
AIMMS User's Guide - Viewing Identifier Selections

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 or order your hard-copy at www.lulu.com/aimms.

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
WWW: www.aimms.com

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. $\text{T}_{\text{E}}\text{X}$, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, and $\text{A}_{\text{M}}\text{S}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$ 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 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ and the LUCIDA font family.

Chapter 7

Viewing Identifier Selections

Although the **Model Explorer** is a very convenient tool to organize all the information in your model, it does not allow you to obtain a simultaneous overview of a group of identifiers that share certain aspects of your model. By mutual comparison of important attributes (such as the definition), such overviews may help you to further structure and edit the contents of your model, or to discover oversights in a formulation.

*Identifier
overviews*

To assist you in creating overviews that can help you analyze the interrelationships between identifiers in your model, AIMMS offers the **Identifier Selector** tool and **View** windows. This chapter helps you understand how to create meaningful identifier selections with the **Identifier Selector**, and how to display such selections using different views.

This chapter

7.1 Creating identifier selections

When you are developing or managing a large and complicated model, you sometimes may need an overview of all identifiers that have some sort of similarity. For example, it may be important to have a simultaneous view of

*Select by
similarity*

- all the constraints in a model,
- all variables with a definition,
- all parameters using a certain domain index, or
- all identifiers that cover a specific part of your model.

In AIMMS, you can create a list of such identifiers using the configurable **Identifier Selector** tool. This tool helps you to create a selection of identifiers according to a set of one or more criteria of varying natures. You can let AIMMS create a once only selection directly in the **Model Explorer**, or create a compound selection in the **Identifier Selector**, which allows you to intersect or unite multiple selections.

*Identifier
selections*

If you need a selection only once, then you can create it directly in the **Model Explorer** by

Creating once only selections

- either manually selecting one or more nodes in the tree, or
- using the **View-Selection** menu to create a custom selection based on one or more of the conditional selection criteria offered by AIMMS (explained below).

In both cases, the resulting list of selected identifiers will be highlighted in the model tree. If you like, you can narrow down or extend the selection by applying one or more subsequent conditional selections to the existing selection.

If you need a specific selection more than once, then you can create it in the **Identifier Selector** tool. The **Identifier Selector** consists of a tree in which each node contains one of the three types of identifier selectors described below. Figure 7.1 illustrates an example selector tree.

The Identifier Selector

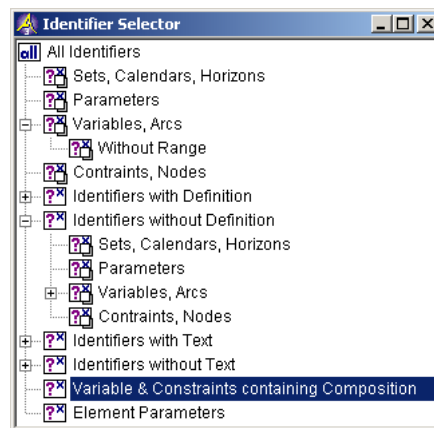










Figure 7.1: The selector tree

In the **Identifier Selector** tool, you can add nodes corresponding to several types of identifier selectors:

Selector types

- a *node-based selector* , where all the identifiers below one or more user-selected nodes in the model tree are added to the selection,
- a *conditional selector* , where the list of identifiers is created dynamically on identifier type and/or the contents of one of their respective attributes,
- a *set-dependent selector* , where the list of identifiers is created dynamically based on a specific set in either the domain or range of identifiers, or
- a *type-based selector* , where the list of identifiers consists of all variables of a certain type (e.g. free, nonnegative, binary) or all constraints of a certain type (\leq , $=$ or \geq).

While the above four selectors allow you to define selections based on a symbolic criteria the five types of identifier selectors below allow you to specify selections based on individual criteria. The main purpose of these selector is to define selections that can be used in the Math Program Inspector (see Chapter 9).

- an *element-dependent selector* , where the list of individual identifiers is created dynamically based of the occurrence of one or more specific elements in the domain,
- a *scale-based selector* , where the list of identifiers is built up from all variables and constraints for which the ratio between the largest absolute value and the smallest absolute value in the corresponding row or column of the matrix exceeds a given value,
- a *status-based selector* , where the list of identifiers is built up from all variables and constraints for which the solution satisfies some property (e.g. feasible, basic, at bound), or
- a *value-based selector* , where the list of identifiers is built up from all variables and constraints for which the level, bound, marginal, or bound violation value satisfy satisfy some property.

Through the **View-Selection** menu in the **Model Explorer** you can only create a new, or refine an existing, selection using a *conditional selector*.

To create a selector, AIMMS offers special dialog boxes which let you specify the criteria on which to select. As an example the dialog box for creating a conditional selector is illustrated in Figure 7.2. In it, you can select (by double

Selection dialog box

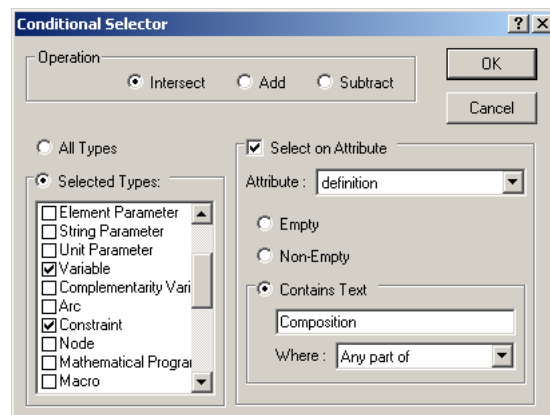


Figure 7.2: The **Conditional Selector** dialog box

clicking) one or more identifier types that you want to be part of the selection and filter on specific attributes that should be either empty, nonempty, or should contain a particular string.

The tree structure in the **Identifier Selector** defines combinations of selectors by applying one of the set operators *union*, *difference* or *intersection* with respect to the identifier selection represented by the parent node. The root of the tree always consists of the fixed selection of all model identifiers. For each subsequent child node you have to indicate whether the node should add identifiers to the parent selection, should remove identifiers from the parent selection, or should consider the intersection of the identifiers associated with the current and the parent selection. Thus, you can quickly compose identifier selections that satisfy multiple selection criteria. The type of set operation applied is indicated by the icon of each node in the identifier selector.

Compound selections

In the **Model Explorer**, the *union*, *difference* and *intersection* operations apply to the identifier selection that is currently highlighted in the model tree. You can use them to add identifiers to the current selection, to remove identifiers from the current selection, or filter the current selection by means of an additional criterion.

Refining model tree selections

The list of identifiers that results from a (compound) identifier selector can be used in one of the following ways:

Using selections

- you can display the identifiers in a **View** window of your choice (explained in the next section),
- you can restrict the set of variables and constraints initially displayed in the **Math Program Inspector** (see Chapter 9),
- by dragging and dropping a selector into the **Model Explorer**, the corresponding identifiers will be highlighted in the model tree, or
- you can use the selector in the definition of a *case type* or *data category* in the **Data Management Setup** tool (see Chapter 18).

The drag-and-drop features of AIMMS make it very easy to fill a **View** window with identifiers from either the model tree, the **Identifier Selector** or other **View** windows. If you drag-and-drop a selection into any other AIMMS window, AIMMS will interpret this as a special search action to highlight all occurrences of the selected identifiers as follows:

Advanced drag and drop

- in the *model tree* all identifiers in the selection will be highlighted,
- in the *page* or *template tree* all pages that contain reference to the identifiers in the selection will be highlighted,
- in an end-user *page*, in edit mode, all objects that contain references to the identifiers will be selected,
- in the *menu builder tree*, AIMMS will highlight all menu items that reference one or more identifiers in the selection, and
- in the *data management setup tree*, all data categories and case types that contain one or more identifiers in the selection will be highlighted.

In addition, AIMMS also supports the 'drag-and-drop-search' action in a **View** window by pressing both the **Shift** and **Control** key during the drop operation.

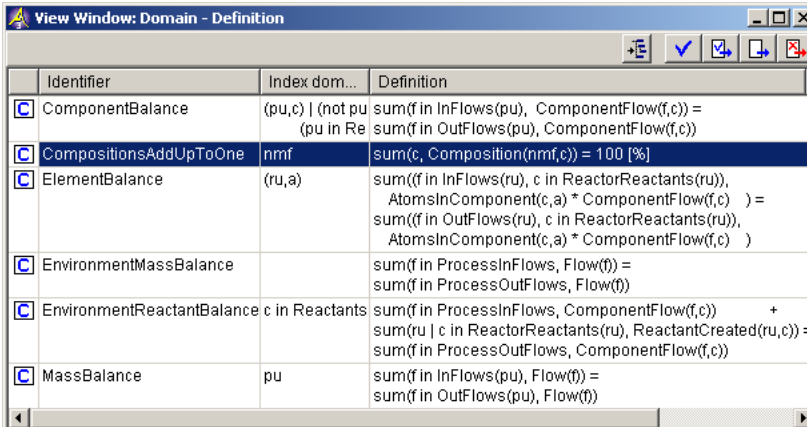
7.2 Viewing identifier selections

After you have created an identifier selection, in either the **Model Explorer** or in the **Identifier Selector**, you may want to compare or simultaneously edit multiple attributes of the identifiers in the selection. In general, sequential or simultaneous, opening of all the corresponding single attribute forms is impractical or unacceptable for such a task. To assist, AIMMS offers special identifier **View** windows.

Overview of attributes

A **View** window allows you to view one or more attributes simultaneously for a number of identifiers. Such a **View** window is presented in the form of a table, where each row represents a single identifier and each column corresponds to a specific attribute. The first column is always reserved for the identifier name. An example of an identifier **View** window is given in Figure 7.3.

Identifier views



Identifier	Index dom...	Definition
<input checked="" type="checkbox"/> ComponentBalance	(pu,c) (not pu (pu in Re	sum(f in InFlows(pu), ComponentFlow(f,c)) = sum(f in OutFlows(pu), ComponentFlow(f,c))
<input checked="" type="checkbox"/> CompositionsAddUpToOne	nrmf	sum(c, Composition(nmf,c)) = 100 [%]
<input checked="" type="checkbox"/> ElementBalance	(ru,a)	sum((f in InFlows(ru), c in ReactorReactants(ru)), AtomsInComponent(c,a) * ComponentFlow(f,c)) = sum((f in OutFlows(ru), c in ReactorReactants(ru)), AtomsInComponent(c,a) * ComponentFlow(f,c))
<input checked="" type="checkbox"/> EnvironmentMassBalance		sum(f in ProcessInFlows, Flow(f)) = sum(f in ProcessOutFlows, Flow(f))
<input checked="" type="checkbox"/> EnvironmentReactantBalance	c in Reactants	sum(f in ProcessInFlows, ComponentFlow(f,c)) + sum(ru c in ReactorReactants(ru), ReactantCreated(ru,c)) = sum(f in ProcessOutFlows, ComponentFlow(f,c))
<input checked="" type="checkbox"/> MassBalance	pu	sum(f in InFlows(pu), Flow(f)) = sum(f in OutFlows(pu), Flow(f))

Figure 7.3: Example of a **View** window

In addition to simply viewing the identifier content in a **View** window, you can also use it to edit individual entries. To edit a particular attribute of an identifier you can just click on the relevant position in the **View** window and modify the attribute value. This can be convenient, for instance, when you want to add descriptive text to all identifiers for which no text has yet been provided, or when you want to make consistent changes to units for a particular selection of identifiers. As in a single attribute form, the changes that you make are not committed in the model source until you use one of the special compile buttons at the top right of the window (see also Section 5.3).

Editing in a View window

Using the **Edit-Open with** menu, or the **Open with** item in the right- mouse pop-up menu, you can open a particular **View** window for any identifier selection in the model explorer or in the identifier selector. Selecting the **Open with** menu will open the **View Manager** dialog box as displayed in Figure 7.4. In the **View Manager** you must select one of the available *view window defini-*

Opening a View window

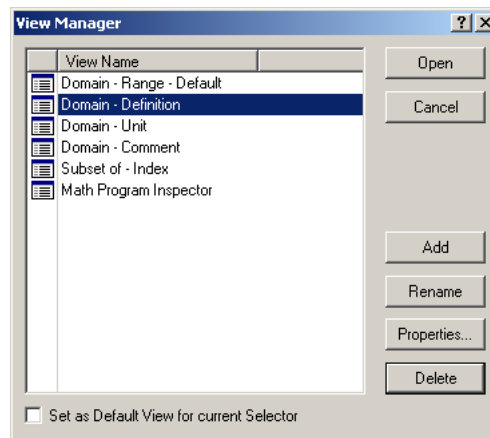


Figure 7.4: The **View Manager** dialog box

tions, with which to view the given identifier selection. For every new project, the **View Manager** will automatically contain a number of basic view window definitions that can be used to display the most common combinations of identifier attributes.

Using the **Add**, **Delete** and **Properties** buttons in the **View Manager**, you can add or delete view window definitions to the list of available definitions, or modify the contents of existing definitions. For every view window definition that you add to the list or want to modify, AIMMS will open the **View Definition Properties** dialog box as illustrated in Figure 7.5. With this dialog box you can

Creating a view window definition

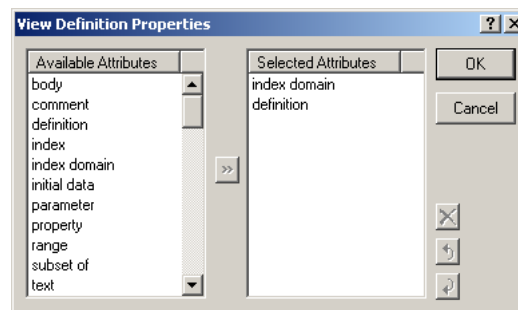


Figure 7.5: **View Definition Properties** dialog box

add or remove attributes from the list of attributes that will be shown in the

View window, or change the order in which the particular attributes are shown.

After opening a **View** window, with the contents of a particular identifier selection, you can add new identifiers to it by dragging and dropping other identifier selections from either the **Model Explorer** or the **Identifier Selector**. Using the **Edit-Delete** menu or the **Del** key, on the other hand, you can delete any subselection of identifiers from the **View** window. At any time you can save the modified identifier selection as a new node in the identifier selector tree through the **View-Selection-Save** menu.

*Changing the
View window
contents*

Besides selecting individual identifiers from the model tree, you can also select whole groups of identifiers by selecting their parent node. For example, if you drag-and-drop an entire declaration section into a **View** window, all the identifiers contained in that section will be added to the view.

*Selecting
identifier groups*

As can be seen at the bottom of the **View Manager** dialog box in Figure 7.4, it is possible to associate a default view definition with every selector in the **Identifier Selector**. As a consequence, whenever you double-click on such an identifier selector node, AIMMS will immediately open a default **View** window with the current contents of that selection.

*Specifying a
default view*