

A short introduction to modeling with Xpress

The software that will be used in this course will be Xpress, a powerful solver for both linear and non-linear problems. We will use the Mosel language with IVE, an interface that is very friendly to use and very easy to learn. Although during the course we will use the full version of the software, there is a [free student version](#) with limited numbers of constraints and variables.

1. Our first model

Let us consider the following blending problem: A chemical firm can produce two products, A and B, from combining three elements: iron, lead, and tin. What is the blending that maximizes the benefit? Assume that all the production is sold.

Resources	Units used per kg of product		Available units
	Product A	Product B	
Iron	7	4	56
Lead	3	5	45
Tin	4	3	48
Net benefit (£/kg)	10	8	

First, we define the **decision variables**:

- x_1 = “kilograms of product A”.
- x_2 = “kilograms of product B”.

Second, we state the **objective function**: $10x_1 + 8x_2$.

Finally, we write the constraints:

- **Constraint on the iron availability**: $7x_1 + 4x_2 \leq 56$.
- **Constraint on the lead availability**: $3x_1 + 5x_2 \leq 45$.
- **Constraint on the tin availability**: $4x_1 + 3x_2 \leq 48$.
- **Variable non-negativity**: $x_1, x_2 \geq 0$.

Therefore, the model is:

$$\left\{ \begin{array}{ll} \text{Max.} & 10x_1 + 8x_2 \\ \text{s.t.} & 7x_1 + 4x_2 \leq 56, \\ & 3x_1 + 5x_2 \leq 45, \\ & 4x_1 + 3x_2 \leq 48, \\ & x_1, x_2 \geq 0. \end{array} \right.$$

Let us see now how to write it and solve it with Xpress.

2. Writing our first model

When we start using Xpress, the screen will look like in Figure 1.

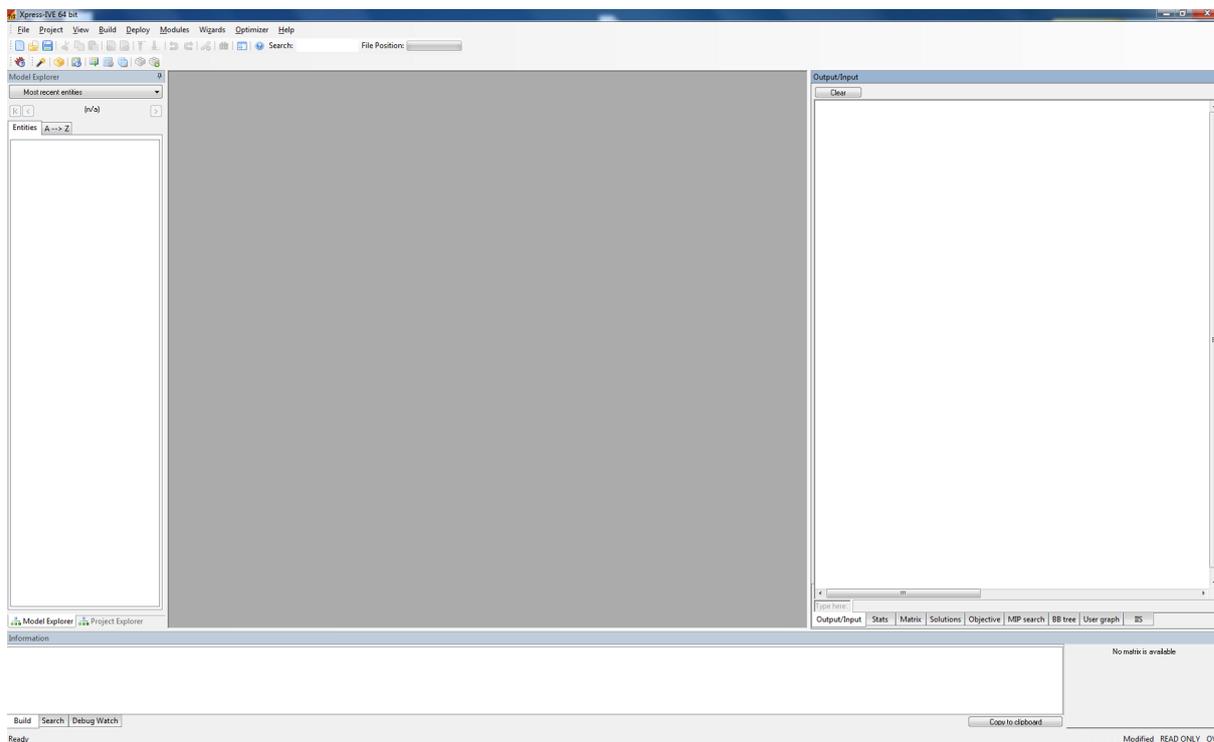


Figure 1: First screen.

Once we create a new file (File → New), we start to write the model.

1. First, there is a line giving a name to the model:

```
model blending
```

We have given it the name **blending**, but we can use any name we like.

2. Now, we add a line that calls the solver:

```
uses "mmxprs"
```

3. Next, we declare the variables of the model:

```
declarations  
  x1,x2:mpvar  
end-declarations
```

A decision variable is an object of type **mpvar**.

4. Now, we write the objective function. We have also added a comment line (begins with the symbol “!”) which is not read at all, but it helps to have the code better organized.

```
! Objective function
obj:= 10*x1 + 8*x2
```

5. Then, we define the constraints, one per line.

```
! Constraints
7*x1 + 4*x2 <= 56
3*x1 + 5*x2 <= 45
4*x1 + 3*x2 <= 48
```

Note that we write the multiplication symbol “*” between the coefficient and the variable (unless it is 0 or 1). Remember also that the constraints that we will use are only of the type “≤”, “≥”, and “=”. We will *never* have inequalities with “<” or “>”.

An important remark is that we do not need to add explicitly the non-negativity constraints. Unless stated otherwise, Xpress assumes that all the variables are non-negative and continuous.

6. Finally, we tell the solver that we are going to maximize the objective function:

```
maximize(obj)
```

7. The last line says that the model is completed:

```
end-model
```

This model can be compiled (which automatically saves the file) with F7. Or we can run it with F6 (which saves and compiles before that). Alternatively, you can use the buttons shown in Figure 2.

Once we have run the model, we can see some relevant information in the output window. Figure 3 shows one of the several tabs. In this one, you can see the algorithm used (simplex dual), the number of iterations, the final objective value, the status (the solution is optimal) and the time used to solve the model. In other tabs you can find even more information (value of the decision variables, slack of each constraint, dual variables, etc.).

Actually, we may prefer to print some of these values in the main output window. Add the following lines before the last line of the model and run the model again:

```
writeln("x1: ",getsol(x1))
writeln("x2: ",getsol(x2))
writeln("Objective value: ",getobjval)
```

You can find the whole model in the file `blending.basic.mos`.

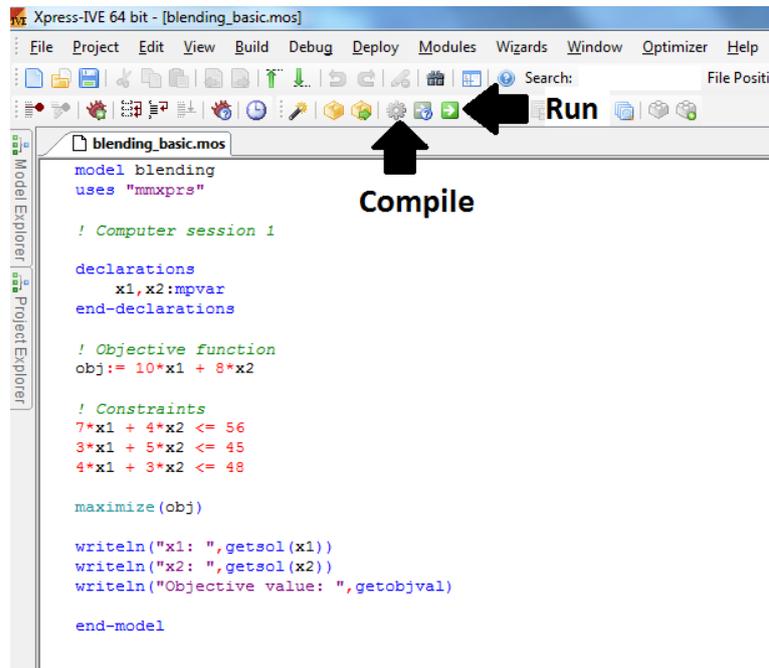


Figure 2: Blending model.

Stats			
Matrix:		Presolved:	
Rows(constraints):	3	Rows(constraints):	3
Columns(variables):	2	Columns(variables):	2
Nonzero elements:	6	Nonzero elements:	6
Global entities:	0	Global entities:	0
Sets:	0	Sets:	0
Set members:	0	Set members:	0
Overall status: Finished LP relaxation.			
LP relaxation:			
Algorithm:	Simplex dual		
Simplex iterations:	2		
Objective:	94.6087		
Status:	LP Optimal		
Time:	0.0s		

Figure 3: Output for the blending model.