
AIMMS User's Guide - Getting Started

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 2

Getting Started

This chapter provides pointers to the various AIMMS examples that are available online and may help you to get a quick feel for the system. It explains the principle steps involved in creating a new AIMMS project, and it provides you with an overview of the various graphical tools available in AIMMS to help you create a complete Analytic Decision Support application. In addition, the chapter discusses the files related to an AIMMS project.

This chapter

2.1 Getting started with AIMMS

For most people, learning to use a new software tool like AIMMS is made substantially easier by first getting to see a few examples of its use. In this way you can get a quick feel for the AIMMS system, and begin to form a mental picture of its functionality.

Learn by example

In addition, by taking one or more illustrative examples as a starting point, you are able to quickly create simple but meaningful AIMMS projects on your own, without having to read a large amount of documentation. Building such small projects will further enhance your understanding of the AIMMS system, both in its use and in its capabilities.

Getting started quickly

To get you on your way as quickly as possible, the AIMMS system comes with a tutorial consisting of

AIMMS tutorial

- a number of live demos illustrating the basic use of AIMMS,
- an extensive library of small example projects each of which demonstrates a particular component of either the AIMMS language or the end-user interface,
- a number of complete AIMMS applications, and
- worked examples corresponding to chapters in the book on optimization modeling.

The library of small example projects deals with common tasks such as

- creating a new project,
- building a model with your project,
- data entry,
- visualizing the results of your model,
- case management, and
- various tools, tips and tricks that help you to increase your productivity.

Example projects

By quickly browsing through these examples, you will get a good understanding of the paradigms underlying the AIMMS technology, and you will learn the basic steps that are necessary to create a simple, but fully functional modeling application.

What you learn

Rather than providing an introduction to the use of AIMMS, the User's Guide deals, in a linear and fairly detailed fashion, with all relevant aspects of the use of AIMMS and its modeling tools that are necessary to create a complete modeling application. This information enables you to use AIMMS to its full potential, and provides answers to questions that go beyond the scope of the example projects.


This User's Guide

The Language Reference deals with every aspect of AIMMS data types and the modeling language. You may need this information to complete the attribute forms while adding new identifiers to your model, or when you specify the bodies of procedures and functions in your model.

The Language Reference

The Optimization Modeling guide provides you with the basic principles of optimization modeling, and also discusses several advanced optimization techniques which you may find useful when trying to accomplish nontrivial optimization tasks.

The Optimization Modeling guide

You can get online help for most of the tools, attribute forms and objects within the AIMMS system through the **Context Help** button  on the AIMMS toolbar. It will point you to the appropriate section in one of the AIMMS books or in one of the help files that accompany the system.

Context sensitive help

The following strategy may help you to use AIMMS as efficiently and quickly as possible.

How to proceed

- Study some of the working examples to get a good feel for the AIMMS system.
- Select an example project that is close to what you wish to achieve, and take it as a starting point for your first modeling project.
- Use context help if you encounter a component of the AIMMS system that you are not familiar with, and are in need of a quick pointer providing some basic information.

- Consult any of the three AIMMS books whenever you need more thorough information about either the use of AIMMS, its language or tips on optimization modeling.

2.2 Creating a new project

Every AIMMS project consists of two main components:

Project components

- an AIMMS project file (with a .prj extension), which contains all project settings as well as all user-interface related components in the project such as end-user pages and menus, and
- an AIMMS model file (with an .amb extension), containing all identifier declarations, procedures and functions that make up the core of your model.

When you start a new AIMMS project, you can either let AIMMS create both components (as explained below), or select an existing .prj or .amb file.

Within an AIMMS session, you can create a new project file through the **File-New Project** menu. Note that this menu is only available when no other project is currently open. It will open the AIMMS **New Project** wizard illustrated in Figure 2.1. In this wizard you can enter the name of the new project, along

Creating a new project file

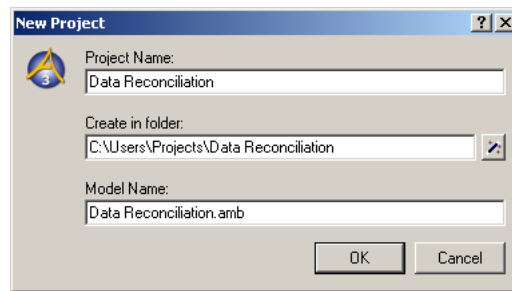



Figure 2.1: The AIMMS **New Project** wizard

with the directory in which the project is to be stored, and the model file (with the .amb extension) to be associated with the project.

By default, the AIMMS **New Project** wizard suggests that the new project be created in a new subdirectory with the same name as the project itself. You can use the wizard button  to the right of the directory field to modify the location in which the new project is created. However, as AIMMS creates a number of additional files and directories in executing a project, you are strongly advised to store each AIMMS project in a separate directory.

Project directory

By default, the AIMMS **New Project** wizard assumes that you want to create a new model file with the same name as the project file (but with a different extension). You can modify the name suggested by the wizard to another existing or nonexistent model file. If the model associated with the project does not yet exist, it will be automatically created by AIMMS.

Model file

After you have finished with the **New Project** wizard, AIMMS will open the **Model Explorer**, an example of which is illustrated in Figure 2.2. The Model Ex-

The Model Explorer

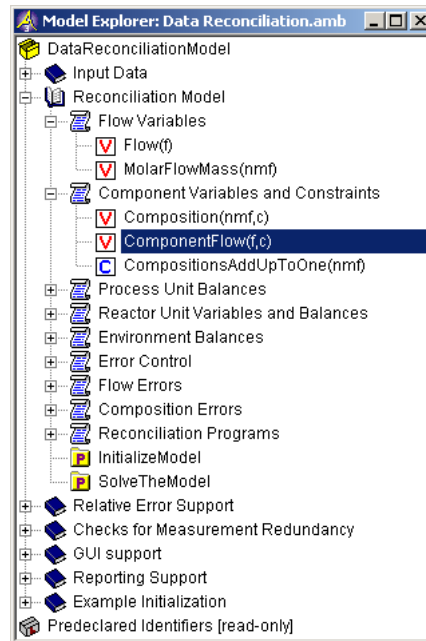


Figure 2.2: The AIMMS Model Explorer

plorer is the main tool in AIMMS to build an AIMMS model, the starting point of building any AIMMS application. In the Model Explorer, the model is presented as a tree of identifier declarations, allowing you to organize your model in a logical manner and make it easy—both for you and others who have to inspect your model—to find their way around. Besides the Model Explorer, AIMMS provides a number of other development tools for model building, GUI building and data management. An overview of these tools is given in Section 2.3.

As an alternative to creating a new project file inside AIMMS itself, you can directly create a new AIMMS project from within the Windows Explorer. After you have installed AIMMS on your machine, the right-mouse pop-up menu in the right pane of the Windows Explorer window allows you to insert a new AIMMS project file into any directory via the **New-AIMMS Project File** menu. This will start a new AIMMS session, and directly open the **New Project** dialog box discussed above to create a new AIMMS project in the selected directory.

Creating a project in the Windows Explorer

You can open an existing AIMMS project in two ways. You can either

- start AIMMS and open the project via the **File-Open Project** menu, or
- double click on the AIMMS project file (with a `.prj` extension) in Windows Explorer.

After opening a project, AIMMS may take further actions (such as automatically opening pages or executing procedures) according to the previously stored project settings.

Starting an existing project

2.3 Modeling tools

Once you have created a new project and associated a model file with it, AIMMS offers a number of graphical tree-based tools to help you further develop the model and its associated end-user interface. The available tools are:

- the *Model Explorer*,
- the *Identifier Selector*,
- the *Page Manager*,
- the *Template Manager*,
- the *Menu Builder*,
- the *Data Manager*, and
- the *Data Management Setup* tool.

These tools can be accessed either through the **Tools** menu or via the project toolbar. They are all aimed at reducing the amount of work involved in developing, modifying and maintaining particular aspects of your model-based end-user application. Figure 2.3 provides an overview of the windows associated with each of these tools.

Modeling tools

The AIMMS **Model Explorer** provides you with a simple graphical representation of all the identifiers, procedures and functions in your model. All relevant information is stored in the form of a tree, which can be subdivided into named sections to store pieces of similar information in a directory-like structure. The leaf nodes of the tree contain the actual declarations and the procedure and function bodies that make up the core of your modeling application. The **Model Explorer** is discussed in full detail in Chapter 4.

The Model Explorer

While the **Model Explorer** is a very convenient tool to organize all the information in your model, the **Identifier Selector** allows you to select and simultaneously view the attributes of groups of identifiers that share certain functional aspects in your model. By mutual comparison of the important attributes, such overviews may help you to further structure and edit the contents of your model, or to discover oversights in a formulation. The **Identifier Selector** is discussed in full detail in Chapter 7

The Identifier Selector

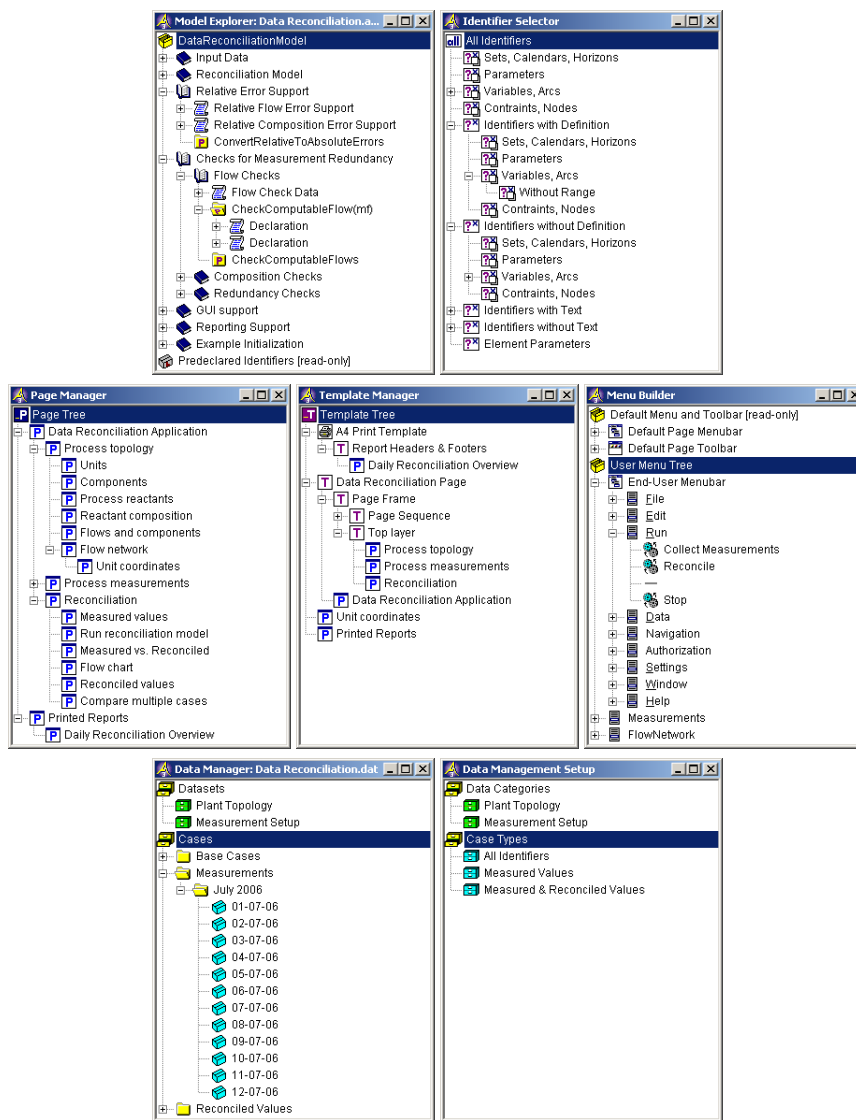


Figure 2.3: Overview of AIMMS tools

The **Page Manager** allows you to organize all end-user windows associated with an AIMMS application (also referred to as end-user pages) in a tree-like fashion. The organization of pages in the page tree directly defines the navigational structure of the end-user interface. Relative to a particular page in the page tree, the positions of the other pages define common relationships such as *parent* page, *child* page, *next* page or *previous* page, which can be used in navigational controls such as buttons and menus. The **Page Manager** is discussed in full detail in Section 12.1.

The Page Manager

Within the **Template Manager**, you can make sure that all end-user pages have the same size and possess the same look and feel. You can accomplish this by creating page templates which define the page properties and objects common to a group of end-user pages, and by subsequently placing all end-user pages into the tree of page templates. The **Template Manager** is discussed in full detail in Section [12.2](#).

The Template Manager

With the **Menu Builder** you can create customized menu bars, pop-up menus and toolbars that can be linked to either template pages or end-user pages in your application. In the menu builder window you can define menus and toolbars in a tree-like structure similar to the other page-related tools, to indicate the hierarchical ordering of menus, submenus and menu items. The **Menu Builder** is discussed in full detail in Section [12.3](#).

The Menu Builder

AIMMS offers an advanced scheme for storing model results and dealing with multiple scenarios through the use of cases and datasets. With the **Data Manager** you can manage the entire collection of cases and datasets constructed for a particular AIMMS application. In addition, you can use the **Data Manager** to initiate a batch run for a set of cases or to view model data for a selection of cases. The **Data Manager** is discussed in full detail in Chapters [17](#) and [18](#).

The Data Manager

With the **Data Management Setup** tool you can specify the types of cases and datasets that are relevant for your application. Such a subdivision may help your end-users to store only the information necessary during a certain stage of your model, or to share common data between multiple cases. The **Data Management Setup** tool is discussed in full detail in Chapter [18](#).

The Data Management Setup tool

2.4 Dockable windows and tabbed MDI mode

As of AIMMS release 3.4, AIMMS supports the use of dockable windows for the tools discussed in the previous section. Dockable windows are an ideal means to keep frequently used tool windows in a development environment permanently visible, and are a common part of modern Integrated Development Environments (IDE) such as *Visual Studio .NET*.

Support for dockable windows

Dockable windows can be used in a *docked*, *auto-hidden*, or *floating* state. Whether a dockable window is in a docked, auto-hidden or floating state can be changed at runtime through their system context menu.

Docking states

When docked, the tool windows are attached to the left, right, top or bottom edge of the client area of the main AIMMS window. By default, all modeling tools discussed in Section [2.3](#) are docked at the left edge of the AIMMS window,

Docked windows

as illustrated in Figure 2.4. When you open a second instance of the same tool it is docked at the right edge of the AIMMS window.

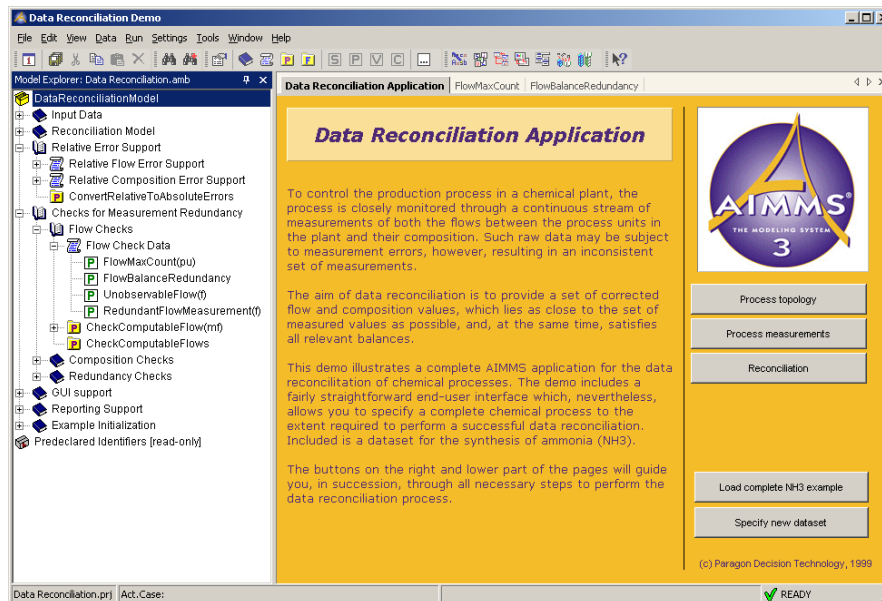



Figure 2.4: Example of dockable windows and tabbed MDI mode

By dragging the windows caption of a docked window and moving the cursor around the edges of the AIMMS window, you can move the docked window to another position. A drag rectangle snaps into place at the appropriate edge, whenever a dockable window is ready to be docked at the location where you moved it. By moving the windows caption into the window area of another, dockable windows can also be docked to each other as tabs. Figure 2.4 illustrates the **Model Explorer** and **Page Manager** docked to each other as tabs. The area of a docked window can also be split into two by dragging another dockable window into the upper, lower, left or right part of docked window. In all these cases, a drag rectangle shows how a dockable window will be docked when you release the mouse at that time.

*Dragging
docked windows*

In auto-hidden state, a dockable window is normally collapsed as a button to the upper, lower, left or right edge of the main AIMMS window. By default, the **Message/Error** window of an AIMMS application, is collapsed to the bottom edge of the main AIMMS window. When you push the button associated with a collapsed window, it is expanded. When an expanded tool window loses focus, it is collapsed again. By clicking the pushpin button  in the caption of a docked/collapsed window, you can change the window's state from docked to auto-hide and back. Whenever a runtime error occurs, the **Message/Error** window will be activated and expand. If you do not close it, the window will

*Auto-hidden
windows*

collapse automatically as soon as you continue working in any other window. If another error occurs, the window will re-expand automatically.

By dragging a tool window away from an edge of the main AIMMS window, it becomes floating. When AIMMS is the active application, floating tool windows are always visible on top of all other (non-floating) AIMMS windows.

Floating tool windows

In addition to dockable windows, AIMMS 3.4 supports *tabbed MDI* mode for all other (document) windows such as attribute windows, identifier views and pages. In Figure 2.4 the main page of the application is displayed in tabbed MDI mode, with the attribute windows of two identifiers in the model accessible through tabs. Tabbed MDI windows occupy the remaining space of the client area of the main AIMMS window that is not occupied by docked windows. This implies that you do not have control over the size of tabbed MDI windows. Therefore, if you use tabbed MDI mode in your AIMMS application, it makes sense to make all the pages in your model resizable (see Chapter ??).

Tabbed MDI mode

When you drag the tab associated with a document window in the document window area, you can move the document window into a new or existing *tab group*, at the left, right, top or bottom of the current tab group. Tab groups effectively split the document window area into multiple parts, each of which can hold one or more tabbed MDI windows. As with dragging dockable windows, a drag rectangle shows where the window will be positioned if you drop it at that moment. Tab groups are very convenient, for instance, if you want to view two attribute windows simultaneously.

Tab groups

By default, AIMMS uses dockable windows and tabbed MDI style document windows when it starts up. If you do not wish to use these features, you can change the style of the AIMMS IDE during an AIMMS session, through the **Windows- IDE Style** menu. If you uncheck the **Use Docking Windows** item, all tool windows that you open from then will be opened as ordinary document windows. If you uncheck the **Use Tabbed MDI Mode** item, AIMMS will display all document windows as ordinary MDI document windows. Thus, if you uncheck both features, all AIMMS windows will be displayed as ordinary MDI document windows, compatible with the window display mode of AIMMS 3.3 and earlier.

Change IDE style

You can also change the IDE style permanently, on a per-project basis. Through the project options contained in the **Options-Project-IDE Style** area (see also Section 22.1), you can permanently change the IDE style settings for a specific project. If a project is not suitable for tabbed MDI mode by design (e.g. because it contains a lot of small pages that cannot be displayed properly in full screen size tabbed MDI windows), you can also disable the **Windows-IDE Style** menu item for such a project. Thus, you can prevent your end-users from changing

IDE style project options

the IDE style of AIMMS if that is undesirable for your project.

2.5 Files related to an AIMMS project

A number of files are associated with every AIMMS project. Some of these files are necessary to run your project, while others are optional or may be generated automatically by AIMMS. The following file types can occur:

Project-related files

- the main project file (with a `.prj` extension),
- one or more model source files (with a `.amb` extension),
- the name change file (with a `.nch` extension),
- one or more data manager files (with a `.dat` extension),
- a user database file (with a `.usr` extension),
- one or more library project files (with a `.libprj` extension),
- encrypted model source files (with a `.aeb` extension),
- project and data backup files (with a `.bak` extension),
- ASCII backup files of the model source (with a `.aim` extension), and
- log, error and listing files from both AIMMS and its solvers (with `.log`, `.err`, `.lis` or `.sta` extensions).

The main project file (with the `.prj` extension) contains the general setup information related to the project, such as the name of the model source file or user database associated with the project, the project options and settings, as well as the data management setup. In addition, the project file contains the specification of all the end-user pages, templates and menus.

The main project file

A model source file (with a `.amb` extension) contains the source code associated with a project in a binary format. The source code of a project may be separated into several `.amb` files, which can be included in the main model file. This allows multiple developers to work on the source of a single project, or to use common declarations and procedures within multiple projects.

The model source

AIMMS has the capability to keep track of the name changes you performed on identifiers in your model, and automatically replace an old identifier name by its new one whenever AIMMS encounters a renamed identifier. AIMMS keeps track of the list of such name changes in the name change file (with a `.nch` extension). Each name change is listed in this file on a separate line containing the old name, the new name and the (GMT) time stamp at which the change was actually performed. The automatic name change capabilities of AIMMS are explained in full detail in Section [5.3](#)

Name change file

Data manager files (with a `.dat` extension) contain a set of cases and datasets created in your AIMMS model-based application. By default, AIMMS will create and open a single data manager file with the same name as the project file. You can open other data manager files (containing other cases and datasets) through the **File-Open** menu.

Data manager files

With every end-user project created with AIMMS, you can associate an end-user database containing the names and passwords of all end-users entitled to run the project or view its results. Such end-user information is stored in an encrypted format in a user database file (with a `.usr` extension). You can use a single user database file with more than one project.

The user database

One or more library projects can be added to your project. A library project can be seen as a complete project (including model source, templates, end-user pages and menus) that can be accessed from within the main project. The `.libprj` file is, as its name might imply, the project file of a library. Libraries in AIMMS make it easier to work with multiple developers on a single project. A detailed description of all multi-developer features can be found in Chapter 3.

Library project files

When shipping your project to an end-user you can choose protect your model by encrypting the model source into a *one way encrypted* model. Such an encrypted model file will have the extension `.aeb` and cannot be viewed by the end-user. The process of generating a one way encrypted model is described in detail in the chapter on project security (see Chapter 21, Section 21.1).

Encrypted source files

During the execution of your model, all log, error and listing information from both AIMMS and its solvers (whether visible in the AIMMS **Message** window or not) is copied to log, error and listing files, which, by default, are stored in the Log subdirectory of the project directory. If you are not interested in this information, you can reduce the amount of information that is copied to these log files by modifying the relevant execution options.

Log, error and listing files

2.5.1 Project backup files

When you are developing an AIMMS model, AIMMS will create regular backups (with a `.bak` extension) of the project files and data associated with your project. By default, such backup files will be stored in the Backup subdirectory of the project directory. In addition to storing a backup file of the binary source file (with the `.amb` extension), most AIMMS versions can also create ASCII backup files of the model source (with a `.aim` extension) for your convenience.

Backup files

You can modify AIMMS' default project backup settings through the **AutoSave & Backups-Project** menu, which will pop up the **Project AutoSave & Backups** dialog box illustrated in Figure 2.5. In this dialog box, you can specify

Project backups

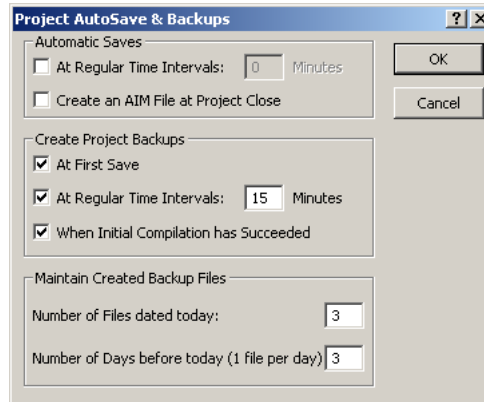


Figure 2.5: The **Project AutoSave & Backups** dialog box

- **Automatic Saves**
 - whether you want AIMMS to automatically save your project at regular time intervals,
 - whether you want AIMMS to create an ASCII .aim file of your model when you close the project,
- **Create Project Backups**
 - whether you want to create a backup of your project after the first time you have saved it,
 - at which times you want AIMMS to store a backup of your project during your session and
 - whether you want to store a backup of your project right after you have successfully opened it, to which you can safely fall back in case anything goes wrong during the current AIMMS session,
- **Maintain created backup files**
 - how old the project backups AIMMS may become,
 - how many project backups AIMMS will retain for changes made today.

When you have checked the .aim file creation check box, AIMMS will create a .aim file of your model whenever you close the project. You can manually create .aim files of your model through the **File-Save As** menu in the **Model Explorer**. The .aim file contains an equivalent ASCII representation of the contents of the model tree, mainly intended for your convenience.

Creating .aim files

If you so desire (e.g. after making manual edits to the .aim file), you can restore the model tree from a .aim file through the **File-Open- Model** menu. When you edit a model loaded from a .aim file, AIMMS will automatically convert it back to a binary .amb file when you save the model for the first time.

Loading .aim files

In the **AutoSave & Backups** dialog box, you can indicate at which times you want AIMMS to create backups of your project. By default, AIMMS will save such backup files (with the .bak extension) in the Backup directory of the project directory. You can modify these settings in the **Project-Directories** folder of the **AIMMS Options** dialog box (see Section 22.1). A backup file contains a copy of the project including all .prj, .libprj, .amb files and possibly the .nch files, but not including the .dat file. The name of the backup contains a reference to the date/time of the backup. To keep the number of backup files down to a reasonable number, the dialog box of Figure 2.5 allows you to indicate how many backup files AIMMS will retain when closing the project.

Backup policy

Through the **AutoSave & Backups-Data** menu, you can specify that you want AIMMS to automatically create backups of the data used during a particular session of your project. The menu will pop up the **Data Backup** dialog box illustrated in Figure 2.6. Similarly as with the project backup files, you can

Data backups

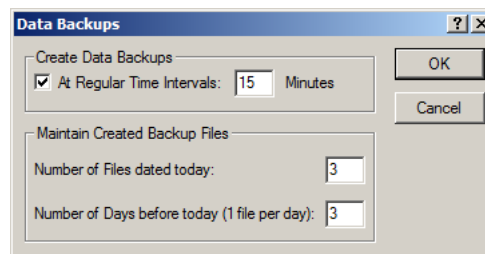


Figure 2.6: The **Data Backup** dialog box

indicate whether AIMMS should automatically create backup backup files of the session data at regular intervals, as well as how many data backup files should be retained. Data backup files also have the .bak extension and contain a reference to the date/time of the backup.

Besides the automated backup scheme built into AIMMS, you can also create backup files of either your project or the session data manually. You can create manual backup files through the **File-Backups-Project-Backup** and **File-Backups-Data-Backup** menus. When you create a project or data backup file manually, AIMMS will request a name of a .bak file in which the backup is to be stored.

Manually creating backup files

Through the File-Backups menu, you can also restore project and data backup files. When you decide to restore the project files, AIMMS will request you to select a project backup file to restore from, after which you get an overview of the file differences (compared to the current project files) in the **Restore**

*Restoring
backup files*

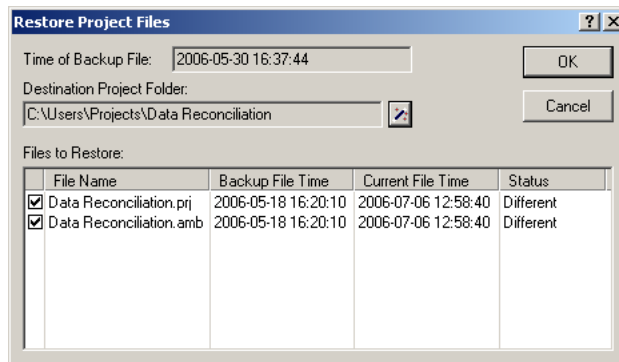


Figure 2.7: The **Restore Project Files** dialog box

Project Files dialog box illustrated in Figure 2.7. In this dialog box, you can indicate which parts of the project backup you want to be restored. This allows you, for instance, to selectively restore the model file, but retain the current project file.

2.5.2 Project user files

Along with the project-related files created by AIMMS, you may need to distribute some other files with your project. Such files include, for instance, bitmap files displayed on buttons or in the background of your end-user pages, or files that contain project-related configuration data. Instead of having to include such files as separate files in the project directory, AIMMS also allows you to save them within the project file itself. Both within the AIMMS language as well as in the end-user interface, you can reference such *project user files* as if they were ordinary files on disk.

Project user files

User project files are convenient in a number of situations. The most common reasons to store files as project user files are listed below.

*Why use project
user files?*

- You want to reduce the number files that you have to ship to your end users. This situation commonly occurs, for instance, when the end-user interface of your project references a large number of bitmap files.
- You want to hide particular configuration data files from your end-users, which might otherwise only confuse them.

You can import files into the project file through the **Tools-Project User Files** menu, which will pop up the **Project User Files** dialog box illustrated in Figure 2.8. In this dialog box, you can create new folders to organize the files

*Importing
project user files*

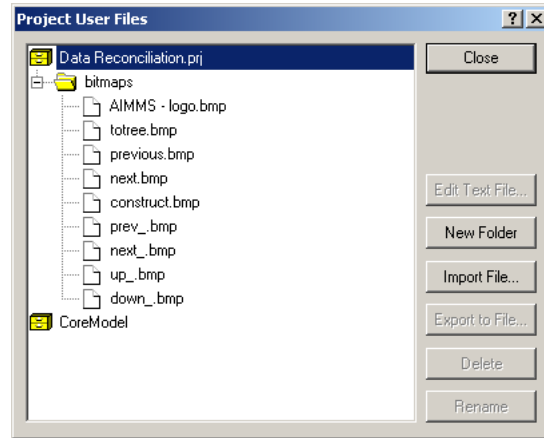


Figure 2.8: The **Project User Files** dialog box

you want to import into the project file. The dialog box of Figure 2.8 already contains a folder **bitmaps**, which is automatically added to each new AIMMS project and filled by AIMMS with the bitmaps used on AIMMS' data pages (see Section 5.4). When you are inside a folder (or just within the main project file), you can import a file into it through the **Import File** button, which will open an ordinary file selection dialog box to select the disk file to be imported.

When your project, next to the main project file, also includes a number of library project files (see Section 3.1), AIMMS allows you to store user files in the library project files as well. Thus, if a page defined in a library refers to a particular bitmap file, you can also store that bitmap as a user file directly into the corresponding library project file. In the dialog box of Figure 2.8, the *CoreModel* node at the root of the tree refers to a library that is included in the project that serves as the running example throughout this book. Underneath this node you can add user files that will be stored in the library project file for the *CoreModel* library.

*User files in
library projects*

When you import a bitmap file (with the *.bmp* extension) into the project file, AIMMS will compress its contents before storing it into the project file (and decompress it before referencing its contents). Such compression may lead to a considerable storage reduction when you are importing large bitmap files, and forms an additional argument to store bitmap files used within the end-user interface into the project file.

*Bitmap files are
compressed*

You can reference project user files both from within the AIMMS language and the properties of various objects with the graphical end-user interface. The basic rule is that AIMMS considers the project file as a virtual disk indicated by “<prj>”. You can use this virtual drive in, for instance, READ, WRITE and PUT statements within your model. Thus, the statement

```
READ from file "<prj>:config\\english.dat";
```

reads the model data from the project user file "english.dat" contained in a (developer-created) **config** folder within the project file.

You can access project files in library projects by using the virtual disk notation “<lib:library-name>”, where *library-name* is the name of the library project. Thus, to read the same file as in the previous paragraph from the *CoreModel* library shown in Figure 2.8, the following statement can be used.

```
READ from file "<lib:CoreModel>:config\\english.dat";
```

Similarly, you can reference project user files on page objects in the end-user interface of your project. Figure 2.9 illustrates the use of a bitmap file stored in the project file on a bitmap button. For all object properties expecting a file

*Referencing
project user files*

*Referencing
user files in
library projects*

*Use in end-user
interface*

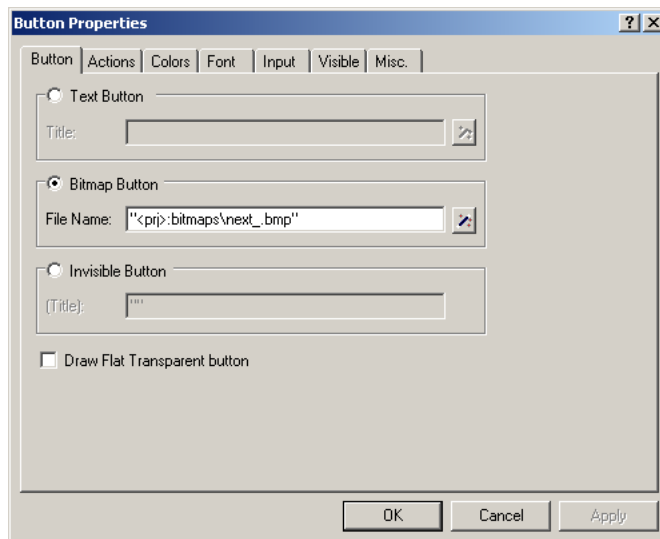


Figure 2.9: Bitmap button referencing a project user file

name (such as the **File Name** property of the bitmap button illustrated in Figure 2.9), you can easily select a project user file by pressing the wizard button , and selecting the **Select Project File** menu item. This will pop up a project user file selection dialog box similar to the dialog box shown in Figure 2.8.

2.6 AIMMS 3 licensing

AIMMS offers the following two types of licenses:

*AIMMS 3
licensing*

- single-user licenses, and
- network licenses.

Each of these two types of licenses are protected in a different manner.

Single-user licenses can be used by a single user on a single computer. To enforce the single-user character, AIMMS requires that single-user licenses be protected by either

*Single-user
license
protection*

- a hardware dongle, which, depending on the type of dongle, must be connected to a USB or parallel port of your computer, or
- a nodelock file, which must be activated to match the hardware characteristic of your computer.

When ordering the Windows version of the AIMMS software you can indicate whether you want your AIMMS system to be protected by a dongle or by a nodelock. Which choice to make is very dependent on your situation and the intended use of the AIMMS software. The Linux platform does not support dongles.

You can choose

Dongles offer you the most flexibility when you want to use AIMMS on multiple computers, and do not want the hassle of having to deactivate and activate a nodelock on these computers. On the other hand, dongles occasionally break, you can forget to take the dongle with you, they can be stolen, and, because of their size, get lost quite easily, especially if you are moving them around a lot.

*Pros and cons of
dongles*

Nodelock files are stored on the harddisk of your computer, and are, therefore, much less vulnerable to loss. Only if your computer is stolen, or in case of a harddisk crash, you must contact Paragon before being able to activate your nodelock on a replacement computer. On the other hand, if you are frequently working on multiple computers, you have to remember to deactivate the nodelock on the old computer, prior to being able to activate it on the new one, every time. In addition, you need access to the internet to activate or deactivate a nodelock.

*Pros and cons of
nodelocks*

If you decide to request a dongle for license protection, a physical shipment of the dongle to your site is required before you can start using AIMMS. If you request nodelock protection, we will send you the AIMMS license number and activation code by e-mail, after which you can start using AIMMS directly.

*Physical
shipments*

If you request your license to be protected by a dongle, an AIMMS dongle is sent to you along with your AIMMS 3 CD-ROM. If you upgrade from AIMMS 2.20, you can continue to use your existing green Activator dongles used by AIMMS 2.20, *but only with the Win32 version of AIMMS*. The green Activator as well as the grey Sentinel dongle must be connected to the parallel port of your computer. The purple Sentinel dongle must be connected to a USB port of your computer. The AIMMS setup program will only install the required device drivers for accessing the grey and purple Sentinel dongles. If you still use the green Activator dongles supplied with AIMMS 2.20, you can obtain the required drivers separately from our FTP site <ftp.aimms.com>.

AIMMS dongles

If you have ordered an AIMMS 3 network license, no license protection needs to be installed locally on your computer. Instead, you need the host name and port number of the server running the AIMMS 3 network license server. For more information about installing the network license server itself, please refer to the documentation of the AIMMS 3 network license server.

Network licenses

2.6.1 Personal and machine nodelocks

AIMMS offers two types of nodelocks:

Two types of nodelocks

- personal nodelocks, and
- machine nodelocks.

If you choose for nodelock protection you are free to choose between a personal or a machine type of nodelock. In this section you will find the characteristics of both types of nodelocks. If you are unsure which type of nodelock to choose, we recommend that you start with a personal nodelock, as you can change a personal nodelock into a machine nodelock at any time, but not the other way around.

Personal nodelocks are intended for use by a single AIMMS user, who still wishes to have the freedom to use AIMMS on multiple computers, for instance if you want to easily switch between your desktop computer at work, a notebook computer and your home computer. Personal nodelocks have the following characteristics:

Personal nodelock

- Personal nodelocks can be transferred to another computer 3 times per 24 hours. This allows you to take your AIMMS license home in the evening and back to work the next morning without any problems.
- Personal nodelocks have a limited lifetime of 60 days, and should be renewed within that period to extend the lifetime to its full 60-day period. If the nodelock is not renewed within its 60-day lifetime, this does not invalidate your AIMMS license in any way—you only have to renew your nodelock prior to being able to use your AIMMS system again. Note that the renewal process does not require any manual intervention, as

AIMMS will try to automatically connect to our internet license database to renew your nodelock once every day you are using AIMMS.

- Both activation and nodelock renewal of personal nodelocks require a working connection to the internet. As a consequence, in the absence of an internet connection you can continue to work uninterrupted for a period of 60 days, before an internet connection is required to renew your nodelock.
- With every activation or nodelock renewal AIMMS will also update your license files if new license files are available (e.g. if your system is in maintenance), and will inform you of any messages that are available for you in our database.
- Because of their volatile nature, PDT will replace a personal nodelock without any questions asked in case of loss of or damage to your computer.
- You can switch your personal nodelock to a machine nodelock at any time.

Machine nodelocks are intended for permanent use on a single computer. They are recommended for server applications, and can also be used for personal use if you are sure you will be using AIMMS on a single computer, or do not have internet access. Machine nodelocks have the following characteristics:

*Machine
nodelock*

- Machine nodelocks can be transferred to a replacement computer 3 times per 365 days.
- Machine nodelocks have an unlimited lifetime (unless deactivated).
- Machine nodelocks can be either activated online if your computer is connected to the internet, or offline through the license activation area on the AIMMS website.
- License files will only be retrieved when the machine nodelock is activated, or by explicit request.
- In case of failure, PDT will, in principle, only replace machine nodelocks on the same computer.
- Once you have chosen for a machine nodelock, it is not possible to switch back to a personal nodelock.

Although a personal nodelock and the software version check on the start page make a regular connection to the internet (the personal nodelock connects to a license database for nodelock renewal and the version check connects to a version database), we do respect your privacy and will not register patterns in your personal usage of the AIMMS software in any way. During activation no personal information will be transferred, only your computer name and some of its hardware characteristics. During deactivation we register the date and time of deactivation to enforce the transfer limit.

Privacy

The connection to our internet license database is implemented as a web service. Thus, if you are able to browse the web, you should also have no trouble activating an AIMMS nodelock. If your computer connects to the internet through a proxy server, AIMMS by default tries to detect and use the proxy settings also used by Microsoft Internet Explorer.

Internet connection and proxy settings

It should be noted that the use of auto-configuration scripts in determining the proxy server will fail if these use any other scripting language than Javascript. This is due to the libraries underlying the SOAP library used by AIMMS to connect to our license server. If you are in this situation, you should manually configure the proxy settings, as described below.

Automatic configuration scripts

If AIMMS does not succeed in automatically detecting the proxy settings that apply in your network environment, AIMMS also allows you to manually set the proxy settings during the activation process. If the online activation process does not succeed directly, AIMMS gives you the option to either continue with an offline activation process, or to manually supply the proxy settings that apply to your network environment through the dialog box illustrated in Figure 2.10. In this dialog box you can choose between

Manual proxy setting

- the *Current User* settings also used by Microsoft Internet Explorer (default),
- the *Local Machine* settings which are stored in the registry, if these are available on your machine, or
- *Custom proxy* settings that you have received from your IT department.

In the latter case, you can also (optionally) provide a user name and password to authenticate with the proxy server. In most cases, however, setting these will not be necessary, and Windows authentication will be sufficient.

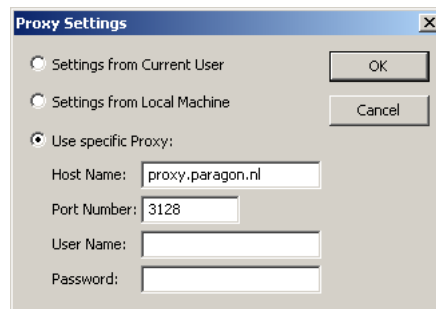


Figure 2.10: The AIMMS **Proxy Configuration** dialog box

2.6.2 Installing an AIMMS license

When you start up AIMMS for the first time after installation, AIMMS will open the **License Configuration** dialog box illustrated in Figure 2.11. Through this

Managing your AIMMS licenses

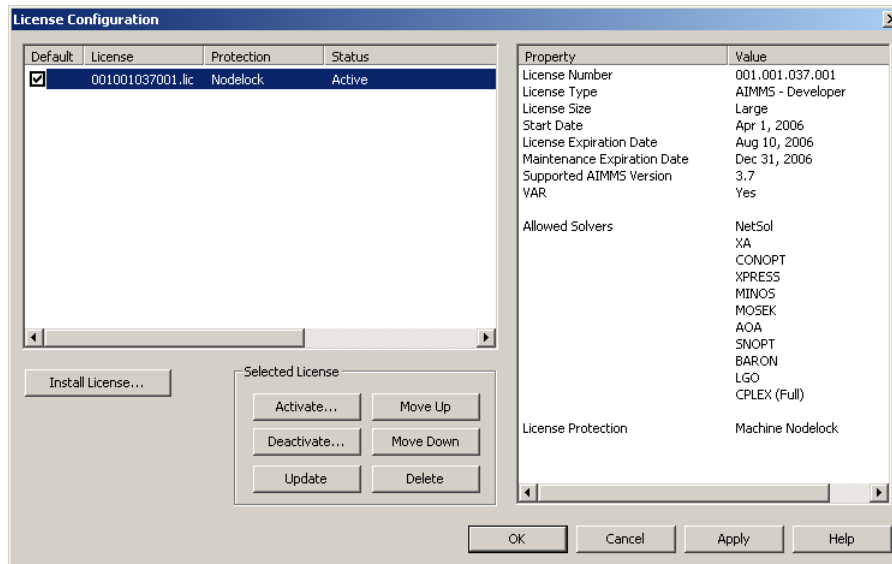


Figure 2.11: The **License Configuration** dialog box

dialog box you can install new AIMMS licenses and manage all AIMMS licenses that already have been installed on your computer.

To install a new license, press the **Install License ...** button in the **License Configuration** dialog box. This will start a wizard, that will guide you through the license installation procedure step by step. The wizard can help you to install

Installing a new license

- existing AIMMS licenses,
- nodelocked licenses,
- dongled licenses,
- network licenses,
- evaluation licenses, and
- student licenses.

To successfully complete the installation of licenses of each type, you should make sure to have the following information available.

To install a single-user AIMMS license that is protected by a nodelock, you need the following information:

*Single-user
nodelocked
licenses*

- your AIMMS license number, and
- the associated activation code that you received from Paragon.

You have the choice to request a personal nodelock or a machine nodelock. A personal nodelock must be requested online, a machine nodelock can be requested online or offline. Refer to Section 2.6.1 for a more detailed introduction to personal and machine nodelocks.

To install a single-user AIMMS license that is protected by a dongle, you need the following items:

*Single-user
dongled licenses*

- an AIMMS dongle attached to a USB or parallel port of your PC, and
- the associated set of license files that you received from Paragon.

To install an AIMMS network license, you need the following information from your system administrator:

*Network
licenses*

- the name of the AIMMS network license server,
- the port number of the AIMMS network license server, and
- the name of the license profile to which you want to connect (optional).

To install an AIMMS evaluation license you need the following information

*Evaluation
licenses*

- your AIMMS evaluation license number, and
- the associated activation code that you received from Paragon when requesting an evaluation license.

You must have a working connection to the internet (not necessarily on the machine on which you installed AIMMS) to activate an evaluation license. Evaluation licenses expire 30 days after activation. Note that each evaluation license can be activated only once, and that you can only activate a single evaluation license per AIMMS release on a specific computer, regardless of the number of evaluation licenses you have requested on our web site.

To install an AIMMS student license you need the following information:

Student licenses

- your AIMMS student license number, and
- the associated activation code that you received from the university that purchased the AIMMS Educational Package.

You must have a working connection to the internet to activate a student license. Student licenses expire one month after the end of the current academic year. Student licenses can be activated multiple times.

2.6.3 Managing AIMMS licenses

AIMMS allows you to have multiple AIMMS licenses installed on your computer. You may have multiple licenses installed, for instance, for the following reasons:

Managing multiple AIMMS licenses

- you have requested a trial license for a new AIMMS version which you want to run next to your existing license,
- you have temporarily borrowed or hired an AIMMS license with more capabilities than your regular license,
- your system administrator has created multiple network license profiles, each of which you may want to use to run AIMMS.

In this section we will describe how you can instruct AIMMS which license to use.

In the **License Configuration** dialog box displayed in Figure 2.11, all AIMMS licenses installed on your machine will be displayed in the left pane of the dialog box. The license details of each license are displayed in the right pane of the dialog box. During startup AIMMS will consider all licenses in the left pane of the **License Configuration** dialog box which have the **Default** column checked, and will use the first valid license it finds starting from top to bottom. Using the **Move Up** and **Move Down** buttons you can change the order in which AIMMS will search the list.

Default licenses

Both personal and machine nodelocks can be transferred to other computers. Personal nodelocks can be transferred upto three times a day, allowing you to take your license with you wherever you want. Machine nodelocks can be transferred three times per year, to a computer replacing the computer on which the nodelock is currently installed. To transfer a nodelocked license, you must

Transferring licenses

- deactivate the nodelock on the currently active computer, and
- activate it on the computer to which you want to transfer the license.

You can deactivate an active nodelock using the **Deactivate** button in the **License Configuration** dialog box. Deactivation will only succeed if there is no conflict with the transfer limit for the given nodelock type. This makes sure that there will never be a problem activating a deactivated license. After successful deactivation the license will not be removed from the list but be marked as inactive. If the license is not active on any computer, you can reactivate the license through the **Activate** button.

In case you want to activate a nodelock on a computer, but have forgotten to deactivate the nodelock on a computer to which you currently have no access, AIMMS allows you, as a courtesy, to request an emergency nodelock 3 times per 365 days. Emergency nodelocks have a lifetime of 7 days, and during this time you can arrange for someone to deactivate the license on the computer containing the active nodelock. During the activation sequence, AIMMS will automatically ask whether you would like to receive an emergency nodelock when it discovers that the license is active on another computer.

*Emergency
nodelocks*

2.6.4 Location of license files

AIMMS keeps its license and configuration files in the folder

Paragon Decision Technology

of the common application area of your computer. On Windows XP, and Windows Server 2003, the common application area is located, by default, at

C:\Documents and Settings\All Users\Application Data

On Windows Vista and Windows 7, this folder is located under C:\ProgramData. The Paragon Decision Technology folder contains three subfolders

- Config, containing the license and solver configuration files,
- Licenses, containing all license files,
- Nodelocks, containing all nodelock files installed on your computer, and
- AnyUser, containing the license configuration files for all users on your computer (see below).

The AIMMS installation makes sure that these subfolders are writable for everyone, allowing you to install and uninstall licenses on your computer.

To prevent tampering with nodelocked licenses, AIMMS keeps track of the location of the nodelock files associated with a license. You should, therefore, not manually move or copy the AIMMS nodelock files as this may invalidate your nodelock.

*Do not move
nodelock files*

You can specify whether the license and solver configuration that AIMMS uses is the same for any user of the machine, or different for each individual user. To prevent problems when running AIMMS as part of a computer service, AIMMS 3.12 will by default use the same configuration for any user. To modify this behavior, you should edit the file UserDistinction.cfg in the common Config folder. In the file UserDistinction.cfg.default, straightforward directions are given on how to edit it. If no UserDistinction.cfg file exists, AIMMS will use the UserDistinction.cfg.default file instead. You can use this file as a base for setting up your own configuration.

*User specific
configuration*

In the scenario where all users of the same pc use the same license and solver configuration, the configurations that are modified by a user are stored in the AnyUser folder of the Common folder. *Any User*

In the scenario where each specific user of the pc has its own license and solver configuration, the configurations that are modified by a user are stored in the local application data folder. On Windows XP and Windows Server 2003, this local folder is usually located at: *Current User*

```
C:\Document and Settings\\Application Data\Paragon
Decision Technology
```

On Windows Vista and Windows 7 it is:

```
C:\Users\\AppData\Roaming\Paragon Decision Technology
```

When AIMMS needs to read the current configuration, it will first look in the (any)user folder as specified by the aforementioned UserDistinction.cfg file; if it cannot be found there, it will try to read the configuration from the common application data folder. When saving a modified configuration, AIMMS always writes to the (any)user folder. *Accessing configuration files*

Inside the Config folder of the (any)user folder, each major AIMMS version (3.6, 3.7, 3.8, etc.) will create its own specific subfolder when it needs to write a configuration file. During an attempt to read, AIMMS will first look for the specific file in the folder that matches its own major version number, and otherwise it will subsequently search through the folders of previous versions. In other words, when you upgrade to a new AIMMS version, initially your configuration will be the same as the one you were using for the previous AIMMS version, but if you change something in the configuration, this will only affect the configuration of the AIMMS version you are working with. *Version dependent configurations*

If you are using an AIMMS Network License, then your local machine does not need to have any license file installed. The only required file is the license configuration file, that contains the info of where the License Server is located on your LAN. When logging on to the License Server, the licensing info is sent directly from the server to the running AIMMS session, except for some secondary license related files: *Network License Client Files*

- the .SLV file (containing the default solver configuration),
- the .VID file (containing the VAR identification info), and
- the .CPX file (the *CPLEX* license file).

These secondary license files are temporarily copied to the folder NetworkCache which is located in the (any)user folder as described above.

You can specify a project dependent solver configuration by placing a solver configuration file with the name 'solvers.slv' in the project directory. AIMMS will first look for this file and if it cannot find it will look for other solver configuration files. See the AIMMS Help for more information.

*Project
dependent
configuration*