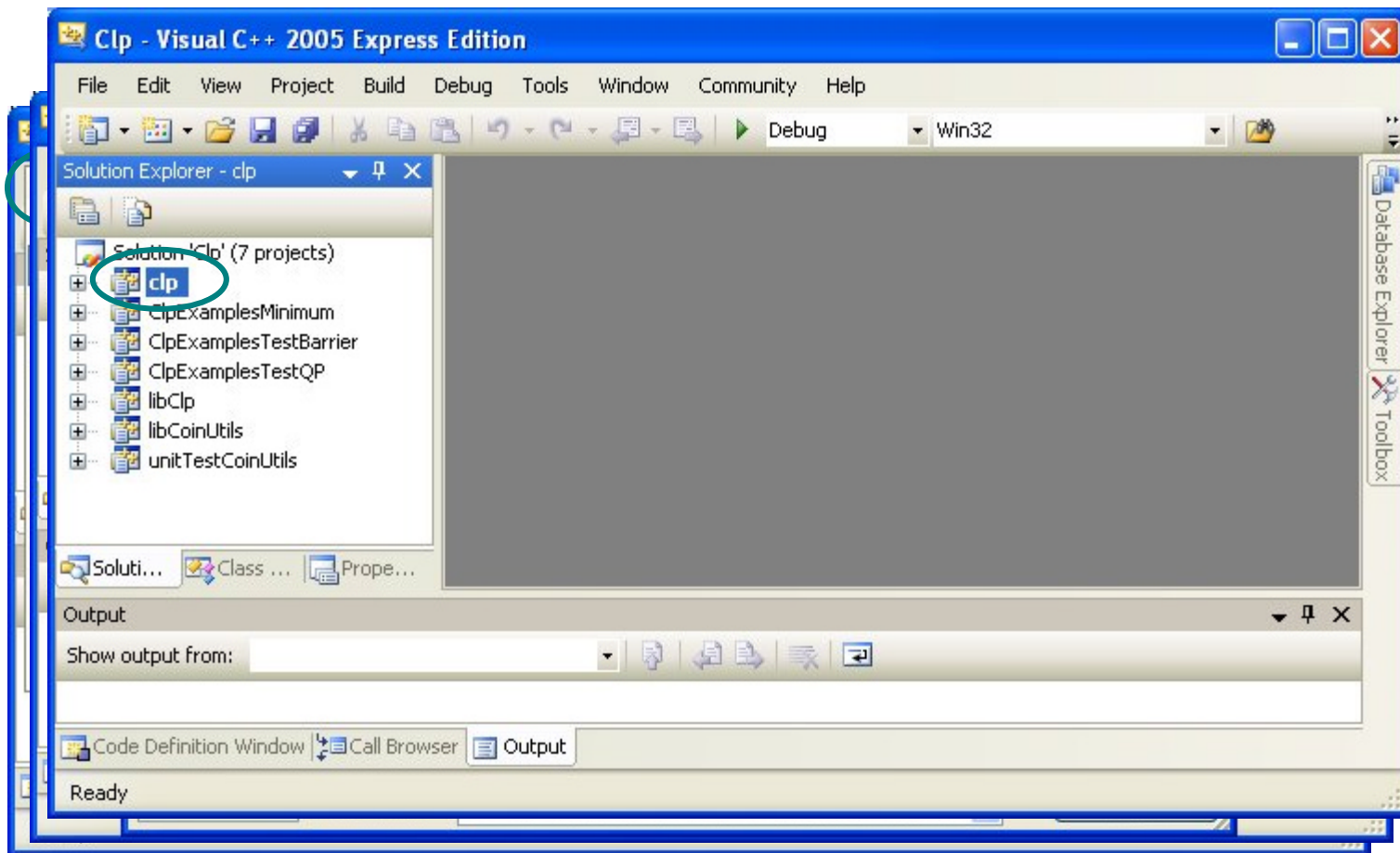
Build Clp with Visual C++ 2005 Express Edition



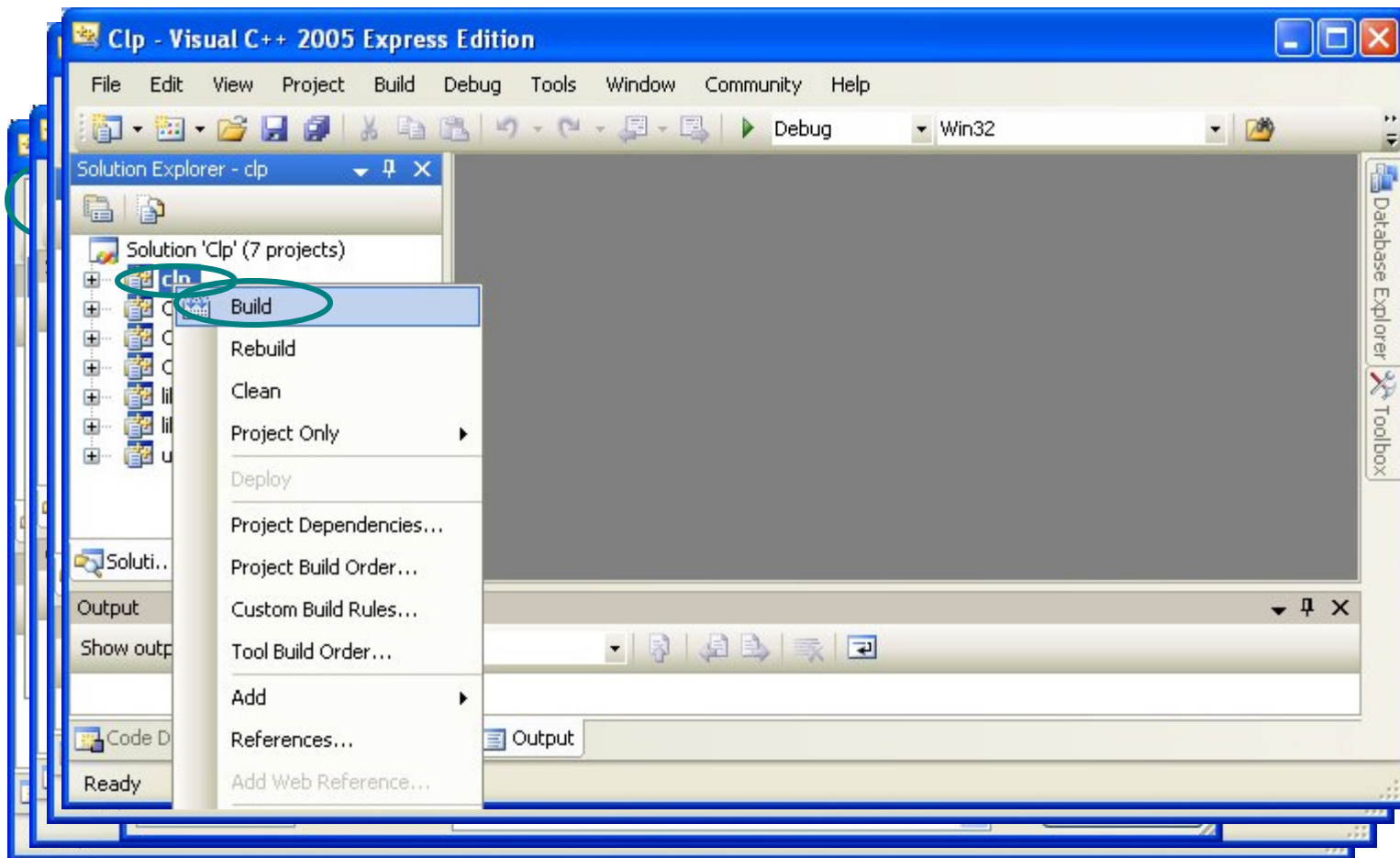
Build Clp with Visual C++ 2005 Express Edition



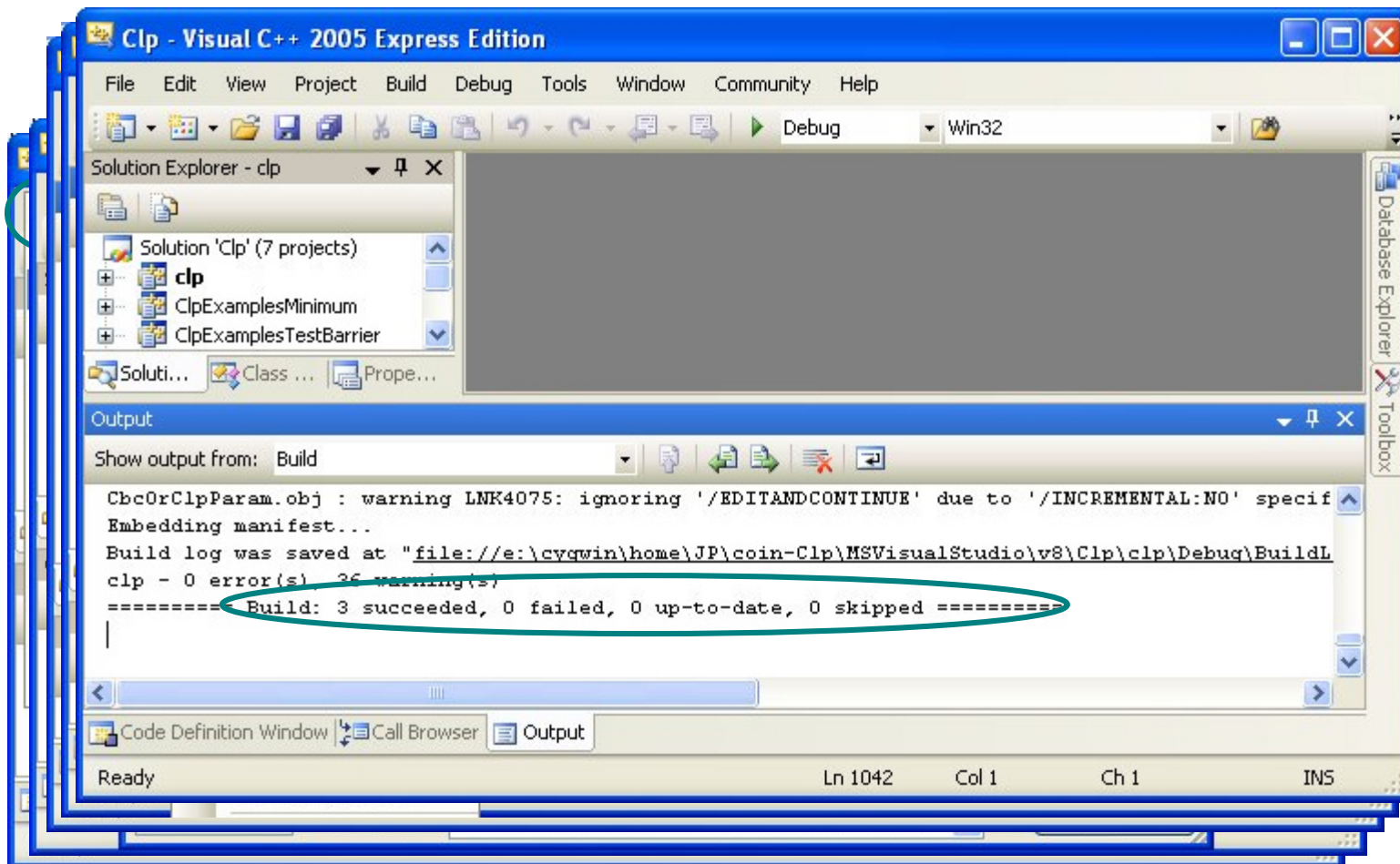
Build Clp with Visual C++ 2005 Express Edition



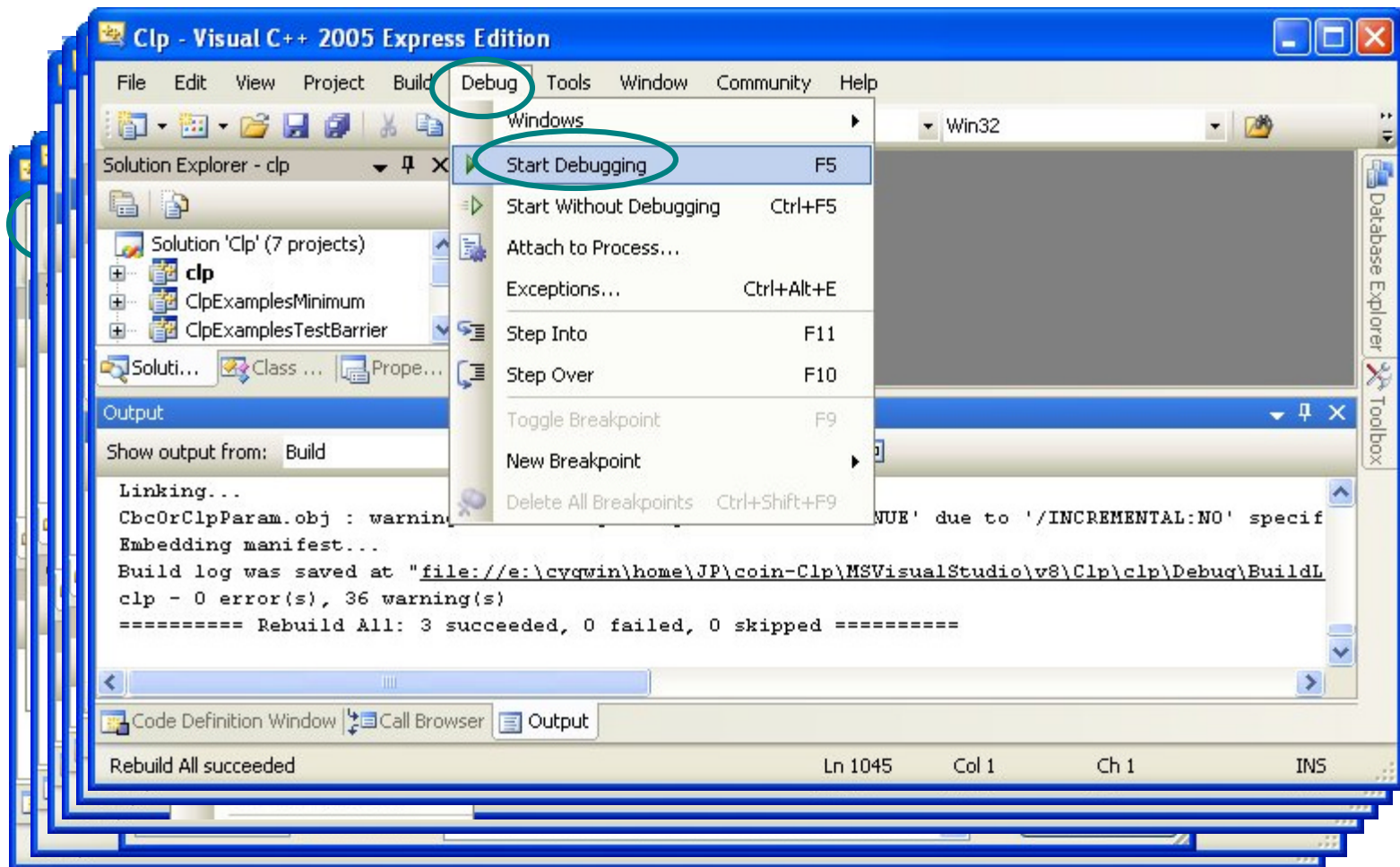
Build Clp with Visual C++ 2005 Express Edition



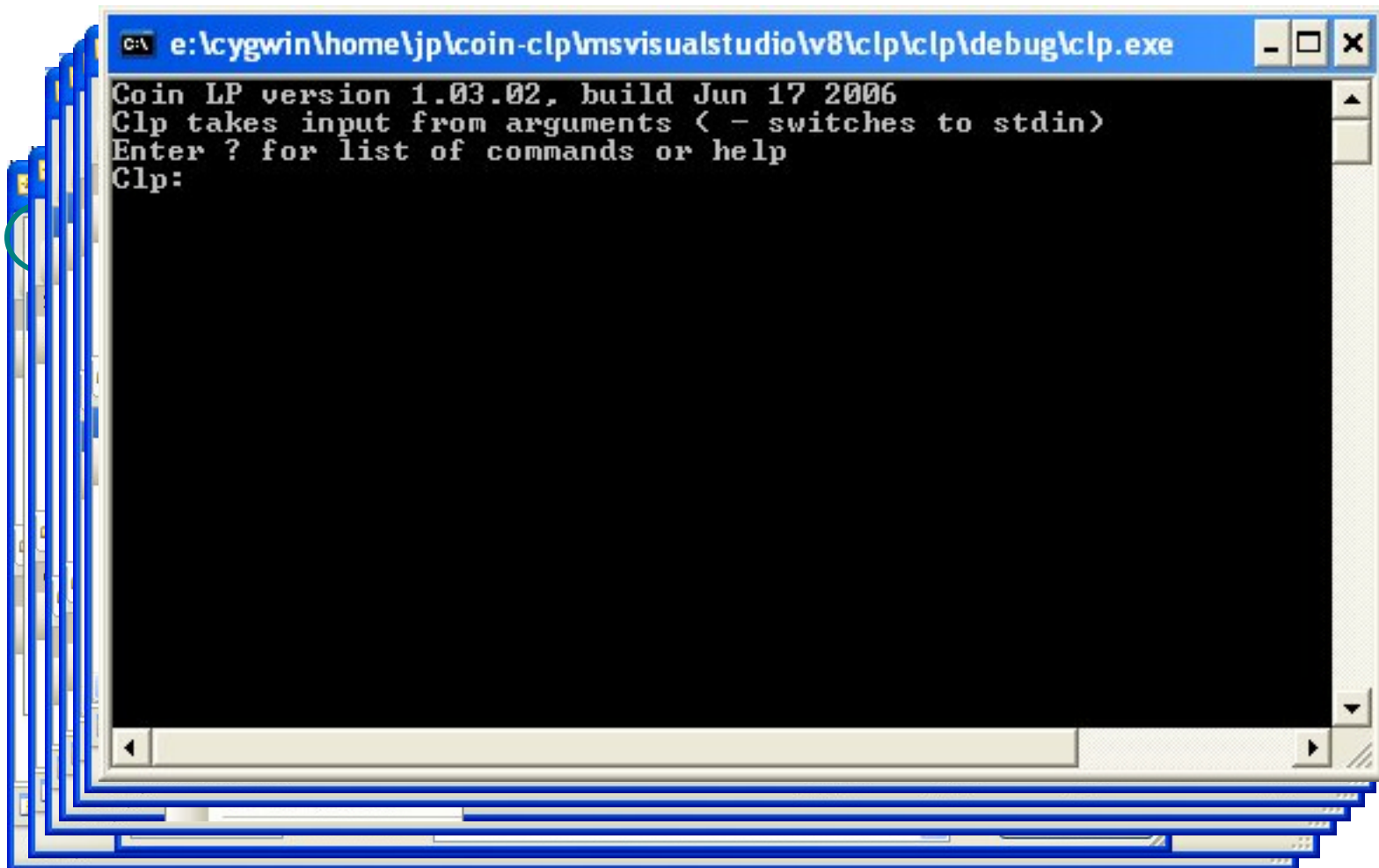
Build Clp with Visual C++ 2005 Express Edition



Build Clp with Visual C++ 2005 Express Edition



Build Clp with Visual C++ 2005 Express Edition

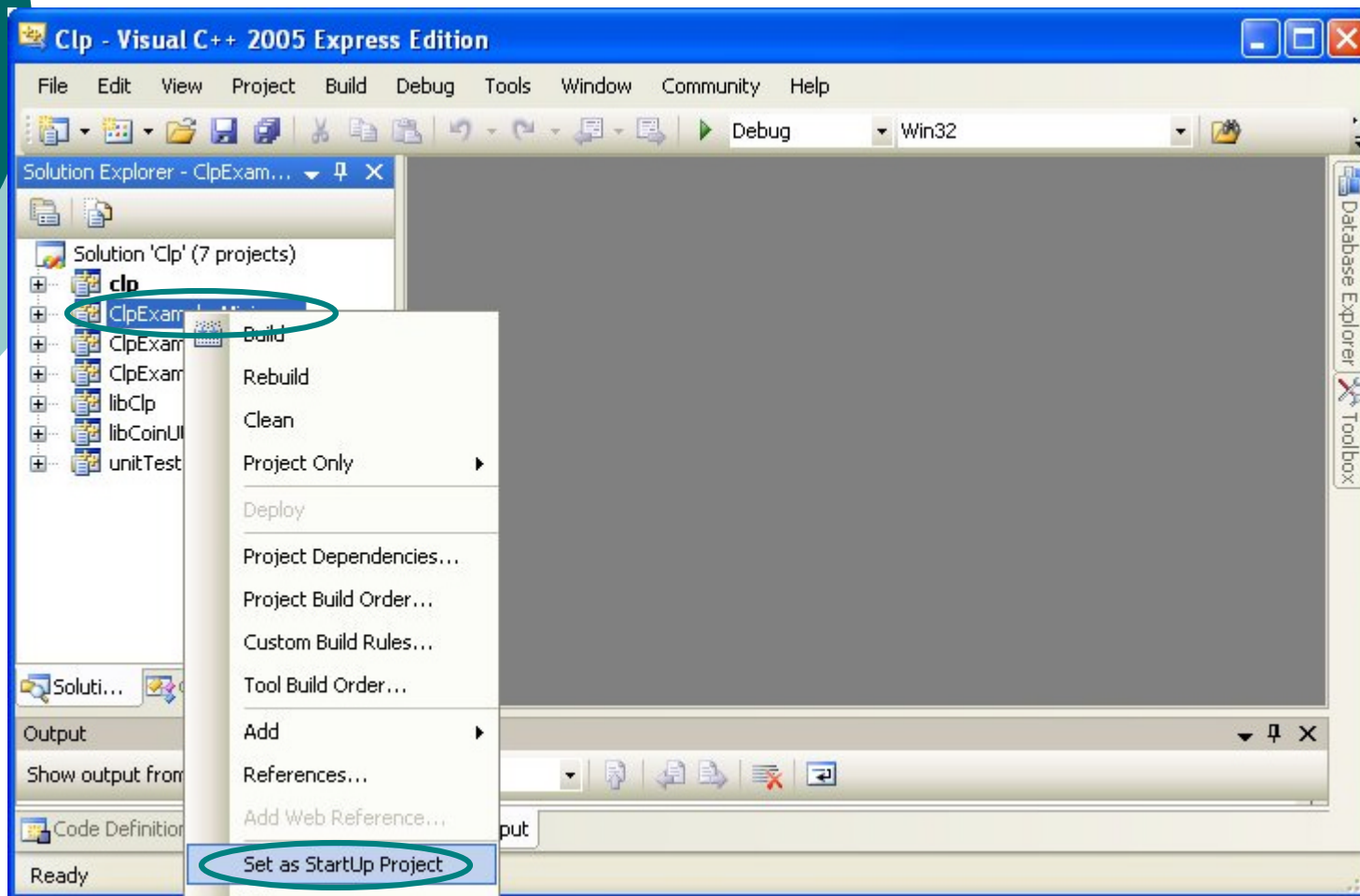




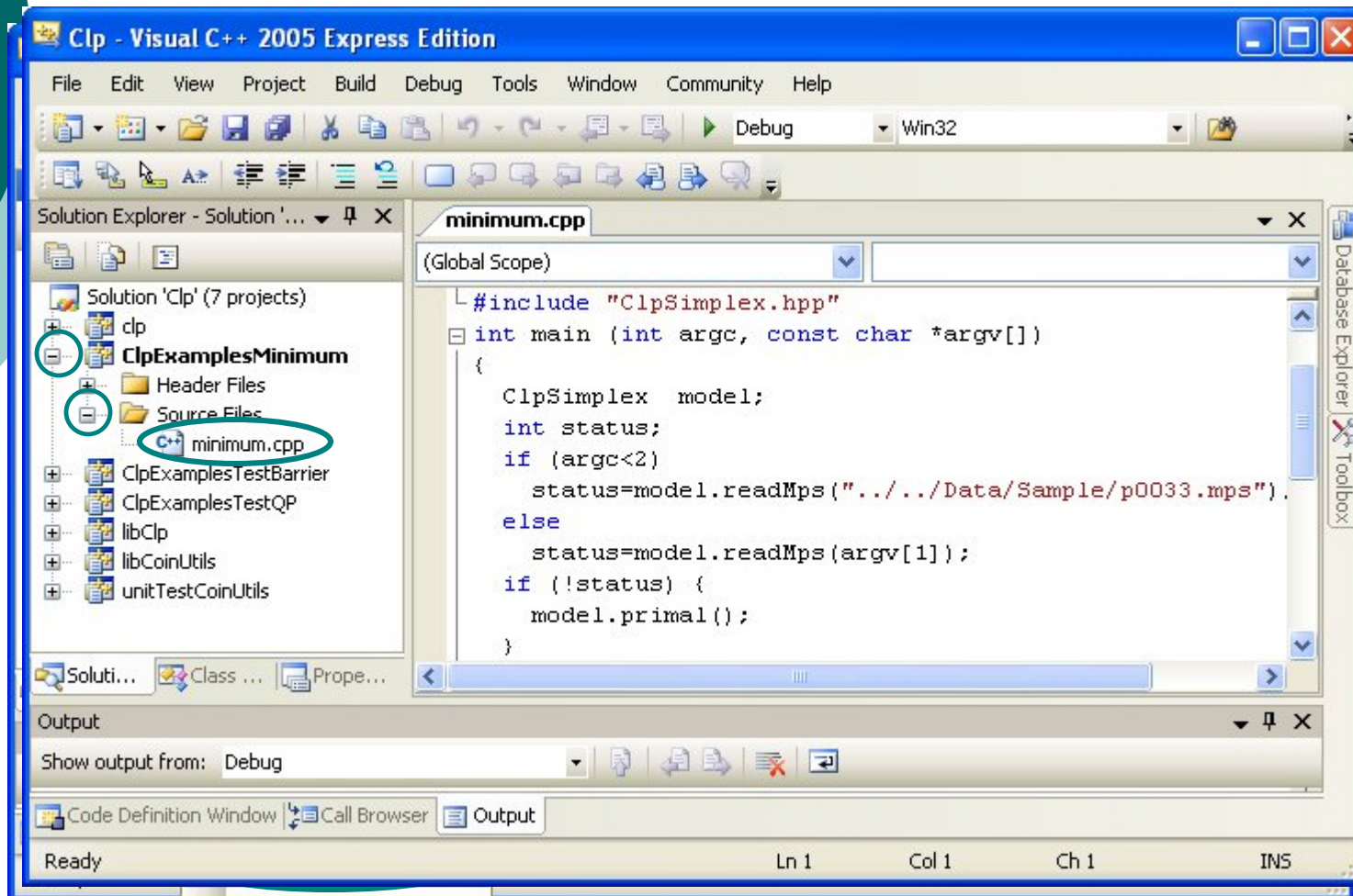
Building Examples

`coin-Clp/Clp/examples/minimum.cpp`
Windows

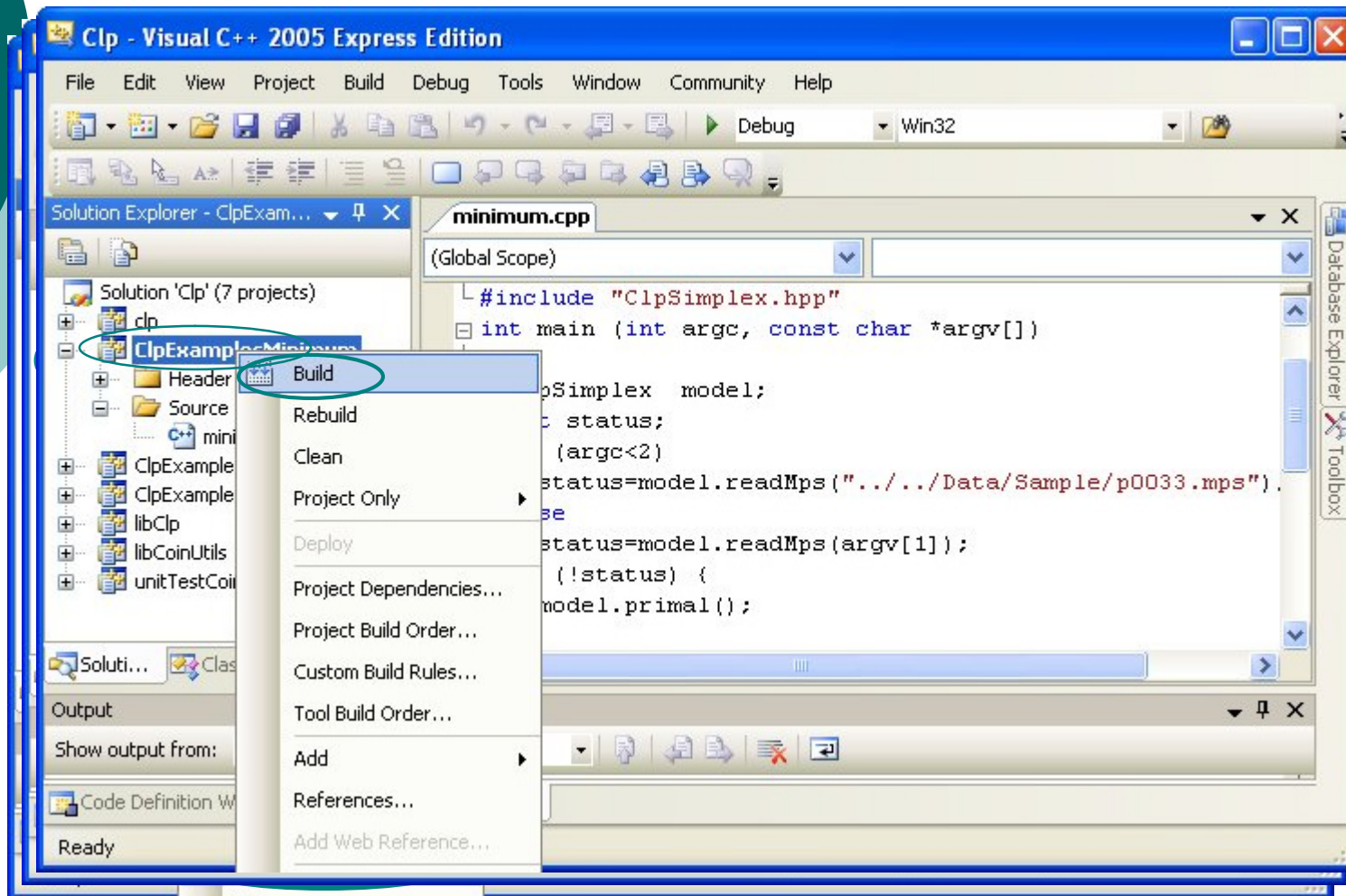
Build and run coin-Clp/Clp/examples/minimum.cpp



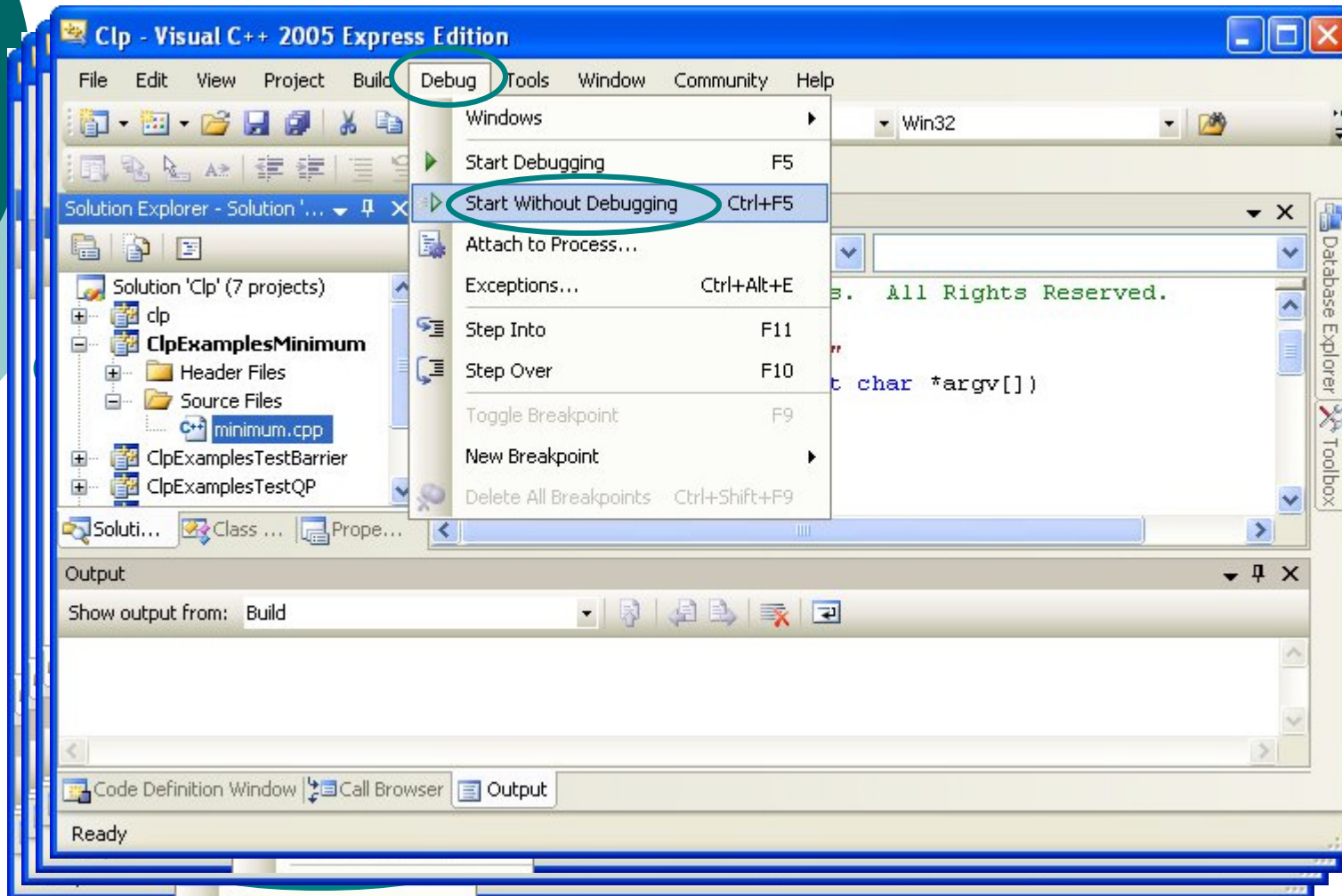
Build and run coin-Clp/Clp/examples/minimum.cpp



Build and run coin-Clp/Clp/examples/minimum.cpp



Build and run coin-Clp/Clp/examples/minimum.cpp



Build and run coin-Clp/Clp/examples/minimum.cpp

```
Clp - Visual C++ 2005 Express Edition
C:\WINDOWS\system32\cmd.exe
Coin0001I At line 15 NAME          P0033
Coin0001I At line 16 ROWS
Coin0001I At line 34 COLUMNS
Coin0001I At line 109 RHS
Coin0001I At line 118 BOUNDS
Coin0001I At line 152 ENDATA
Coin0002I Problem P0033 has 16 rows, 33 columns and 98 elements
Clp0027I Model was imported from ../Data/Sample/p0033.mps in 0 seconds
Clp0006I 0 Obj 0 Primal inf 18.4925 (10) Dual inf 5.59875e+011 (32)
Clp0006I 29 Obj 2520.57
Clp0000I Optimal - objective value 2520.57
Press any key to continue . . .
```



Summary: www.coin-or.org

- List of Projects
 - Project Trac Pages
 - Download
 - Source: Subversion or tarballs
 - Binaries
 - Build code
 - Unix
 - Microsoft Visual Studio
 - Documentation
 - Tickets: Bugs/Enhancements/Features
 - Wiki pages user community can update
 - Mailing lists