AMPLDev User Guide

OptiRisk Systems

Version: 3.2.0

Prepared by
Neha Murarka, Victor Zverovich, Christian Valente, Cristiano Arbex Valle and Gautam Mitra
OptiRisk Systems

OptiRisk Systems
One Oxford Road
Uxbridge, Middlesex, UB9 4DA
United Kingdom
www.optirisk-systems.com
+44 (0) 1895 256484

Contents

Part I

Overview of AMPLDev

Chapter 1

Scope and Purpose

1.1 What is AMPL?

AMPL[48] is an algebraic modelling language that is used for formulating and solving optimization problems. AMPL supports linear and non-linear programming optimization models covering both discrete and continuous variables. Stochastic extension of AMPL (SAMPL) has comprehensive support for stochastic programming; SAMPL is an extension of AMPL. You can easily create models and establish connection to various data sources, such as spreadsheets, databases or plain text files, within AMPL. From the model and data, the AMPL translator generates an internal representation, passes it to one of the many solvers supported by AMPL and provides you with the results. The language is simple and generally follows the mathematical notation used by both mathematicians and practitioners in the field of operations research.

AMPLDev is a graphical interface for AMPL. It is based on the popular Eclipse development platform and is available as a stand-alone application and as a plug-in for Eclipse.

For a general introduction to AMPL see [48]; a description SAMPL may be found in (reference).

AMPLDev as a stand-alone application

This version is a complete bundle of the necessary components required to run AMPLDev without the complications of installing any prerequisites. You can think of it as a core Eclipse IDE with the AMPLDev plug-in preinstalled. The only drawback is that this stand-alone version of AMPLDev may have limited capabilities of extending with other Eclipse plug-ins.

If you do not require Eclipse for any other reason, then we recommend installing this version of AMPLDev.

1.2 Who can use AMPLDev?

AMPLDev can be used by anyone who is considering using an algebraic modelling language and would like a quick and easy way to start off the process of learning AMPL, and even for those who are already using AMPL and would like a modern IDE for this language. The common areas that use such a tool are distribution, production, scheduling and other areas that have large-scale optimization problems.

To use AMPLDev, you are expected to have a basic knowledge of programming and some optimisation modelling experience. Experience with Eclipse or any other development environment, although not required, will considerably speed up the learning process.

1.3 Why use AMPLDev?

AMPLDev gives AMPL users the benefits of a modern Integrated Development Environment (IDE) which has the following features.

- A smart editor with context-sensitive syntax highlighting.
- Efficient error reporting with the ability to instantly go to the error location.
- A solution view which organises and separates the results from solving the model.
- A project explorer that allows you to organise all your projects and corresponding folders with useful context menus that directly allow you to run AMPL files.
- Built-in interactive AMPL console.
- Outline view that shows the model components: parameters, sets, variables, objectives and constraints.

For users who are new to AMPL such as students who are learning modelling using AMPL for the first time, the carefully designed Graphical User Interface (GUI) helps to easily get started. At the same time AMPLDev is beneficial for advanced users who need such features as built-in interactive console, integration with version control systems provided by Eclipse and projects supporting other programming languages in addition to AMPL.

For the users who wish to install the plug-in version of AMPLDev, Eclipse is absolutely free to use and has minimal installation requirements. Integrating AMPL with Eclipse is a huge advantage if you are already an Eclipse user and can have one software that takes care of all your development needs. In addition, you can even extend the AMPLDev plug-in if you wish to customize it further for your needs.

Chapter 2

Installing AMPLDev

This chapter describes how to set up your computer for using AMPLDev. The first section covers installing the AMPLDev stand-alone application which is the essential step of getting started with AMPLDev. The second section describes how to install the AMPLDev plug-in in Eclipse; primarily for advanced users.

2.1 AMPLDev stand-alone application

This is the fastest way to start using AMPLDev. Download the archive from the website onto your local machine, unless you have already received the archive via other means.

Once you extract all the files from the archive, in the root folder you will find the AMPLDev executable, double-click it and you are good to go!

The stand-alone application does not require any Java Runtime Environment (JRE) or Eclipse installations as these items of software are all bundled in with the executable.

2.2 AMPLDev plug-in for Eclipse users (Not supported yet)

If you already have Eclipse set up on your machine (a version from 4.1 onwards), then you may skip the next two sections.

First, install the JRE on your machine. Following that, the next section describes how to get Eclipse and install it. The last subsection explains how to install the AMPLDev plug-in into your Eclipse Platform so that you can program in AMPL using Eclipse.

Java Runtime Environment installation

The JRE needs to be installed to run Eclipse. You can download the latest version of the JRE by following the link, `http://www.java.com/en/download/manual.jsp`. It is recommended to use the JRE of version 7 or higher for the Eclipse version (Eclipse Juno 4.2); our system which is based on this version, is described in this manual. If you already have a JDK installed, make sure it is version 1.7 and up.

TIP:

Many computers already have the JRE installed but if you are not sure, the Java website has a nice tool that checks your computer for you and asks you to download the latest version if you do not have any or if you need an upgrade. You can either search for 'Do I have Java?' or use the following link, `http://www.java.com/en/download/installed.jsp?detect=jre&try=1`.

Getting Eclipse

Once you have the JRE set up, your computer is now ready for Eclipse. The version used throughout the book is Eclipse Juno 4.2 and can be freely downloaded from the Eclipse website (`http://www.eclipse.org/downloads/`). For users, who wish to use a previous version, it is recommended to use only versions from Eclipse 4.1 onwards.

On the download page, there are many different types of Eclipse Juno packages available; we recommend that you download the Classic version. Most of the others focus on a specific development area such as Java or PHP but the Classic version is an all round Integrated Development Environment (IDE) with all the relevant tools needed for our purpose. Eclipse is available for major operating systems (Windows, Linux and Mac OS) and supports 32-bit and 64-bit platforms.

TIP:

Some 64-bit machines may have a 32-bit JRE installed; that is fine as long as the Eclipse is also 32-bit. Hence when downloading Eclipse, it does not depend upon your machine but upon the JRE installed. For example, a 64-bit Eclipse will not work on a 64-bit machine with a 32-bit JRE; either get a 32-bit Eclipse or a 64-bit JRE.

You can choose from numerous mirror sites, available across Europe, Asia, North America, South America and Australia, depending upon your location for better download speeds.

Once the file is downloaded, unzip it at your desired location. As such, there is no setup file that needs to be run for the installation. Then run the eclipse executable (eclipse.exe for Windows' users) which is located inside the extracted 'eclipse' folder. But don't forget to install the AMPLDev plug-in if you would like to use AMPL!

Installing AMPLDev plug-in

There are two ways to install the AMPLDev plug-in; these two ways are described below.

Installing AMPLDev plug-in via update site - recommended

1. Once you have downloaded the AMPL plug-in archive on your computer, start Eclipse and click on **Install New Software...** in the **Help** menu.
2. Click on the **Add...** button.
3. If you have extracted the plug-in file, click on **Local...** and navigate to the AMPL plug-in folder or do the same with **Archive...**, if you have not extracted the AMPL plug-in.
4. Click on OK and the **Install New Software...** window will list the available AMPL plug-in.
5. Select the plug-in and click Next.
6. Follow the default settings and the AMPL plug-in will be installed.
7. Restart Eclipse and you should have AMPL support!

TIP:

If you cannot see the AMPL plug-in listed as an option to install, uncheck the **Group items by category**.

Installing AMPLDev plug-in manually

1. Extract all the files from the AMPL plug-in archive to any location. The location does not matter as you will soon be copying the required folders into Eclipse.

2. Once the archive has been completely extracted, you will find a 'plugins' folder inside it. Copy the contents.

3. Navigate to the folder where you have installed your Eclipse. Inside the 'eclipse' folder, you will see a folder named 'dropins'. Paste the previously copied 'plugins' folder into the 'dropins' folder.

4. Repeat the previous two steps for the 'features' folder in the extracted plug-in archive by copying the contents into Eclipse's dropins' 'features' folder.

5. Simply start Eclipse or restart it (if it was already running) and you should have AMPL support!

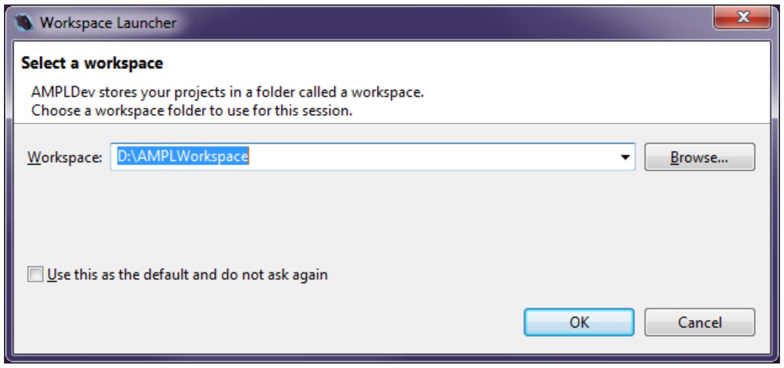
Chapter 3

AMPLDev User Interface

In this chapter the Graphical User Interface for AMPLDev is described. The basic interface is described here along with its various parts and some important menu items as well. You will notice that there are many features provided by Eclipse which are for more advanced users and are not paramount to using AMPL. For Eclipse features, we recommend referencing the Eclipse Help sections. This document pertains to AMPLDev and we will describe the various functionalities, views and graphical tools relevant to AMPL.

3.1 Workbench

Once the computer has been set up for AMPLDev (see chapter 2), run the ampldev executable (ampldev.exe for Windows' users) to start an AMPLDev session. When AMPLDev starts, it will ask for a workspace; this is essentially a directory on the computer in which it will save the projects (see figure 3.1) you create.



AMPLDev stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: D:\AMPLWorkspace

Use this as the default and do not ask again

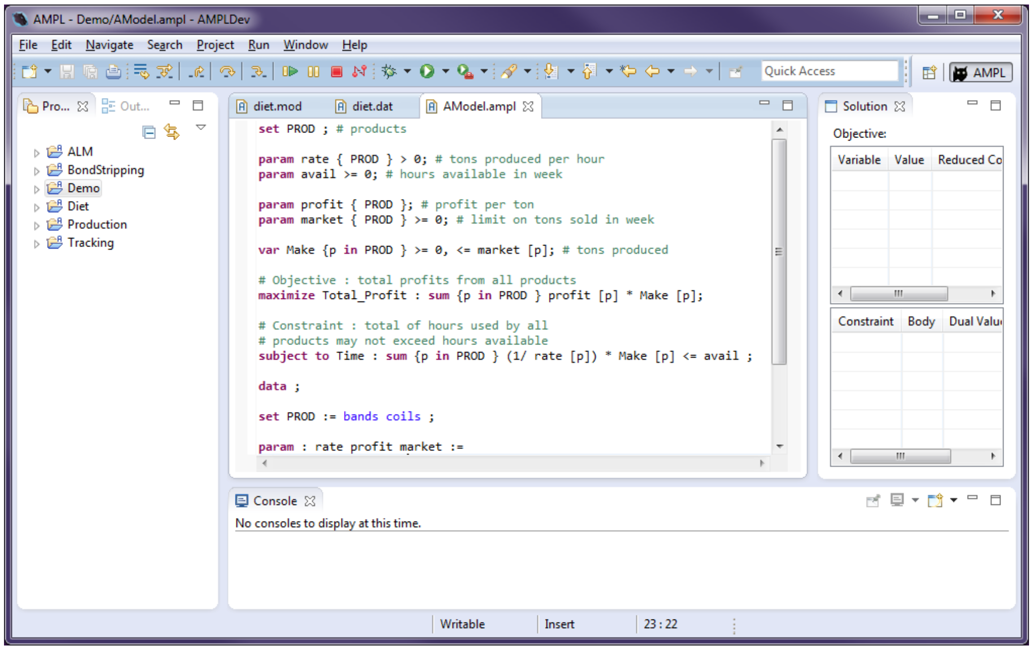
Figure 3.1: Workspace selection

TIP:

You can change the workspace any time by choosing **Switch workspace...** from the **File** menu.

The workbench is the interface that you will see when you start AMPLDev (3.2) . This is where you can manage the whole project life-cycle, from creation to the final product. A workbench window has menus and toolbars along with perspectives, where a perspective is a customized collection of views and editors based on the purpose of the perspective.

AMPLDev allows users to open multiple sessions of the workbench. Therefore while a window/-workbench is open, by running the eclipse executable file again, another window will open that can be used with a *different* workspace.



```
set PROD ; # products

param rate { PROD } > 0; # tons produced per hour
param avail >= 0; # hours available in week

param profit { PROD }; # profit per ton
param market { PROD } >= 0; # limit on tons sold in week

var Make {p in PROD } >= 0, <= market [p]; # tons produced

# Objective : total profits from all products
maximize Total_Profit : sum {p in PROD } profit [p] * Make [p];

# Constraint : total of hours used by all
# products may not exceed hours available
subject to Time : sum {p in PROD } (1/ rate [p]) * Make [p] <= avail ;

data ;

set PROD := bands coils ;

param : rate profit market :=
```

```
No consoles to display at this time.
```

Figure 3.2: AMPLDev workbench

3.2 Perspectives

Although perspectives is an advanced topic and meant particularly for people interested in using AMPLDev along with Eclipse, we have mentioned it here briefly since AMPLDev as a stand-alone does allow you to change perspectives.

A perspective is a customized layout of views that is set in an efficient and useful way for the purpose it was created. These views are set around an editor which is used to edit your files. You can move the views around to place them in a manner that you desire and these preferences will be saved for the next time you start a session.

There are two ways to get the AMPL perspective. The AMPLDev stand-alone generally starts with the AMPL perspective already but for some systems settings, it might not. Switching to the AMPL perspective can be achieved in one of the following ways:

- In **Window** menu, select **Open Perspective...**, choose **Other...** and then click on AMPL.
- Create an AMPL project; and if you are not in the AMPL perspective, AMPLDev will ask you to change to the AMPL perspective upon the creation of the project.

Although it is not necessary to use the AMPL perspective for using AMPL, the layout is what best suits for AMPL modelling.

There are also many AMPL-specific shortcuts that are available easily when in this perspective. They will be discussed in later chapters but one example is the toolbar which has buttons to create AMPL projects and files without going through the **File** menu.

3.3 Views

Views provide for smart ways to represent any kind of information; such as the Project Explorer which shows an easy-to-understand tree structure of all the parts of a project: files, folders etc, collectively known as resources. Views are also handy to navigate through large systems of information.

A perspective contains a set of views that open automatically when the perspective is selected. Although, you can also open views that are not part of the current perspective via the **Window** menu item and then selecting **Show View...**. As described previously, these additional views opened in a perspective are saved under the perspective.

Views can be moved around at any time by clicking and holding the left mouse button on its title bar; they can either be left detached or docked around the Workbench window. They can also be maximised by double-clicking on the title bar and doing it a second time will bring it back to its original size and position. To manage the screen space better, views can be stacked on top of each other and viewed by clicking on the tabs above the stacked views. The active view will have a highlighted tab.

Most views also have menus of their own which can be accessed by clicking the drop down arrow in the top right corner of the view. Sometimes they may even have their own toolbars.

The following sections describes the views that form the AMPL perspective.

Project Explorer

As explained in section 3.3, the Project Explorer view (figure 3.3) displays the projects in the workspace in a tree-structure. All the functions are the same as described in the Eclipse manual for the Project Explorer view with the addition of some AMPL specific functions, which are active in the AMPL perspective.

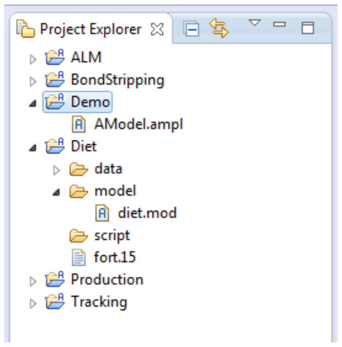


Figure 3.3: The Project Explorer view in the AMPL perspective

The context menu (pop-up menu on right-clicking, figure 3.4) allows you to add AMPL projects and files into the workspace, without going through the File menu wizards. The **Run As...** and the **Debug As...** also have an AMPL quick launch option, if the selection is an AMPL file. On clicking this quick launch, only the selected file is run by AMPL. In order to run multiple files, you must create a launch configuration (see section 5.2)

Console

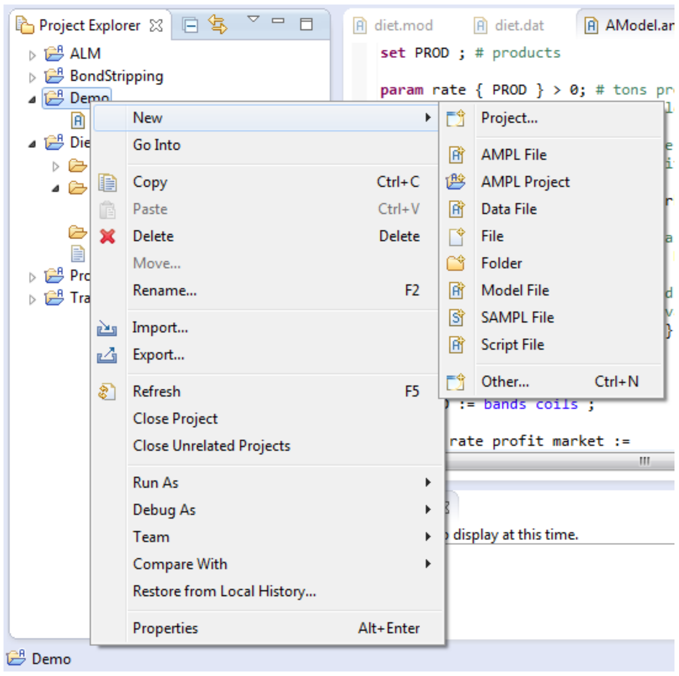
This view displays all the output and runtime errors from AMPL and SAMPL and allows user to enter commands and input data. For more details, see section 5.4.

Outline

This view (3.5) shows the list of model components, such as parameters, sets, variables, objectives and constraints, for the AMPL file currently open in the editor.

Solution

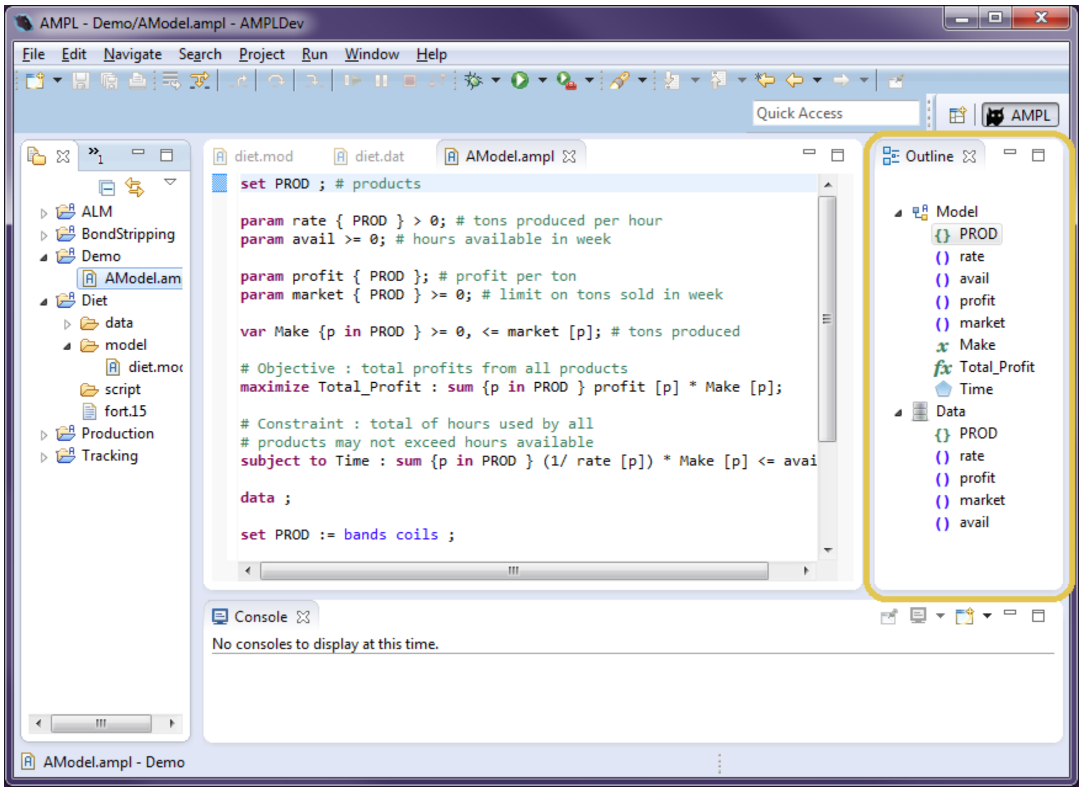
This view displays the solution report for the last run. For more details, see section 5.5.



```
set PROD ; # products

param rate { PROD } > 0; # tons pr
```

Figure 3.4: Context menu in Project Explorer



```
set PROD ; # products

param rate { PROD } > 0; # tons produced per hour
param avail >= 0; # hours available in week

param profit { PROD }; # profit per ton
param market { PROD } >= 0; # limit on tons sold in week

var Make {p in PROD } >= 0, <= market [p]; # tons produced

# Objective : total profits from all products
maximize Total_Profit : sum {p in PROD } profit [p] * Make [p];

# Constraint : total of hours used by all
# products may not exceed hours available
subject to Time : sum {p in PROD } (1/ rate [p]) * Make [p] <= avai

data ;

set PROD := bands coils ;
```

No consoles to display at this time.

Figure 3.5: Outline view

3.4 Editors

The model and data source files are opened, modified and saved using the editor. Different types of files have their own editors associated with them. When you double-click on any file in the views, it will automatically open it in the editor area with its respective editor. If there are multiple editors open at one time, then they are stacked in the editor area but can be separated in the same manner as how views are moved around except that they cannot be detached completely. When a file's name tab has an asterisk (*), it means there are still unsaved changes in that file.

TIP:

Eclipse has extensive documentation on the Eclipse Workbench which can be found online on their website at `http://help.eclipse.org/juno/index.jsp` (Eclipse Juno 4.2) and can also be accessed via the software's **Help** menu. This is the version AMPLDev is built on.

Part II

Modelling with AMPLDev

Chapter 4

Projects within AMPLDev

The following sections describe the different ways you can create a project, add files to it and use some of the available templates.

If you are starting Eclipse for the first time, follow the steps described in section 3.1 to start the Eclipse workbench.

4.1 Concept of a project

A project is a collection of source code and any supporting files such as documentation and data that may be arranged using folders in a way similar to organizing files in the filesystem. Projects are contained within workspaces described in section 3.1.

Some projects are associated with one primary language and occasionally some supporting languages; it is not possible to add unsupported language files in such a project. This does not apply to the workspace which can have a mix of various types of projects under it.

You can organise files and folders in any manner as you please under the project root.

4.2 Creating a new project

There are numerous ways to create a project. In general, you do not need to be in the AMPL perspective to create an AMPL project.

Using the File menu in a non-AMPL perspective

This is the most universal method of creating a new project.

1. Go to the **File** menu.
2. Select **New**.
3. Select **Project...**.
4. A **New Project** dialog will open which will ask you to select a project wizard. Choose AMPL Project under the AMPL category (see figure 4.1) and then click **Next**.
5. Give the project a name. You can either choose the default workspace location, or browse for another location. The **Create model, data and script directories** option as its name suggest creates the three respective folders in your project, if you check it (figure 4.2). You can always create the folders later (using the context menu of the project). These folders are independent of the files you put in them and is only for your personal organization. For example, you can put a .mod file in the data folder. They are not restrictive by type (see figure 3.3).

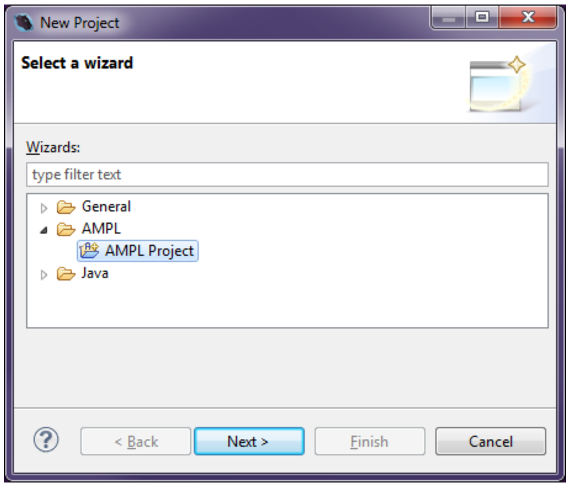


Figure 4.1: Eclipse New Project dialog

6. After clicking **Finish**, AMPLDev will ask you if you want to change to the AMPL perspective, *if* you are not in the AMPL persperctive.
7. The project has been created and you will be able to see it in Project Explorer.

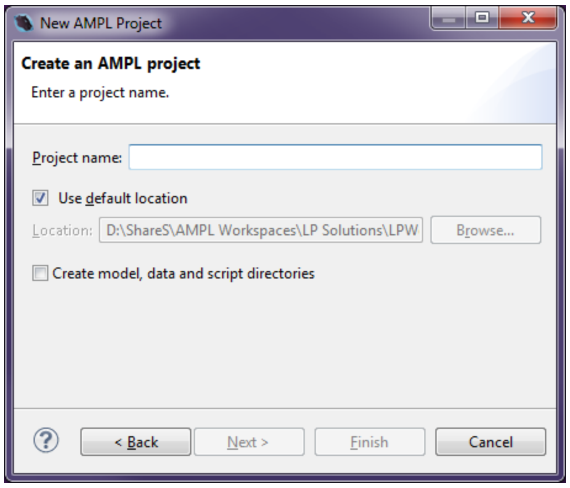
Using the File menu in the AMPL perspective

To create a project in the AMPL perspective is fairly straightforward.

1. Go to **File** menu.
2. Select **New**.
3. Choose **AMPL Project** and follow the steps as described in the previous section to create the project.

TIP:

There is the New button ( ) in the toolbar which is present in all perspectives by default. Clicking this button opens a dialog to create a new file or a project.



[✓] Use default location

Location: D:\ShareS\AMPL Workspaces\LP Solutions\LPW

[ ] Create model, data and script directories

Figure 4.2: AMPL Project wizard

Using the Project Explorer context menu in the AMPL perspective

The context menu is the menu that pops up with the right click (or command + click for a Mac). In the AMPL perspective when right-clicking in the Project Explorer view, there are shortcuts to create a project.

4.3 Creating and adding files to an AMPL project

Before describing the steps to create and add files, here is a brief description on the types of files supported by the AMPL plug-in for Eclipse.

Supported AMPL file extensions

The three traditional AMPL file extensions are .mod for model files, .dat for data files and .run for script files. They are all supported by AMPLDev.

In addition we have introduced a new file extension for AMPL files; you can now create an .ampl file which can also be used for AMPL code. Similarly, we have .sampl for SAMPL code. The .ampl files encompass all aspects of the .mod, .dat and .run file types; for example, you can have TestModel.ampl, TestData.ampl and TestScript.ampl which can have the same code as TestModel.mod, TestData.dat and TestScript.run. This will not make a difference while executing the files because the AMPL translator treats all the files in the same way regardless of their extension.

We recommend using the .ampl extension for new files because the .mod and .dat extensions are often associated with other programs. For example, the .mod extension is often recognized as an extension for audio files.

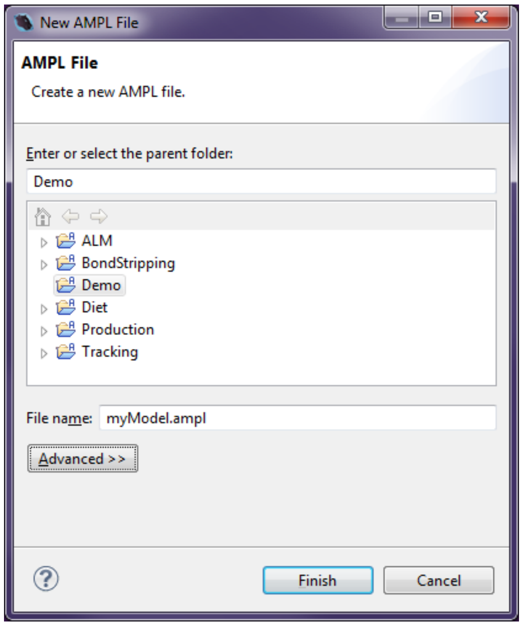
Create AMPL file

Most of the steps are the same as creating a project except this time we choose to create a new file.

1. Go to the **File** menu.
2. Select **New**.
3. If you do not see the options for all the different AMPL file types then select **Other...**. A wizard dialog will open, where you can choose the file you would like to add under the AMPL category and click **Next**.
4. The first field requires you to choose the project or a folder in the project in which the file must be placed. Then choose a filename. As you can see from figure 4.3, the title of the wizard is an **AMPL File**, so you do not need to enter the extension; this wizard will automatically create a .ampl file.
5. Once you have filled in the required information, AMPLDev will create a new file in the chosen project or project folder and open it in the editor.

Using other means

To create and add a file via the Project Explorer context menu in the AMPL perspective, is exactly the same as the project creation described in section 4.2. The New File shortcuts are in the same locations and you can follow the same steps for the file wizard as described in section 4.3.



New AMPL File

AMPL File

Create a new AMPL file.

Enter or select the parent folder: Demo

ALM
BondStripping
Demo
Diet
Production
Tracking

File name: myModel.ampl

Figure 4.3: New AMPL File wizard

Chapter 5

Running Models and Analysing Results

Running a model written in AMPL requires the use of AMPL compiler/interpreter which translates the model and combines the model and data into a machine-readable and solver-specific form, passes it to one of the many solvers supported by AMPL. It then gets the results and executes any script commands. In Eclipse, these runs are referred to as *launches*. It is possible to run a single file or multiple files in the order you want. There is an additional default launch available for projects that contain only one model and one data file.

Note: At this stage, debug functionalities are not implemented, so the **Run** and **Debug** buttons have the same effect for AMPL projects.

5.1 Single file launch

The previous chapter covered how to create a project and add files to it. When you do a single run, there is no relationship between the consecutive launches. For example, you cannot run a model file first and then a corresponding data file; each launch will be independent and Eclipse will not be able to associate the model with the data. (For doing that, refer to section 5.2).

The single file launch allows you to quickly run a file without creating a custom launch configuration. A simple AMPL model that will work in a single file launch, is given below:

```
set PROD;  # products

param rate {PROD} > 0;      # tons produced per hour
param avail >= 0;           # hours available in week

param profit {PROD};        # profit per ton
param market {PROD} >= 0;   # limit on tons sold in week

var Make {p in PROD} >= 0, <= market[p]; # tons produced

# Objective: total profits from all products
maximize Total_Profit: sum {p in PROD} profit[p] * Make[p];

# Constraint: total of hours used by all
# products may not exceed hours available
subject to Time: sum {p in PROD} (1/rate[p]) * Make[p] <= avail;

data;
```

```
set PROD := bands coils;

param:    rate  profit  market :=
  bands    200    25     6000
  coils    140    30     4000 ;

param avail := 40;

option solver cplexamp;
solve;
```

You may want to change the solver based on the ones installed on your machine, changing the statement 'option solver cplexamp;' accordingly.

There are two ways to run this code: context menu or launch toolbar. Both of them can be used in any perspective.

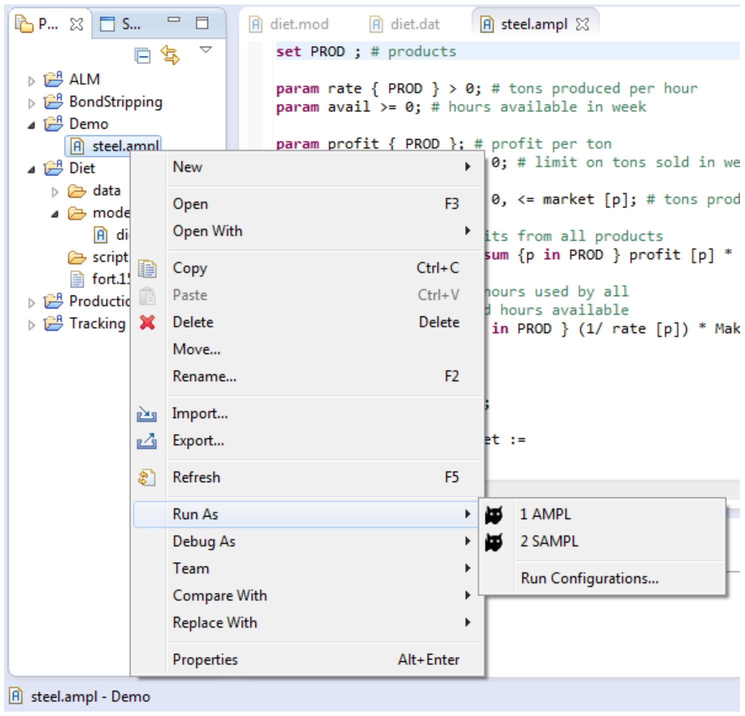
Context menu launch

This method is fairly straightforward.

1. In the Project Explorer, select the file you wish to execute. For this example, we have placed the code in the file steel.ampl.

2. Right click (or command + click for Mac) to open the context menu.

3. Select **Run As** or **Debug As** and you will see another menu pop-up (figure 5.1).



```
set PROD ; # products

param rate { PROD } > 0; # tons produced per hour
param avail >= 0; # hours available in week

param profit { PROD }; # profit per ton
0; # limit on tons sold in we
0, <= market [p]; # tons prod
its from all products
sum {p in PROD } profit [p] *
ours used by all
d hours available
in PROD } (1/ rate [p]) * Mak
;
et :=
```

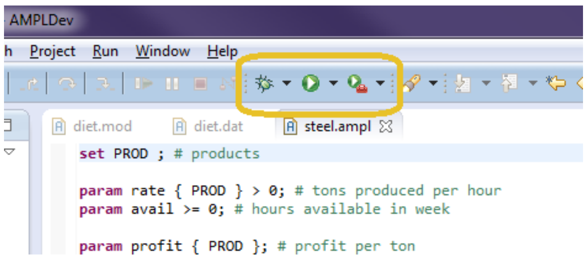
Figure 5.1: Context menu launch

4. This pop-up will have an option saying **1 AMPL**. By selecting this, AMPLDev will automatically create a launch configuration for this file and the file will be executed. The launch configuration will have the same name as the file.

Note: in the Project Explorer, if you right click on a project folder, the **Run As** or **Debug As** will show you an AMPL option, but this is only valid for a default launch, refer to section 5.3.

Launch toolbar

Once you have created a default configuration like in the previous defined steps, you can see these launches in the dropdown menus accessible with the **Run** () and **Debug** () toolbar buttons (figure 5.2).



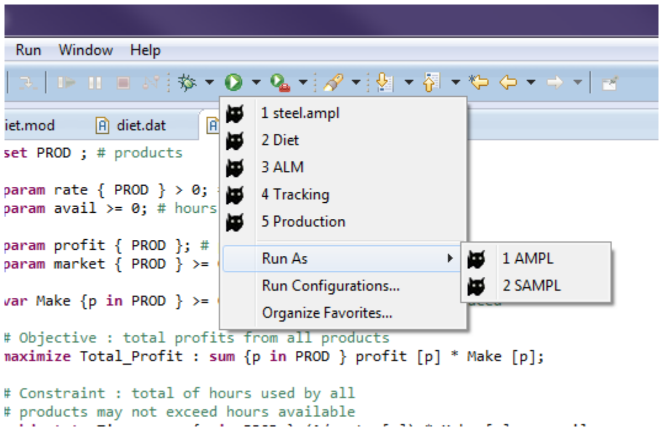
```
set PROD ; # products

param rate { PROD } > 0; # tons produced per hour
param avail >= 0; # hours available in week

param profit { PROD }; # profit per ton
```

Figure 5.2: Launch toolbar

When you click the down arrow next to these buttons, you will see the previous launched configuration for steel.ampl. You will also see the other method of launching the file which is the **Run As** or **Debug As** option, see figure 5.3.



```
set PROD ; # products

param rate { PROD } > 0;
param avail >= 0; # hours

param profit { PROD }; #
param market { PROD } >=

var Make {p in PROD } >=

# Objective : total profits from all products
maximize Total_Profit : sum {p in PROD } profit [p] * Make [p];

# Constraint : total of hours used by all
# products may not exceed hours available
```

Figure 5.3: Launch toolbar Run menu: **1 steel.ampl** is the previously created launch configuration

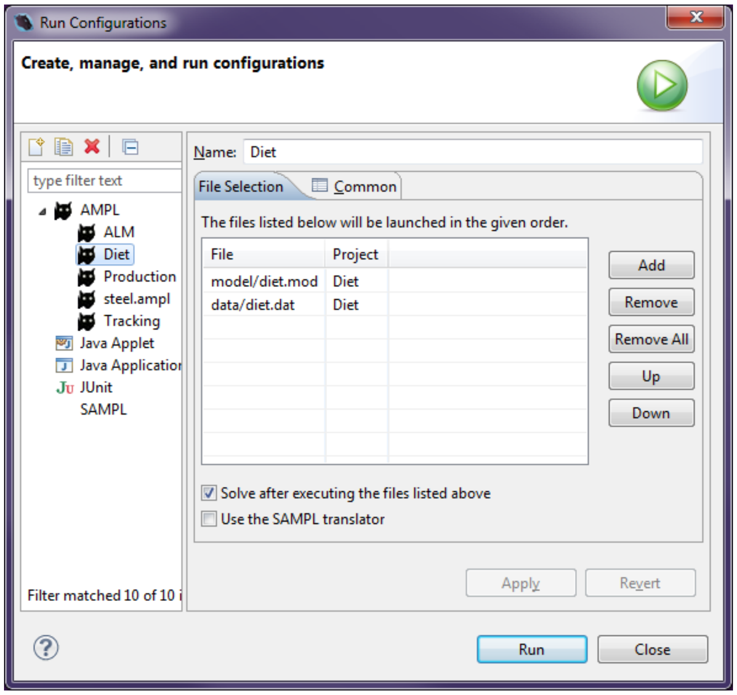
This shortcut creates a launch configuration for the current selection which means if there is a file selected in the editor or in the Project Explorer, it will launch that. In the case of steel.ampl, since there already is a launch configuration it will not create a new one. You can even choose **1 steel.ampl** if you would like to launch that file again.

As mentioned before, the launches are independent of the file extension; so whether you run an .ampl file, a .mod file or a .run file, AMPLDev will treat them the same way. The only exception is for the files with the .dat extensions which will be run in the data mode (as if the files contain the data statement at the beginning).

5.2 Multiple file launch

Having a single file execution is not always desirable; you may like to separate your model from the data or have multiple models for some data and it is not ideal to put all the code in one file. For this reason, there is a multiple file launch.

In order to run multiple files, you will need to use the launch configuration dialog. This can be found in either the context menu or launch toolbar shortcuts mentioned in the single file launch (section 5.1). Looking at figure 5.1 and 5.3, you will see the option **Run Configurations...** (or **Debug Configurations...**). by selecting that you will open the launch configuration dialog (figure 5.4).



Name: Diet

The files listed below will be launched in the given order.

File	Project
model/diet.mod	Diet
data/diet.dat	Diet

☑ Solve after executing the files listed above
☐ Use the SAMPL translator

Figure 5.4: Launch configuration dialog

Double click **AMPL/SAMPL** in the tree on the left, or right click it and select **New**. This will create a new configuration and will open a clean **File Selection** tab on the right. You can give the configuration a name to help differentiate between various configurations.

TIP:

You will notice that the configuration created earlier **steel.ampl** is also present on the left hand side. By selecting that, you can add more files to it later on, if you wish to expand that code into multiple files. So the single file launch can later be adapted for multiple files.

The File Selection tab

In this tab, you can add the files you wish to run together. Start by clicking the **Add** button to add some files. This will open a window with a list of all the files in their respective projects and folders (figure 5.5). If you click the folder, all the files in it will be selected.

After selecting the files, they will be automatically added to the table so you can now select the order in which you would like to execute them by selecting the file in the table and clicking the **Up** and **Down** buttons to move the selection (figure 5.4). Clicking **Apply** just saves the changes, although Eclipse will always ask you to save if there are any unsaved changes before you launch.

You will also notice two checkboxes under the table. The first one (**Solve after executing the files listed above**) is the equivalent of writing 'solve;' at the end of your AMPL code. If checked AMPL will automatically solve after running the files selected in the table. This will also populate the Solution view (see section 5.5). The second one (**Use the SAMPL translator**) specifies whether to use the SAMPL translator instead of the AMPL one.

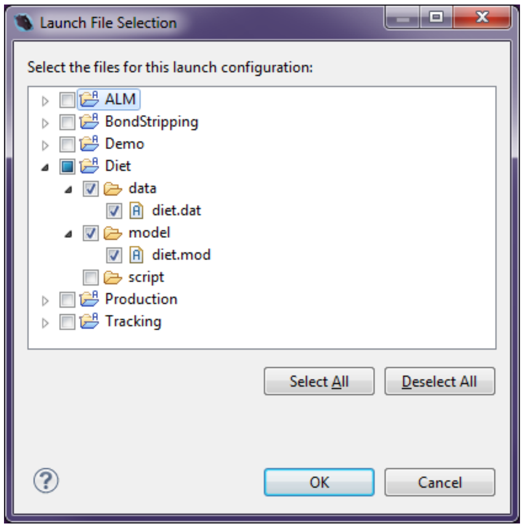


Figure 5.5: Selecting the files for the launch configuration

Once you are ready to run the set of files, hit the **Run** or **Debug** button and your files will be executed.

5.3 Default launch

This launch is available for projects that contain only one model (.mod) and one data (.dat) file. This is to avoid the hassle of creating a launch configuration (as it comes under the multiple file launch), for a simple project structure.

You do not need to do anything special for using this type of launch; AMPLDev automatically checks whether the launch conditions of one model and one data file are satisfied in the project and creates a default configuration. You use the same shortcuts as described in single file launch (section 5.1).

The main difference is that when you have a one model and one data project you can even launch it from the project folder's context menu via **Run As** or **Debug As**. If these conditions are not satisfied then, AMPLDev will ask you to create a launch configuration.

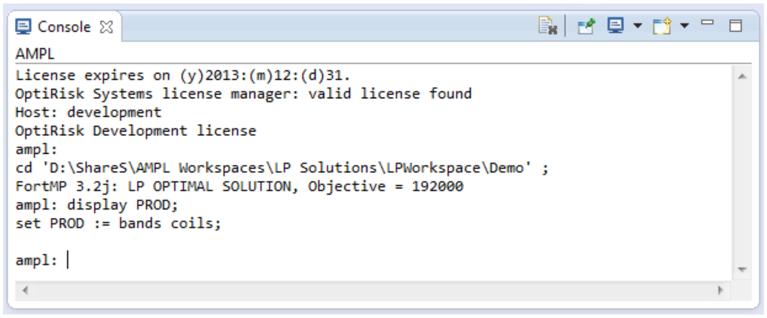
5.4 Viewing output and errors

While a run is in progress the output and errors if there are any will show up in the Console view. This view is generally located in the bottom panel of AMPLDev. If it is not visible you can view it via the **Window** menu, selecting **Show View** and then choose the **Console** view.

The console is interactive; you can enter and execute arbitrary commands there (see figure 5.6). The text entered by the user is colored differently from the output text.

In the case of an error the detailed information and a hyperlink to the error location if available is shown in the console (figure 5.7). In this example, there is a syntax error because 'cost' has been mistyped. The location hyperlink contains the name of the file, the line number and the offset from the beginning of the file. Clicking the hyperlink opens the file in the editor and highlights the line presumably containing the error. Note that in some cases the error may be located in the code above this line, for example in the case of missing semicolon.

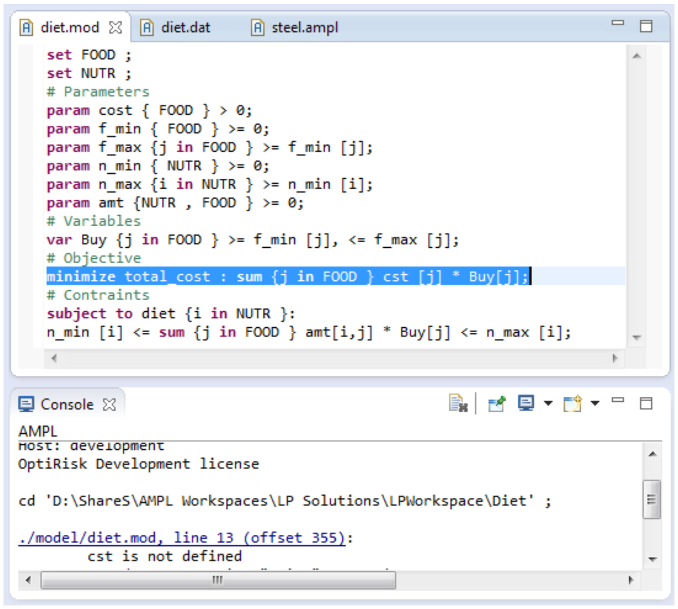
You can use the 'display' and 'print' commands to output various information from your AMPL code to the console.



```
License expires on (y)2013:(m)12:(d)31.
OptiRisk Systems license manager: valid license found
Host: development
OptiRisk Development license
ampl:
cd 'D:\ShareS\AMPL Workspaces\LP Solutions\LPWorkspace\Demo' ;
FortMP 3.2j: LP OPTIMAL SOLUTION, Objective = 192000
ampl: display PROD;
set PROD := bands coils;

ampl:
```

Figure 5.6: Console view



```
set FOOD ;
set NUTR ;
# Parameters
param cost { FOOD } > 0;
param f_min { FOOD } >= 0;
param f_max {j in FOOD } >= f_min [j];
param n_min { NUTR } >= 0;
param n_max {i in NUTR } >= n_min [i];
param amt {NUTR , FOOD } >= 0;
# Variables
var Buy {j in FOOD } >= f_min [j], <= f_max [j];
# Objective
minimize total_cost : sum {j in FOOD } cst [j] * Buy[j];
# Contraints
subject to diet {i in NUTR }:
n_min [i] <= sum {j in FOOD } amt[i,j] * Buy[j] <= n_max [i];
```

```
Host: development
OptiRisk Development license

cd 'D:\ShareS\AMPL Workspaces\LP Solutions\LPWorkspace\Diet' ;

./model/diet.mod, line 13 (offset 355):
        cst is not defined
```

Figure 5.7: Error shown in the Console view

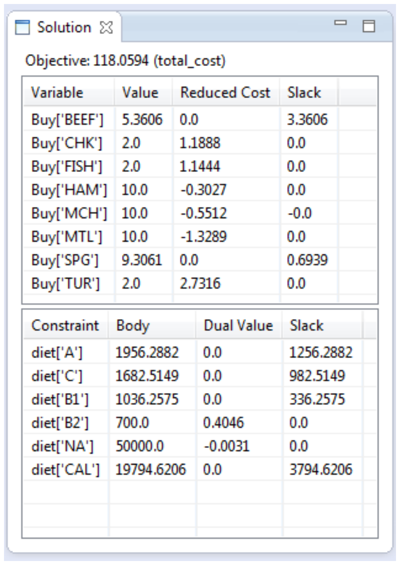
5.5 Viewing the solution

The solution report will be displayed in the Solution view (generally located in the right panel of the AMPL perspective. If you cannot see it then you can open it via **Window** → **Show View...**). This view is only populated if the **Solve after executing the files listed above** option is checked for the launch configuration (see figure 5.4).

There are three sections in the Solution view, refer to figure 5.8. The first part states the **Objective** value and the name of the objective function in brackets.

The first table describes the variables and their corresponding values. **Reduced Cost** is the amount by which the objective value would improve if the value of the corresponding variable is increased by one. **Slack** is the distance from the bound.

The second table describes the constraints where **Body** is the body of the constraint, **Dual Value** is an indication of how much the objective value can be increased if the constraint is relaxed by one unit and lastly **Slack** is the distance between the body of the constraint and its bound.



Objective: 118.0594 (total_cost)

Variable	Value	Reduced Cost	Slack
Buy['BEEF']	5.3606	0.0	3.3606
Buy['CHK']	2.0	1.1888	0.0
Buy['FISH']	2.0	1.1444	0.0
Buy['HAM']	10.0	-0.3027	0.0
Buy['MCH']	10.0	-0.5512	-0.0
Buy['MTL']	10.0	-1.3289	0.0
Buy['SPG']	9.3061	0.0	0.6939
Buy['TUR']	2.0	2.7316	0.0

Constraint	Body	Dual Value	Slack
diet['A']	1956.2882	0.0	1256.2882
diet['C']	1682.5149	0.0	982.5149
diet['B1']	1036.2575	0.0	336.2575
diet['B2']	700.0	0.4046	0.0
diet['NA']	50000.0	-0.0031	0.0
diet['CAL']	19794.6206	0.0	3794.6206

Figure 5.8: Solution view

IMPORTANT NOTE:

For the complete AMPLDev manual which contains the AMPL and SAMPL reference parts outlined in the contents, please download the AMPLDev software or email us at sales@optirisk-systems.com

The complete manual contains examples along with their corresponding source code in AMPL/SAMPL and their algebraic formulations.