Web Site (www.coin-or.org)

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

JP Fasano
IBM Watson
Objective

- Familiarity with web site
  - Projects
  - Faqs, documentation, help
  - Source code
  - Building, running

- Build & run a COIN-OR project
www.coin-or.org Home Page

COIN-OR

COmputational INfrastructure for Operations Research

- open source for the operations research community -

The logo contest has begun! Submissions deadline July 7, 2006.
(32 entries so far.)

Come to the DIMACS Workshop on COIN-OR July 17-20, 2006 at
Rutgers University, NJ.

The Computational Infrastructure for Operations Research (COIN-OR**, or simply COIN) project is an initiative to spur the development of open-source software for the operations research community.

Why open source? The Open Source Initiative explains it well. When software is distributed under the open-source model, it allows...

DIMACS Workshop on COIN-OR, July 17, 2006, Rutgers University
Projects Page

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
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 (Coin-or linear programming) 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 from

\[
\begin{align*}
\min \ & c^T x \\
\text{subject to} \ & \text{row lower bound} \leq Ax \leq \text{row upper bound} \\
\ & \text{column lower bound} \leq x \leq \text{column upper bound}
\end{align*}
\]
Clp Build Instructions on Trac Page

```plaintext
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

Documentation

- User's Guide (single page format)
- Clp html documentation
- FAQ (Frequently Asked Questions)
- Source code examples

Project Links

- COIN-OR Initiative
- mailing list
- Report a bug

Download in other formats:
Plain Text
Build Clp

Unix
Build Clp on Unix

$ svn co https://projects.coin-or.org/svn/Clp/trunk coin-Clp
Build Clp on Unix

Checked out revision 818.

$ cd coin-Clp

$ ./configure -C

$ svn co https://projects.coin-or.org/svn/Clp/trunk coin-Clp
Build Clp on Unix

```bash
$ svn co https://projects.coin-or.org/svn/Clp/trunk coin-Clp
$ ./configure -C
$ make
$ make
```

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
configure: Configuration of Clp successful
configure: Main configuration of Clp successful
Build Clp on Unix

```bash
$ ./configure -C
$ make
$ make test
```

```
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/Clp'
make[1]: Nothing to be done for 'all-am'.
make[1]: Leaving directory `/home/JP/coin-Clp/Clp'
```
Build Clp on Unix
Run Clp executable

Experiment with Clp options
Experimenting with clp executable
Experimenting with clp executable
Experimenting with clp executable

```bash
$ ./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 PROBLEM
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
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
$ ./bin/clp
```
Experimenting with clp executable
Building Examples

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

```
JP@netvista ~/coin-Clp/Clp/examples
$ make DRIVER=minimum
for file in minimum.o; do bla="$bla 'cygpath -w $file'"; done

JP@netvista ~/coin-Clp/Clp/examples
$ ./minimum
```
Build and run coin-Clp/Clp/examples/minimum.cpp
Windows: Obtaining Source

Using TortoiseSVN Client
A Subversion Windows Client

Tigris.org: Open Source Software Engineering Tools

Project highlights: Download, Support, FAQ, Translations, Status, Donate, Report Bug

If you were registered and logged in, you could join this project.

The coolest Interface to (Sub)Version Control

TortoiseSVN

TortoiseMerge

TortoiseSVN is a Subversion client, implemented as a windows shell extension.

TortoiseSVN is a really easy to use version control / source control software for Windows under a GPL license, i.e. it's free!

Since it's not an integration for a specific IDE you can use it with...
TortiseSVN: Checkout
TortiseSVN: Checkout

Repository

URL of repository:
https://projects.coin-or.org/svn/Clp/trunk

Checkout directory:
F: \temp\COIN\Coin-Clp

Revision

HEAD revision

OK
TortiseSVN: Checkout

![Checkout Window]

- **Action**: Added
- **Path**: 
  - D:\COIN\Coin-Clp\CoinUtils\src\CoinDenseVector.cpp
  - D:\COIN\Coin-Clp\CoinUtils\src\CoinMessageHandler.cpp
  - D:\COIN\Coin-Clp\CoinUtils\src\CoinSignal.hpp
  - D:\COIN\Coin-Clp\CoinUtils\src\CoinPresolveIsolated.cpp
  - D:\COIN\Coin-Clp\CoinUtils\Makefile.am
  - D:\COIN\Coin-Clp\CoinUtils\inc
  - D:\COIN\Coin-Clp\CoinUtils\inc\config_coinutils.h.in
  - D:\COIN\Coin-Clp\CoinUtils\coinutils_addlibs.txt.in
- **Completed**: 
  - D:\COIN\Coin-Clp\CoinUtils - at revision: 530
  - At revision: 819

8.53 MBytes transferred
Added: 462 Updated: 2
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

The image shows a screenshot of the Visual C++ 2005 Express Edition IDE with a project named 'Clp'. The build log indicates that the build was successful with 3 succeeded builds, 0 failed, 0 up-to-date, and 0 skipped.

```
ClpProcessor.obj : warning LNK4075: ignoring 'EDITANDCONTINUE' due to '/INCREMENTAL:NO' specified
Embedding manifest...
Build log was saved at "file://e\cyquin\home\J\coin-Clp\MSVisualStudio\v8\Clp\clp\Debug\Build1"
Clp - 0 error(s), 0 warning(s)
======== build: 3 succeeded, 0 failed, 0 up-to-date, 0 skipped =========
```
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
Windows: Alternatives to using Visual Studio IDE

- On Windows the Unix build process can be used with Cygwin or MinGW.
  - www.cygwin.com
  - www.mingw.org
- The gcc or Microsoft cl compiler can be used.
Summary: www.coin-or.org

- 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
Remaining Time:

- www.coin-or.org
- From list of projects, select project to build
- Go to projects trac page
  - Download source
  - Build source