

# Using COIN-OR Solvers with Visual Studio

Horand Gassmann, Jun Ma, and Kipp Martin

June 25, 2009

## **Abstract**

This is for Windows users that want to run COIN-OR solvers directly from a file or a modeling language, or want to write applications with Visual Studio projects that use COIN-OR solvers. This download is plug-and-play it is not necessary to build any COIN-OR projects from source code.

## Contents

<b>1</b>	<b>The Download</b>	<b>3</b>
<b>2</b>	<b>Calling COIN-OR Solvers with Model Instances</b>	<b>3</b>
<b>3</b>	<b>Calling COIN-OR Solvers using a Modeling Language</b>	<b>3</b>
<b>4</b>	<b>Using Visual Studio to Build Application</b>	<b>3</b>
<b>5</b>	<b>Example Projects</b>	<b>3</b>

## List of Figures

## List of Tables

## 1 The Download

## 2 Calling COIN-OR Solvers with Model Instances

In the `bin` directory, the user will find the following solvers (see the project link for more information on the solvers):

- **bonmin.exe** – a solver for mixed-integer nonlinear programs – see <https://projects.coin-or.org/Bonmin>
- **cbc.exe** – a solver for mixed-integer linear programs – see <https://projects.coin-or.org/Cbc>
- **clp.exe** – a solver for linear programs – see <https://projects.coin-or.org/Clp>
- **couenne.exe** – a global optimizer for mixed-integer nonlinear programs – see <https://projects.coin-or.org/Couenne>
- **ipopt.exe** – an optimizer for continuous nonlinear programs – see <https://projects.coin-or.org/Ipopt>
- **symphony.exe** – a solver for mixed-integer linear programs – see <https://projects.coin-or.org/SYMPHONY>

See the project pages for a more detail on each of the solvers and which optimization instance format they take.

For the convenience of the user, the `bin` directory also contains the **OSSolverService.exe**. This executable is linked to libraries for all of the above solvers, and can be used in lieu of any of them. One advantage of using the *OSSolverService.exe* is its flexibility. You can call any of the above solvers with an instance in MPS, nl, or OSiL format. In addition, the **OSSolverService.exe** returns the solver solution in the OSrL XML format which is easily parsed.

See also for calling remote solvers.

## 3 Calling COIN-OR Solvers using a Modeling Language

Mention GAMSlinks

## 4 Using Visual Studio to Build Application

## 5 Example Projects