

---

## **AIMMS User's Guide - Pages and Page Objects**

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 © 1993–2018 by AIMMS B.V. All rights reserved.

AIMMS B.V.  
Diakenhuisweg 29-35  
2033 AP Haarlem  
The Netherlands  
Tel.: +31 23 5511512

AIMMS Inc.  
11711 SE 8th Street  
Suite 303  
Bellevue, WA 98005  
USA  
Tel.: +1 425 458 4024

AIMMS Pte. Ltd.  
55 Market Street #10-00  
Singapore 048941  
Tel.: +65 6521 2827

AIMMS  
SOHO Fuxing Plaza No.388  
Building D-71, Level 3  
Madang Road, Huangpu District  
Shanghai 200025  
China  
Tel.: ++86 21 5309 8733

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

AIMMS is a registered trademark of AIMMS B.V. IBM ILOG CPLEX and CPLEX is a registered trademark of IBM Corporation. GUROBI is a registered trademark of Gurobi Optimization, Inc. KNITRO is a registered trademark of Artelys. WINDOWS and EXCEL are registered trademarks of Microsoft Corporation.  $\text{\TeX}$ ,  $\text{\LaTeX}$ , and  $\text{\AMS-}\text{\TeX}$  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 AIMMS 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 AIMMS B.V.

**AIMMS 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 AIMMS 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, AIMMS 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, AIMMS 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 AIMMS B.V. using  $\text{\TeX}$  and the LUCIDA font family.

## **Part III**

---

# **Creating an End-User Interface**

## Chapter 10

### Pages and Page Objects

After you have created a model in AIMMS to represent and solve a particular problem, you may want to move on to the next step: creating a graphical end-user interface around the model. In this way, you and your end-users are freed from having to enter (or alter) the model data in text or database tables. Instead, they can make the necessary modifications in a graphical environment that best suits the purposes of your model. Similarly, using the advanced graphical objects available in AIMMS (such as the Gantt chart and network flow object), you can present your model results in an intuitive manner, which will help your end-users interpret a solution quickly and easily.

*Creating an end-user interface*

This chapter gives you an overview of the possibilities that AIMMS offers you for creating a complete model-based end-user application. It describes pages, which are the basic medium in AIMMS for displaying model input and output in a graphical manner. In addition, the chapter illustrates how page objects (which provide a graphical display of one or more identifiers in your model) can be created and linked together.

*This chapter*

---

#### 10.1 Introduction

A *page* is a window in which the data of an AIMMS model is presented in a graphical manner. Pages are the main component of an end-user interface for a model-based decision support application. An example of an end-user page is given in Figure 10.1. The page shown here provides a comprehensive graphical overview of the results of an optimization model by means of a *network flow object* in which flows which require attention are colored red. By clicking on a particular flow in the network object, additional information about that flow is shown in the tables on the left of the page.

*What is a page?*

Pages are fully designed by application developers for use by the end-users of an application. Thus, you, as a developer, can decide at what position in the interface particular model data should be presented to the end-user, and in which format. In addition, by automatically executing procedures when opening or closing a page or when modifying data, you can make sure that

*Page design*

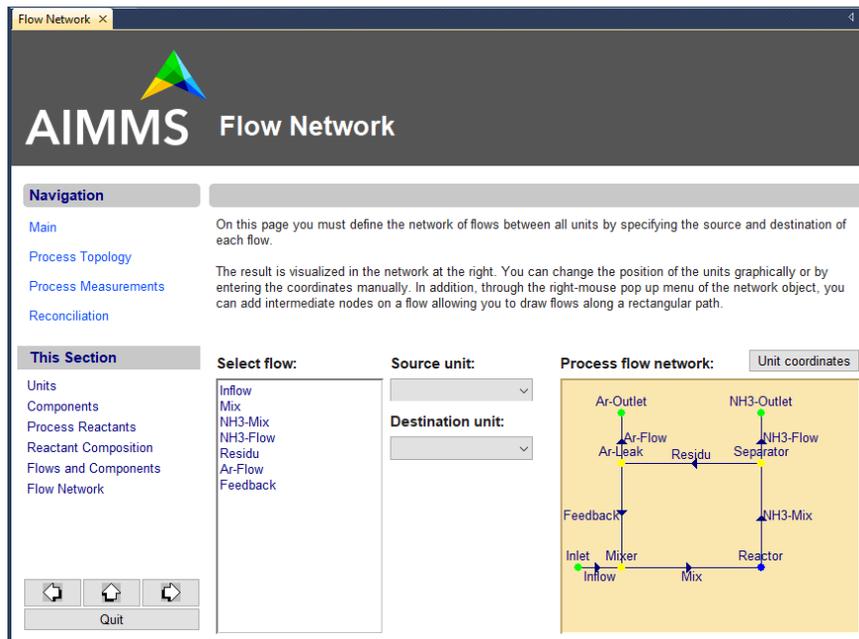


Figure 10.1: Example of a end-user page

all the necessary computations are performed before certain model results are displayed.

## 10.2 Creating pages

Creating an end-user page is as easy as adding a new node to the page tree in the **Page Manager** (see Chapter 12). Figure 10.2 illustrates the page tree associated with the example application used throughout this guide. As all the trees in the AIMMS modeling tools work alike, you can use any of the methods described in Section 4.3 to add a new page node to the page tree.

*Creating pages*

In addition to inserting a new empty page into the page tree, you can also copy an existing page or an entire subtree of pages, by either a simple cut, copy and paste or a drag-and-drop action in the tree (see Section 4.3). All copied pages will have the same content as their originals.

*Copying pages*

The node name of every page (as displayed in the page tree) is the unique name or description by which the page is identified in the system. When you add new pages to the tree, AIMMS will name these *Page 1*, *Page 2*, etc. You can change this name using the standard methods for changing names of tree nodes as described in Section 4.3.

*Page name*

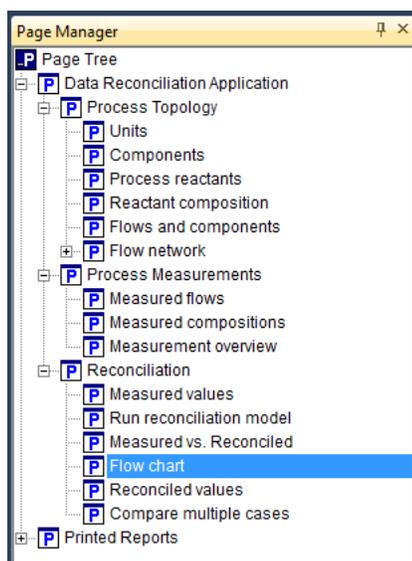


Figure 10.2: Example of a page tree

By default, the node name is the title that will be displayed in the frame of the page window when the page is opened. In the page **Properties** dialog box (see Section 11.2) you can, however, specify a different page title to be displayed, which can either be a constant string or a reference to a string parameter in the model. The latter is useful, for instance, if you intend to set up an end-user interface in multiple languages.

*Page title*

Its position in the page tree determines the navigational properties of the page. It will determine how any button with references to the next or previous page, or any navigation object or menu linked to the page, will behave. These navigational aspects of the **Page Manager** tool are discussed in more detail in Chapter 12.

*Position in page tree*

Every page that you add to the page tree, is also automatically added to the template tree in the **Template Manager**. By moving the page to a different position in the template tree, the page automatically inherits all the properties such as page size or background, and all objects specified on the template pages hierarchically above it. The **Template Manager** and the use of templates is explained in full detail in Chapter 12.

*Using templates*


---

### 10.3 Adding page objects

All visible components on a page are instances of the collection of *page objects* as offered by AIMMS. Such page objects are mostly used to visualize the

*Page objects*

input and output data of your model in various ways. They also include simple drawing objects, such as lines and circles, and buttons for navigation and execution.

Before you can add page objects to a page, the page must be in *edit mode*. When you open a page using the **Page Manager**, it is opened in *user mode* by default. When you want to open a page in edit mode from the **Page Manager**, you can do so using the right mouse pop-up menu. If a page is already opened in user mode, you can reopen it in edit mode using the  button on the page toolbar. When you open the page from the **Template Manager**, it is opened in edit mode by default.

*Edit mode*

AIMMS provides the most common graphical data objects such as

- row-oriented composite tables,
- 2-dimensional tables,
- pivot tables,
- graphs, and
- charts.

*Common data objects*

These objects can be used both for displaying and for modifying the data in your model. The data displayed in such objects are always directly linked to one or more identifiers in your model.

Placing a data object onto a page can be done without any programming. The following straightforward actions are required:

*Adding an object*

- select the type of the graphical object to be displayed,
- drag a rectangle onto the page of the intended size of the object, and
- choose the identifier in the model holding the data that you want to display.

You can select the object type that you want to add to the page from the **Object** menu. Alternatively, you can select any of the most common object types using the **Page Edit** toolbar, as depicted in Figure 10.3. If you move the cursor to one

*Selecting the object type*



Figure 10.3: The **Page Edit** toolbar

of the buttons of the toolbar, a tooltip will appear. After you have selected an object type, the page cursor will change to a cross allowing you to drag the rectangle in which the object will be contained. Figure 10.4 illustrates such a rectangle just prior to linking it to one or more AIMMS identifiers.

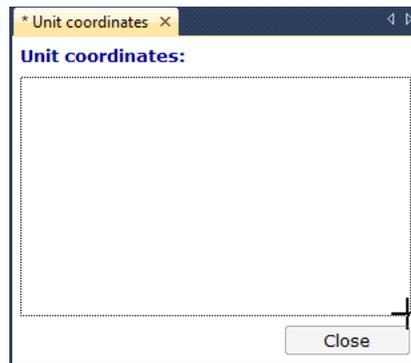


Figure 10.4: Drawing an object region

In order to let you drag object regions in an aligned manner, AIMMS allows you to associate a grid with a page, and align object regions to that grid automatically via the **View** menu. Alternatively, you may align objects later on, or make them the same size via the **Edit-Alignment** menu (see Section 11.1).

*Object grid and alignment*

After you have indicated the object region, you must select an identifier to be associated with that object. To support you in this task AIMMS provides an **Identifier Selection** dialog box as illustrated in Figure 10.5. You can select any single identifier from the list on the right.

*Selecting an identifier ...*

Additional help is offered for models with many identifiers. By selecting a subtree of the model tree on the left-hand side of the dialog box, you can narrow down the selection of identifiers on the right-hand side to those which are declared within the selected subtree. With the **Filter...** button you can narrow the selection down even more, by only displaying those identifier types that you are interested in.

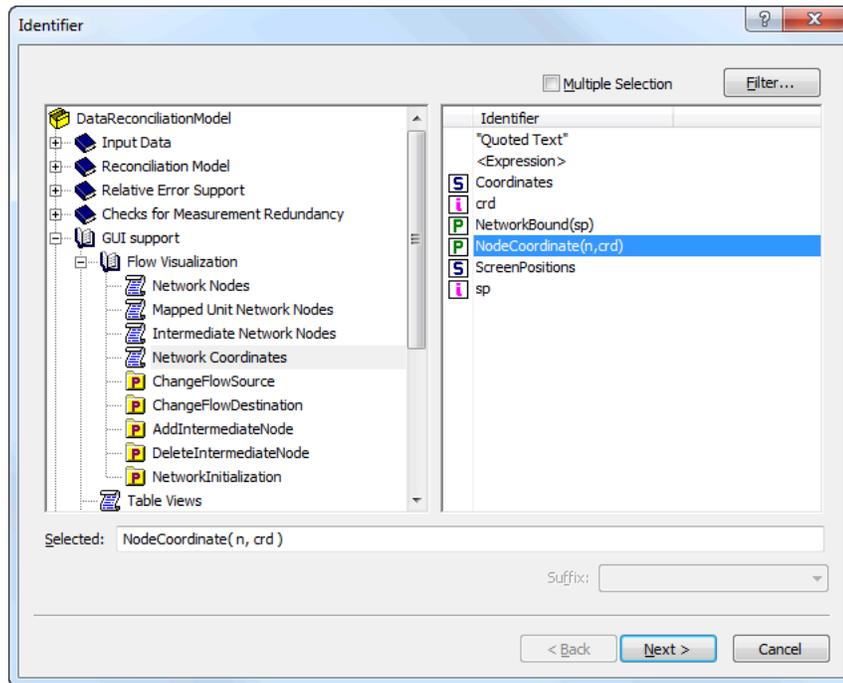
*... from a subselection*

When your project contains one or more library projects, AIMMS only allows you to select identifiers that are part of the interface of a library on any page not included in such a library (see also Section 3.2). If the page is part of the page tree of a library, AIMMS allows you to select from *all* the identifiers declared in the library.

*Selecting from a library*

By restricting access from within pages outside of the library to the identifiers in the library interface only, AIMMS allows you to freely modify the internal implementation of your library. No other part of the application will be inflicted if you make changes to identifier declarations that are not included in the library interface.

*Ensuring your freedom*

Figure 10.5: The **Identifier Selection** dialog box

In its simplest form, you can use the **Identifier Selection** dialog box to select an entire identifier of the appropriate dimension to fill a selected object. However, the **Identifier Selection** dialog box will also let you consider selecting *slices* of identifiers, or provide automatic *links* between objects. These advanced subjects will be discussed in detail in Section 10.4 below.

*Slices and linking*

After you have selected the identifier(s) necessary to fill the page object with the appropriate model data, AIMMS will draw the object using default settings for properties such as fonts, colors and borders. Later on, you can change these properties (or even modify the defaults) via the **Properties** dialog box of the object (see also Section 11.2).

*Object properties*

If the object region displayed in Figure 10.4 is used to draw a table object, and the identifier selection dialog box in Figure 10.5 is used to select the identifier `NodeCoordinate(n, crd)`, the table in Figure 10.6 results.

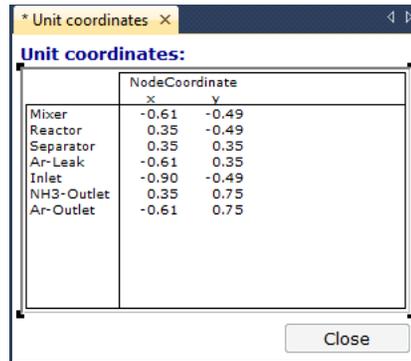
*Example*

---

### 10.3.1 Displaying expressions in page objects

In addition to indexed identifiers, AIMMS also allows you to display expressions in a page object. This is convenient, for instance, when you want to display

*Displaying expressions*



The dialog box titled "Unit coordinates" contains a table with the following data:

|            | NodeCoordinate |       |
|------------|----------------|-------|
|            | x              | y     |
| Mixer      | -0.61          | -0.49 |
| Reactor    | 0.35           | -0.49 |
| Separator  | 0.35           | 0.35  |
| Ar-Leak    | -0.61          | 0.35  |
| Inlet      | -0.90          | -0.49 |
| NH3-Outlet | 0.35           | 0.75  |
| Ar-Outlet  | -0.61          | 0.75  |

A "Close" button is located at the bottom right of the dialog box.

Figure 10.6: Example of a newly created table object

some data which is not directly available in your model in the form of an (indexed) identifier, but which can be easily computed through an expression referring to one or more identifiers in your model. In such a case, you do not have to create an additional defined parameter containing the expression that you want to display, but rather you can directly enter the expression in the **Identifier Selection** dialog box, as illustrated in Figure 10.7.

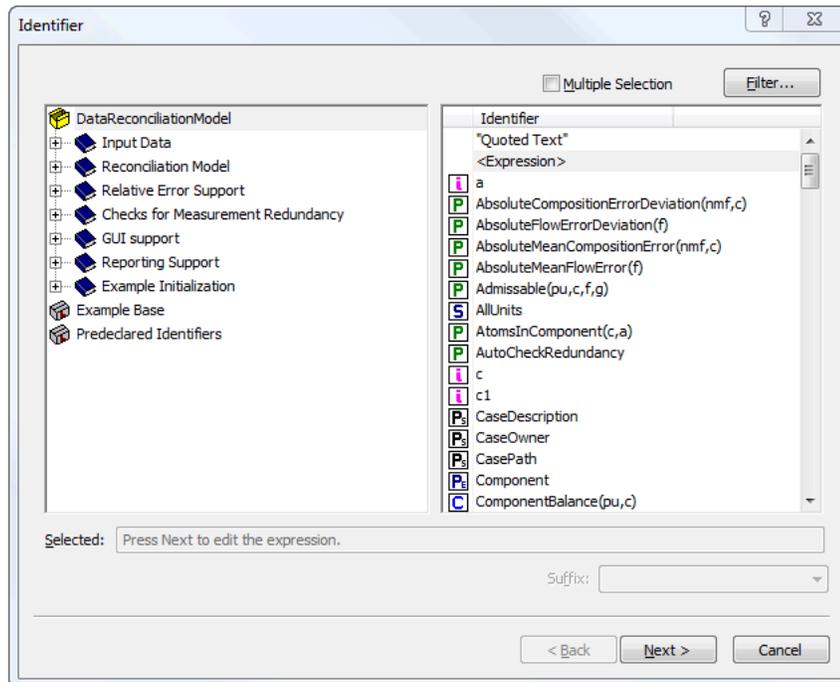


Figure 10.7: Selecting an expression in a page object

When you have indicated that the page object should display an expression rather than an indexed identifier, AIMMS will display the **Expression Definition** dialog box illustrated in Figure 10.8. In this dialog box you must specify the exact definition of the expression you want to be displayed in the page object.

*Entering an expression*

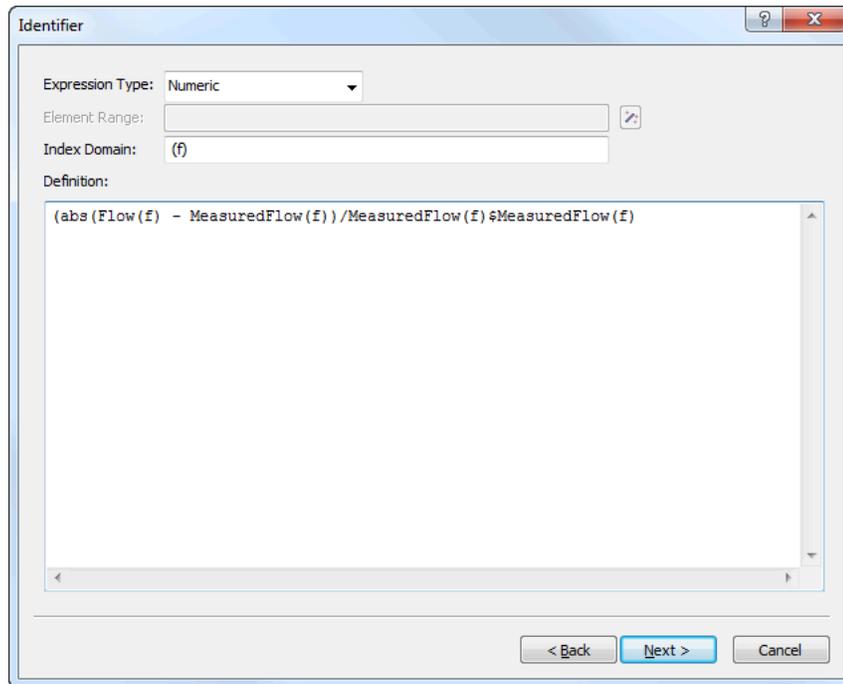


Figure 10.8: Entering an expression for a page object

In the **Expression Type** field of the **Expression Definition** dialog box, you must select the type of the expression you entered. AIMMS only allows the display of

*Specifying the expression type*

- numeric,
- element-valued, and
- string-valued.

expressions. AIMMS does not allow the display of set expressions. If the expression type is element-valued, you must also indicate the element range of the expression, i.e. the set in which the expression will hold its values.

Finally, in the **Index Domain** field of the **Expression Definition** dialog box you must specify the index domain over which the expression is defined. Contrary to the `IndexDomain` attribute in a parameter declaration form, AIMMS only accepts a list of indices in this field, i.e. you cannot add a domain condition (see also Section 4.1 of the Language Reference). If you want to restrict the domain

*Specifying the index domain*

of the expression, you specify the domain condition as a \$ condition within the expression definition (see also Section 6.1.9 of the Language Reference). This is illustrated in Figure 10.8, where `MeasuredFlow(f)` serves as a domain condition on the domain `f`.

### 10.3.2 Creating advanced page objects

In addition to common graphical data objects such as tables, bar charts and curves, AIMMS also supports a number of advanced graphical objects. These objects are designed for specialized, but widely-used, application areas. The most notable advanced objects available in AIMMS are:

*Advanced data objects ...*

- an interactive *Gantt chart* for time-phased scheduling and planning applications, and
- a *network flow object* for applications in which two-dimensional maps or flows play a central role.

Advanced data objects have the characteristic that multiple model identifiers are required to represent the visual result. For instance, in the network flow object you need a set identifier to denote the set of nodes to be displayed and their coordinates in the network, as well as a parameter to indicate the flow values between these nodes. Figure 10.9 illustrates the selection dialog box of a network flow object. To enter the appropriate identifiers for each

*... are based on multiple identifiers*

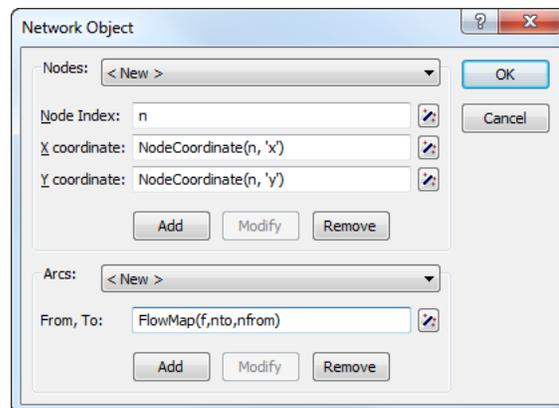


Figure 10.9: Identifier selection for the network flow object

required component, you can open the common **Identifier Selection** dialog box described above by pressing the wizard button  at the right of each individual component.

In this User's Guide you will only find the basic mechanisms for adding or modifying pages and page objects. Full details of all object types, and their properties and settings, are described in the on-line help facility which is always available when you are running AIMMS.

*Object help*

In addition to data-related objects, AIMMS also supports various other types of objects such as:

*Non-data objects*

- drawing objects (such as line, circle, rectangle, picture and text objects), and
- buttons to initiate model execution and page navigation.

Drawing objects and buttons are positioned on a page in exactly the same manner as the data objects described above, except that a link to one or more AIMMS identifiers is not required.

---

## 10.4 Selecting identifier slices and linking objects

After you have selected an indexed identifier (or expression) in the **Identifier Selection** dialog box, a second dialog box appears, as illustrated in Figure 10.10. In this dialog box, you have several options to refine your choice,

*Advanced identifier selection*

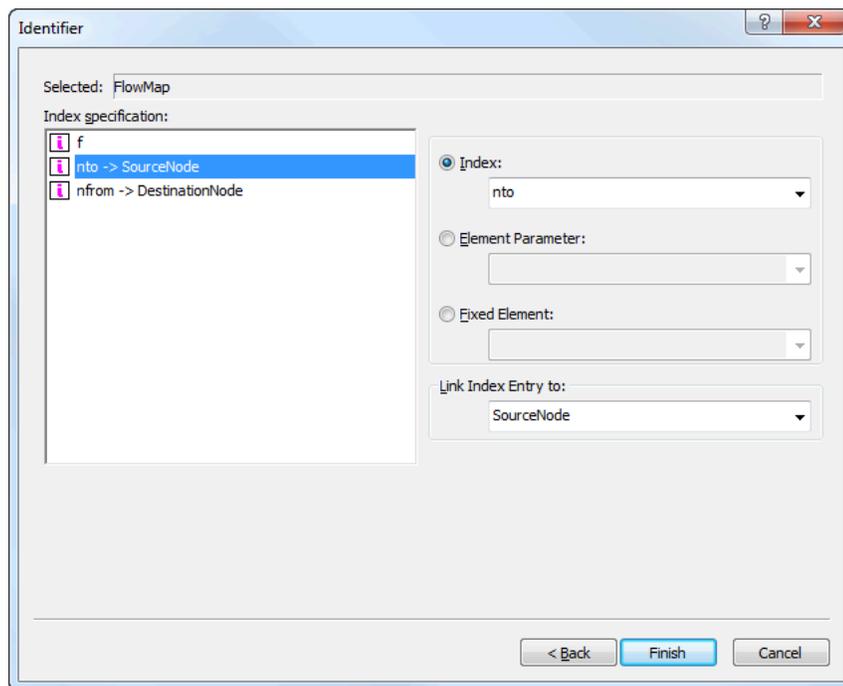


Figure 10.10: Advanced **Identifier Selection** options

each of which will be described in this section.

By default, AIMMS assumes that you want to associate the full identifier with the object in hand. However, with the dialog box of Figure 10.10 AIMMS allows you to modify several domain-related issues before displaying the identifier. More specifically, for every individual dimension in the index domain of the identifier, you can:

*Slicing and subset restriction*

- restrict that dimension to those elements that are included in a particular subset associated with the domain set by using a subset index,
- select a slice of the identifier by fixing that dimension to the value of a particular scalar element-valued parameter that assumes its values into the corresponding domain set, or
- select a slice of the identifier by fixing that dimension to a specific element in the corresponding domain set.

In the dialog box of Figure 10.10 AIMMS lets you select specific elements, element parameters or subset indices on the right-hand side of the dialog box to restrict the dimension that is selected on the left-hand side.

By fixing a particular dimension to an element parameter or a set element, the total number of dimensions of the displayed data is reduced by one. Thus, by fixing one dimension of a two-dimensional parameter, only a one-dimensional table will be displayed. The number of dimensions is not reduced when the display is restricted to elements in a subset. In this case, however, the object will display less data.

*Dimension reduction*

For a table object, the **Identifier Selection** dialog box also lets you determine the order of the dimensions and a split of the dimensions. This allows you to specify which dimensions are shown rowwise and which columnwise, and in which order. If you do not insert a split manually, AIMMS will determine a default split strategy.

*Index order and table split*

Finally, the identifier selection options offer you the possibility of establishing a link between a particular dimension of the selected identifier and a (scalar) element parameter that assumes its values into the corresponding domain set. As an example, consider the dialog box of Figure 10.10. In it, the dimension `nfrom` of the identifier `FlowMap(f, nfrom, nto)` is linked to the element parameter `SourceNode`, and the dimension `nto` to the element parameter `DestinationNode`.

*Index linking*

In the **Properties** dialog boxes of a linked object, AIMMS displays the link using a “->” arrow. Thus, the parameter `FlowMap` from the example above, will be displayed as

*Link notation*

```
FlowMap( f, nfrom -> SourceNode, nto -> DestinationNode )
```

This special link notation is only valid in the graphical interface, and cannot be used anywhere else in the formulation of your model.

When the identifier `FlowMap(f,nfrom,nto)` is displayed in, for instance, a table object, AIMMS will, as a result of the specified index links, automatically assign the values of `nfrom` and `nto` associated with the currently selected table entry to the element parameters `SourceNode` and `DestinationNode`, respectively.

*Effect of index linking*

Index linking is a very powerful AIMMS feature that allows you to effectively implement several attractive features in an end-user interface without any programming effort on your part. Some representative uses of index linking are discussed below.

*Use of index linking*

- You can use index links involving one or more element parameters in a particular page object as a way of triggering AIMMS to automatically update one or more other page objects that contain identifier slices fixed to these element parameters. These updates will occur as soon as a user clicks somewhere in the particular page object in which the indices were linked. An illustrative example of such automatic linkage of page objects is shown below.
- You can use index linking to keep track of the current user selection in an object when executing a procedure within your model. This allows you to do some additional data processing, or perform some necessary error checks for just that tuple in a multidimensional identifier, whose value has most recently been modified by the end-user of your application.

Consider the page shown in Figure 10.11. The tables and lists underneath the

*Example*

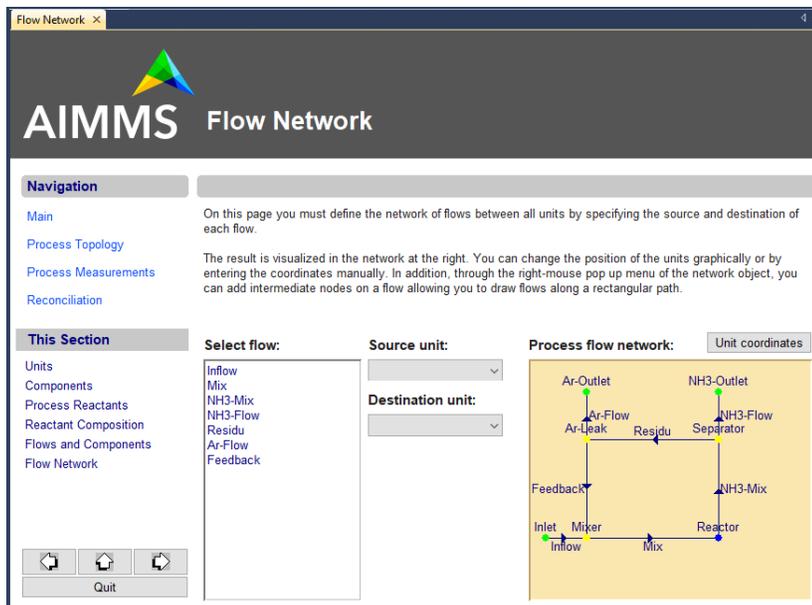


Figure 10.11: Example of index linking

**flow data** text in the center part of the page display detailed information regarding the currently selected flow in the network flow object shown in the right part of the page. This was accomplished as follows. The index  $f$  representing the flows in the network flow object on the right was linked to a single element parameter `FlowE1` in the set `Flows`. The tables and lists on the left of the screen contain identifier slices fixed to the element parameter `FlowE1`. Take, for instance, the values in the column named `Measured` in the table object on the lower left part of the screen. This column corresponds to the one-dimensional identifier slice `MappedMeasuredComposition(c,FlowE1)`. As a result of the link, the column `Measured` automatically displays detailed information for the flow selected by the end- user in the flow chart on the right.