

---

## **AIMMS Tutorial for Beginners - Building the Model**

This file contains only one chapter of the book. For a free download of the complete book in pdf format, please visit [www.aimms.com](http://www.aimms.com)

Copyright © 1993–2011 by Paragon Decision Technology B.V. All rights reserved.

Paragon Decision Technology B.V.	Paragon Decision Technology Inc.	Paragon Decision Technology Pte.
Schipholweg 1	500 108th Avenue NE	Ltd.
2034 LS Haarlem	Ste. # 1085	80 Raffles Place
The Netherlands	Bellevue, WA 98004	UOB Plaza 1, Level 36-01
Tel.: +31 23 5511512	USA	Singapore 048624
Fax: +31 23 5511517	Tel.: +1 425 458 4024	Tel.: +65 9640 4182
	Fax: +1 425 458 4025	

Email: [info@aimms.com](mailto:info@aimms.com)  
WWW: [www.aimms.com](http://www.aimms.com)

ISBN xx-xxxxxx-x-x

AIMMS is a registered trademark of Paragon Decision Technology B.V. IBM ILOG CPLEX and sc CPLEX is a registered trademark of IBM Corporation. GUROBI is a registered trademark of Gurobi Optimization, Inc. KNITRO is a registered trademark of Ziena Optimization, Inc. XPRESS-MP is a registered trademark of FICO Fair Isaac Corporation. MOSEK is a registered trademark of Mosek ApS. WINDOWS and EXCEL are registered trademarks of Microsoft Corporation. T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, and A<sub>M</sub>S-~~L~~T<sub>E</sub>X are trademarks of the American Mathematical Society. LUCIDA is a registered trademark of Bigelow & Holmes Inc. ACROBAT is a registered trademark of Adobe Systems Inc. Other brands and their products are trademarks of their respective holders.

Information in this document is subject to change without notice and does not represent a commitment on the part of Paragon Decision Technology B.V. The software described in this document is furnished under a license agreement and may only be used and copied in accordance with the terms of the agreement. The documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Paragon Decision Technology B.V.

**Paragon Decision Technology B.V. makes no representation or warranty with respect to the adequacy of this documentation or the programs which it describes for any particular purpose or with respect to its adequacy to produce any particular result. In no event shall Paragon Decision Technology B.V., its employees, its contractors or the authors of this documentation be liable for special, direct, indirect or consequential damages, losses, costs, charges, claims, demands, or claims for lost profits, fees or expenses of any nature or kind.**

**In addition to the foregoing, users should recognize that all complex software systems and their documentation contain errors and omissions. The authors, Paragon Decision Technology B.V. and its employees, and its contractors shall not be responsible under any circumstances for providing information or corrections to errors and omissions discovered at any time in this book or the software it describes, whether or not they are aware of the errors or omissions. The authors, Paragon Decision Technology B.V. and its employees, and its contractors do not recommend the use of the software described in this book for applications in which errors or omissions could threaten life, injury or significant loss.**

This documentation was typeset by Paragon Decision Technology B.V. using L<sup>A</sup>T<sub>E</sub>X and the LUCIDA font family.

# Chapter 3



## Building the Model

---

### 3.1 Starting a new project


Assuming that AIMMS 3 has already been installed on your machine. If there is an AIMMS 3 shortcut on your desktop, double click it to start AIMMS 3, otherwise execute the following sequence of actions to start AIMMS:

*Starting AIMMS*


- ▶ press the **Start** button  on the taskbar,
- ▶ go to the **Programs** submenu, and
- ▶ select and click on the AIMMS icon  to start AIMMS.

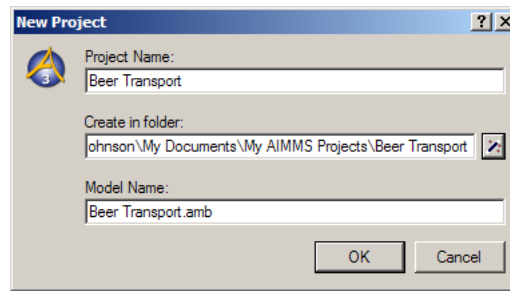
Next, you will see the AIMMS splash screen. Once AIMMS has started, the splash screen will disappear and the AIMMS window will open. Should you encounter the AIMMS **Tip of the Day** dialog box, close it, because it is not relevant to you at this point.

*Specifying a project name*

Press the **New Project** button , which is located in the leftmost position on the AIMMS toolbar. The dialog box shown in Figure 3.1 will then appear, requiring you to take the following actions:

*Creating a new project from within AIMMS*

- ▶ specify 'Beer Transport' as the project name, and
- ▶ press the wizard button  to select the folder for your AIMMS projects if the default folder  
'... \My Documents \My AIMMS Projects \Beer Transport'  
is not desired, and
- ▶ press the **OK** button.

Figure 3.1: The **New Project** wizard

Next, the AIMMS **Model Explorer** and the AIMMS **Page Manager** will be automatically opened. We will look at the AIMMS **Model Explorer** first.

---

## 3.2 The Model Explorer

When opened for the first time, the AIMMS **Model Explorer** will display the initial model tree shown in Figure 3.2. In this initial model tree you will see *Initial model tree*

- a single *declaration section*, where you can store the declarations used in your model,
- the predefined procedure *MainInitialization*, which is not relevant for this tutorial,
- the predefined procedure *MainExecution*, where you will put the execution statement necessary to solve the mathematical program, and
- the predefined procedure *MainTermination*, which is again not relevant for this tutorial.

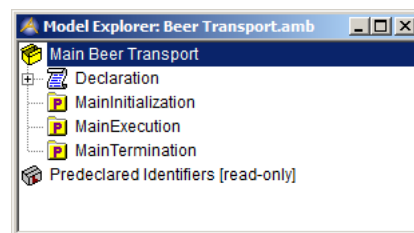









Figure 3.2: The initial model tree


### 3.3 Entering sets and indices

The declaration of model identifiers requires you to first ‘open’ the declaration section. You can do this either by clicking the  icon or by double-clicking on the scroll icon . Note that double-clicking on the name of the declaration section instead of on its icon will open the attribute form of the declaration section and will therefore, at this point, not lead to the desired result. After opening the declaration section the standard identifier buttons      on the toolbar will be enabled.

*Opening the declaration section*

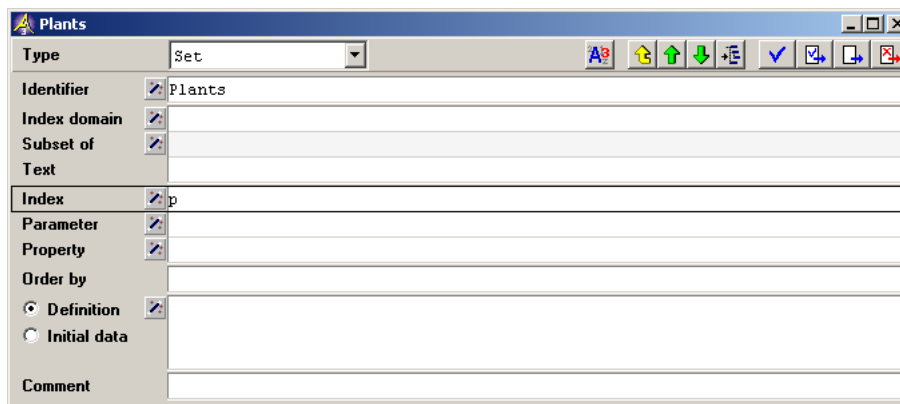
To create a set of plants you should take the following actions:

*Creating the set ‘Plants’*

- ▶ press the **Set** button  to create a new set identifier in the model tree,
- ▶ specify ‘Plants’ as the name of the set, and
- ▶ press the *Enter* key to register the name.

Next, you need to declare the index  $p$  as an attribute of the set ‘Plants’. You can open the attribute form by double-clicking on the node ‘Plants’ in the model tree. The resulting initial attribute form of the set ‘Plants’ is shown in Figure 3.3.

*Opening its attribute form*




Plants	
Type	Set
Identifier	Plants
Index domain	
Subset of	
Text	
Index	p
Parameter	
Property	
Order by	
<input checked="" type="radio"/> Definition <input type="radio"/> Initial data	
Comment	

Figure 3.3: The initial attribute form of the set ‘Plants’

To declare the index  $p$  as an attribute of the set ‘Plants’, execute the following sequence of actions:

*Declaring the index  $p$*

- ▶ move the mouse cursor to the ‘Index’ attribute field, and click in the (empty) edit field,
- ▶ enter the letter  $p$ , and
- ▶ complete the attribute form by pressing the **Check, Commit and Close** button .

Next, create the set 'Customers' with associated index  $c$  in exactly the same way as you created the set 'Plants' with index domain  $p$ . Figure 3.4 contains the resulting model tree.

*Creating the set 'Customers'*

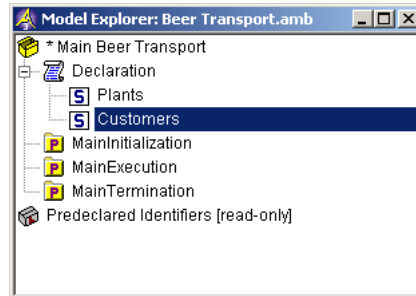



Figure 3.4: An intermediate model tree

The asterisk on the left of the project name indicates that additions to your project have not yet been saved to disk. To save your work, please press the **Save Project** button  on the toolbar.

*Saving your changes*


### 3.4 Entering parameters and variables

In this section you will declare the parameters and variables that are needed in your model. The sets 'Plants' and 'Customers' and their associated indices will be used to specify the index domain for the parameters and variables.

*Domain specification*

The declaration of a parameter is similar to the declaration of a set. To enter the parameter 'Supply( $p$ )', you should execute the following actions:

*Creating the parameter 'Supply'*

- ▶ press the parameter button  on the toolbar to create a new parameter in the model tree,
- ▶ specify 'Supply( $p$ )' as the name of the parameter, and
- ▶ press the *Enter* key to register the name.

Note that parentheses are used to add the index domain  $p$  to the identifier 'Supply'.

The parameter 'Demand( $c$ )' can be added in the same way. Should you make a mistake in entering the information, then you can always re-edit a name field by a single mouse click within the field.

*Creating the parameter 'Demand'*

The last model parameter 'UnitTransportCost' is a two-dimensional parameter with index domain  $(p, c)$ . After entering 'UnitTransportCost(p,c)', the resulting model tree should be the same as in Figure 3.5.

*Creating the parameter 'UnitTransport-Cost'*

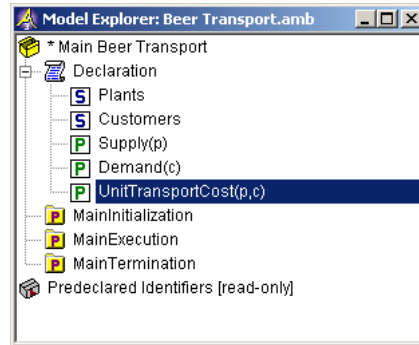




Figure 3.5: An intermediate model tree

Declaring a variable is similar to declaring a parameter.

*Creating the variable 'Transport'*

- ▶ press the variable button  on the toolbar to create a new variable in the model tree,
- ▶ specify 'Transport(p,c)' as the name of the variable, and
- ▶ press the *Enter* key to register the variable.

After opening the attribute form of the variable by double-clicking on the node 'Transport' in the model tree, press the wizard button  in front of the 'Range' attribute field. The resulting dialog box provides the opportunity to specify the range of values that the variable 'Transport' is allowed to take. In this case, select the 'Standard Range', then select 'nonnegative', and finally press the *OK* button (see Figure 3.6).

*Specifying range attribute*

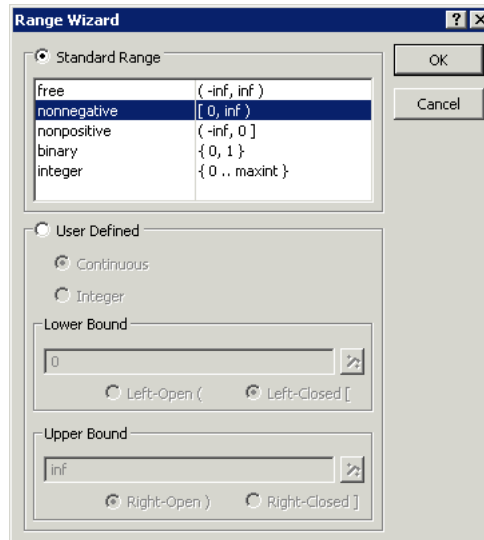



Figure 3.6: The AIMMS range wizard

It should be clear by now how to create the variable ‘TotalTransportCost’. This variable will be used to specify the objective function. After entering its name, open the attribute form. There is no need to specify the range attribute, since the default range ‘free’ will suffice. You are now ready to enter the following definition of this particular variable:

```
sum[ (p,c), UnitTransportCost(p,c) * Transport(p,c) ]
```

Simply enter the above definition in the ‘Definition’ attribute field. You could type the entire sentence yourself, but you can also let AIMMS do some of the typing for you. Considering the parameter ‘UnitTransportCost(p,c)’, the following two support features are quite useful.

- Type the letter *u* or *U*, and press the *Ctrl-Spacebar* combination for automatic name completion.
- Another option available to you is to drag the name ‘UnitTransportCost(p,c)’ from the model tree to the edit field of the ‘Definition’ attribute.

The attribute form should now have the same content as shown in Figure 3.7. By pressing the **Check, Commit and Close** button , you can verify whether AIMMS will accept the definition you entered.



*Creating the variable ‘TotalTransportCost’*

*Specifying definition attribute*

TotalTransportCost	
Type	Variable
Identifier	TotalTransportCost
Index domain	
Text	
Range	free
Unit	
Default	
Property	
Nonvar status	
Definition	$\text{sum} [ (p,c) , \text{UnitTransportCost}(p,c) * \text{Transport}(p,c) ]$
Comment	

Figure 3.7: The completed attribute form for the variable 'TotalTransportCost'

### 3.5 Entering constraints and the mathematical program

Creating the supply and demand constraints, each with their own definition, requires the same actions as creating a variable with a definition (as you just completed). The only difference is that you must use the  button instead of the  button. The following two forms should be the result of your efforts.



*The supply and demand constraints*

SupplyRestriction	
Type	Constraint
Identifier	SupplyRestriction
Index domain	p
Text	
Unit	
Property	
Definition	$\text{sum} [ c, \text{Transport}(p,c) ] \leq \text{Supply}(p)$
Comment	

Figure 3.8: The completed attribute form for the constraint 'SupplyRestriction'

DemandRequirement	
Type	Constraint
Identifier	DemandRequirement
Index domain	c
Text	
Unit	
Property	
Definition	$\text{sum}[ p, \text{Transport}(p,c) ] \geq \text{Demand}(c)$
Comment	

Figure 3.9: The completed attribute form for the constraint 'DemandRequirement'

A mathematical program, unlike sets, parameters, variables and constraints, does not have a special button on the toolbar. By using the identifier button , you obtain access to all the other types of AIMMS identifiers. After pressing this button, select the 'Mathematical Program' entry alongside the , press the OK button, and enter 'LeastCostTransportPlan' as the name of the mathematical program.

*Creating the mathematical program*

To complete the attribute form of the mathematical program as illustrated in Figure 3.10. Among the attributes, AIMMS has automatically filled **Direction**, **Constraints**, **Variables** and **Type** attributes with default values and there is no need to change them for this project. You only need to fill the **Objective** attribute.

*Specifying its attributes*

LeastCostTransportPlan	
Type	Mathematical Progr
Identifier	LeastCostTransportPlan
Objective	TotalTransportCost
Direction	minimize
Constraints	AllConstraints
Variables	AllVariables
Text	
Type	Automatic
Violation penalty	
Comment	

Figure 3.10: The completed attribute form of the mathematical program

The **Objective** attribute wizard requires you to select a scalar variable. In the identifier selection wizard (see Figure 3.11), simply select the scalar variable 'TotalTransportCost', and press the *Finish* button.

*Selecting the objective*

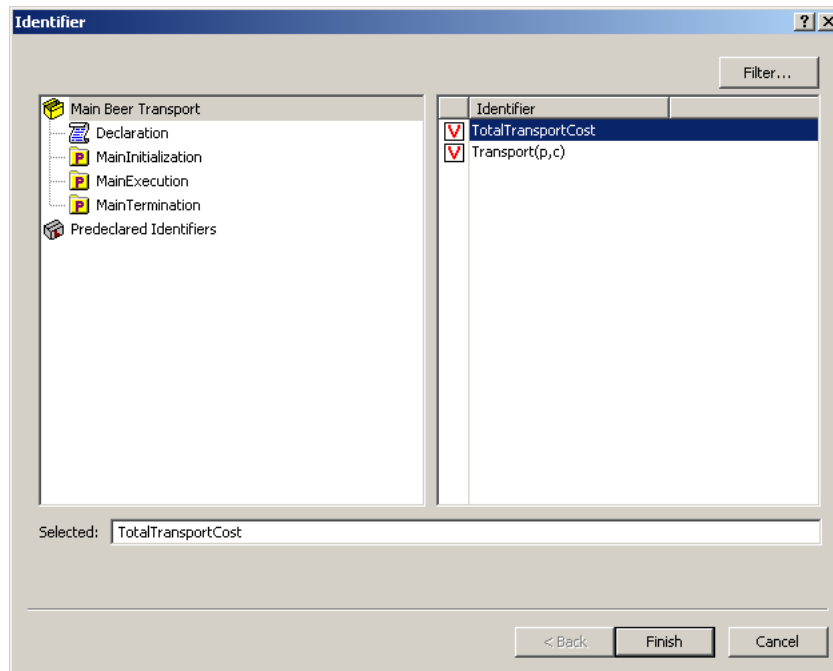



Figure 3.11: The identifier selection wizard

---

### 3.6 Viewing the identifiers

You have now entered and declared all model identifiers. The resulting model tree is shown in Figure 3.12. By pressing the *F5* key you can instantly check the validity of your model. You will only receive a message in the event of an error. Once the validity of your model has been verified, you should save your work by pressing the **Save Project** button .

*Checking your model*

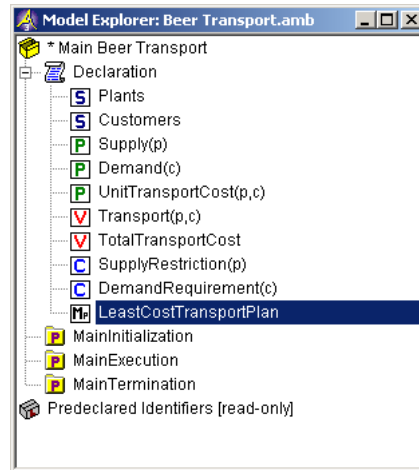


Figure 3.12: The final model tree

Even though the Model Explorer is a convenient medium with which to build and inspect your model, AIMMS provides two other ways to view your model.

If you would like to see a text(ASCII) representation of the model, you can do the following:

*View text model*

- ▶ select node(s) in AIMMS **Model Explorer**,
- ▶ go to the **View - Text Representation** menu and execute the **Selected Part(s)** command(see Figure 3.13).

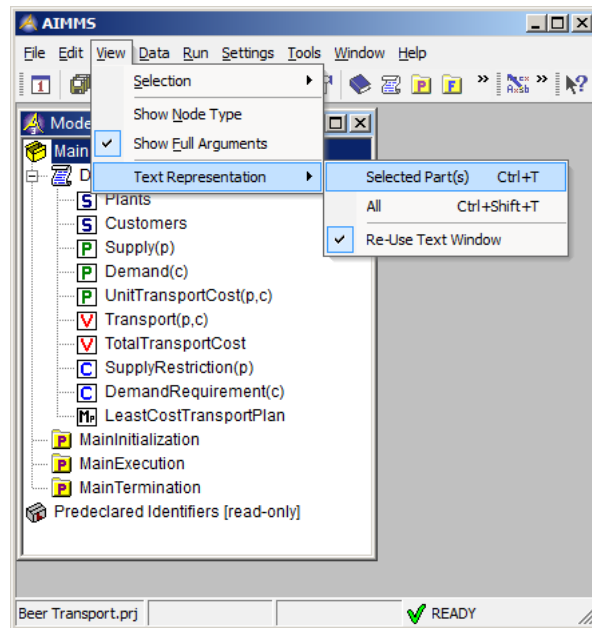
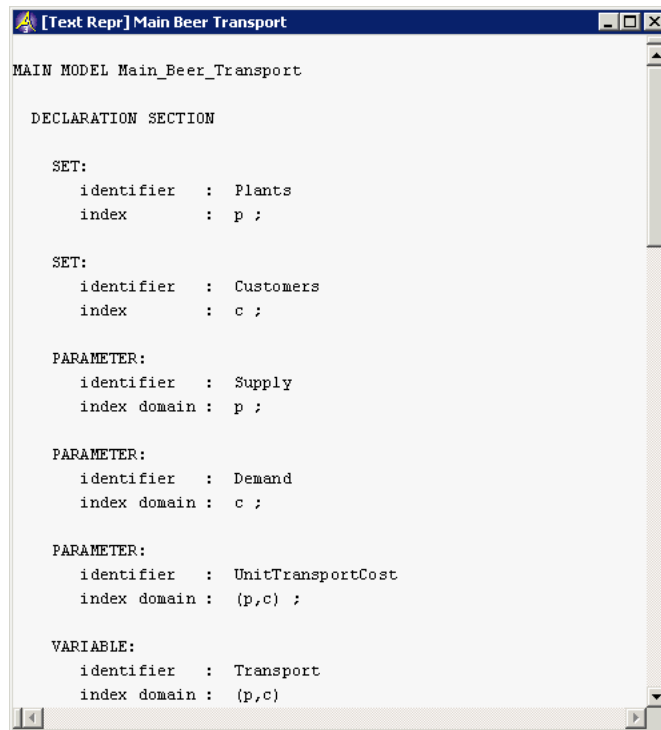


Figure 3.13: View text model

The text model provides a simple overview of selected identifiers. For instance, Figure 3.14 shows the text model when the root node **Main Beer Transport** is selected.



```

MAIN MODEL Main_Beer_Transport

DECLARATION SECTION

SET:
  identifier : Plants
  index      : p ;

SET:
  identifier : Customers
  index      : c ;

PARAMETER:
  identifier : Supply
  index domain : p ;

PARAMETER:
  identifier : Demand
  index domain : c ;

PARAMETER:
  identifier : UnitTransportCost
  index domain : (p,c) ;

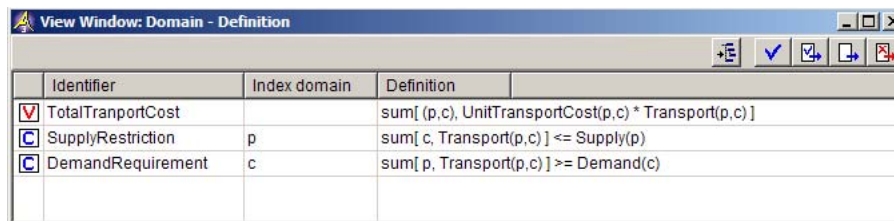
VARIABLE:
  identifier : Transport
  index domain : (p,c)

```

Figure 3.14: text model

Another way to inspect the model is by AIMMS **Identifier Selector**. This allows you to view several identifiers with similar properties at the same time. In this tutorial you will encounter one such example of a predefined view, namely all identifiers with a definition (see Figure 3.15). AIMMS allows you to make your own views as you desire.

*Identifier  
overviews*




Identifier	Index domain	Definition
<input checked="" type="checkbox"/> TotalTransportCost		sum[ (p,c), UnitTransportCost(p,c) * Transport(p,c) ]
<input checked="" type="checkbox"/> SupplyRestriction	p	sum[ c, Transport(p,c) ] <= Supply(p)
<input checked="" type="checkbox"/> DemandRequirement	c	sum[ p, Transport(p,c) ] >= Demand(c)

Figure 3.15: View window with identifier definitions

You can create a view window by executing the following steps:

*Creating a view*

- ▶ press the **Identifier Selector** button  on the toolbar,
- ▶ select the 'Identifiers with Definition' node, and
- ▶ use the right mouse and select the **Open With...** command from the popup menu (see Figure 3.16).

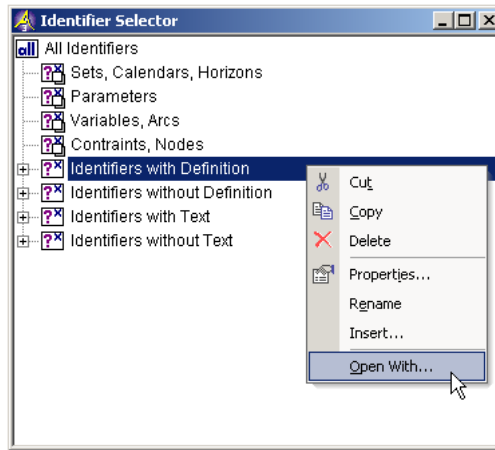


Figure 3.16: Identifier Selector window

For the selected identifiers the view can be constructed as follows:

- ▶ select the 'Domain - Definition' entry from the **View Manager** window (see Figure 3.17), and
- ▶ press the *Open* button to obtain the overall view.

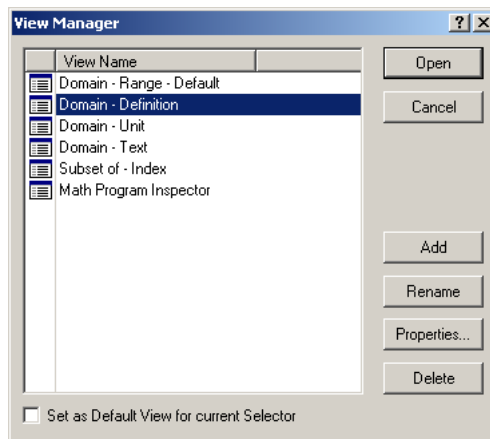


Figure 3.17: View Manager window