
AIMMS User's Guide - Project Security

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 21

Project Security

When you are creating a model-based end-user application there are a number of security aspects that play an important role. *Project security*

- How can you protect the proprietary knowledge used in your model?
- How can you prevent the end-users of your application from modifying the project (thereby creating a potential maintenance nightmare)?
- How can you distinguish between the various end-users and their level of authorization within your application?

AIMMS offers several security-related features that address the security issues listed above. These features allow you to *This chapter*

- irreversibly encrypt the source code of your model,
- password-protect, (reversibly) encrypt, and/or license the project and model files that are part of your application,
- introduce authorization levels into your model, and
- set up an authentication environment for your application.

This chapter describes these mechanisms in full detail, together with the steps that are necessary to introduce them into your application.

21.1 One way encryption

AIMMS supports two manner of encryption of your model source. It is up to you to choose the encryption scheme that works for you. *Two ways of encryption*

- If your only concern is to protect your investment in model development, but do not need on-site access to your model and do not want to license (parts) of your model strictly coupled to an AIMMS license number, the easiest way to accomplish this protection is to use the encryption scheme discussed in this section.
- If you do need on-site access to your model and/or want to protect your model by strictly coupling it to an AIMMS license number, you should use the more involved VAR licensing scheme discussed in Section [21.2](#).

By using AIMMS' one way encryption scheme you simply produce a version of *all* model files in your model that are irreversibly encrypted. You can create a one way encrypted version of your model through the **File-Export** menu, which will open the **Project Export** dialog box illustrated in Figure 21.1. By checking

One way encryption

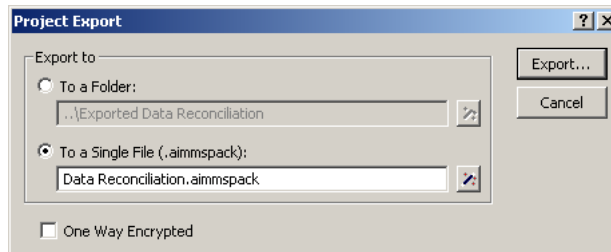


Figure 21.1: The **Project Export** dialog box

the **One Way Encrypted** check box, all `.amb` model files to which you have developer access, will be replaced by equivalent `.aeb` (AIMMS encrypted base) files.

All the model source in the `.aeb` files is irreversibly encrypted, and access to the model tree is unconditionally prohibited in a one way encrypted project.

No access to .aeb models

If you choose to one way encrypt your project and your main project does not yet have a VAR license (see Section 21.2), then AIMMS will automatically add a VAR license to the exported project. Through the **One Way Encryption Settings** dialog box illustrated in Figure 21.2, AIMMS lets you choose to which

Restricting access



Figure 21.2: The **One Way Encryption Settings** dialog box

AIMMS licenses you want to restrict access of the one way encrypted project and/or lets you specify an expiration date. These settings are stored in a `.var` with a fixed name, `_OWE_.var`. If your developer project is already VAR licensed, then the exported project will inherit the VAR license of the developer project.

In the **Project Export** dialog box, you are offered the possibility to export all the files in your project to an export directory of your choice, ready to be shipped to your customer(s). Alternatively, you can ship your application as a single .aimmspack file (see also Section 16.2).

Exporting your project

21.2 VAR licensing project components

If you have invested considerably in the development of an AIMMS model for a particular application area, it is not unreasonable that you should want to protect your investment and strictly control the usage of your application. To support you in this task, AIMMS allows you to password-protect the access to, encrypt, and license the use of individual project and model files through AIMMS' VAR licensing scheme. Through VAR licenses you can strictly enforce your licensing requirements, by coupling the use of your project to individual AIMMS license numbers issued to your customers.

Protecting your investment

Through the **Settings-Project Security** menu you can set up the protection of your (library) project and model files. It will open the **Project Security** dialog box illustrated in Figure 21.3. If you want to password-protect or license your

Protect a project

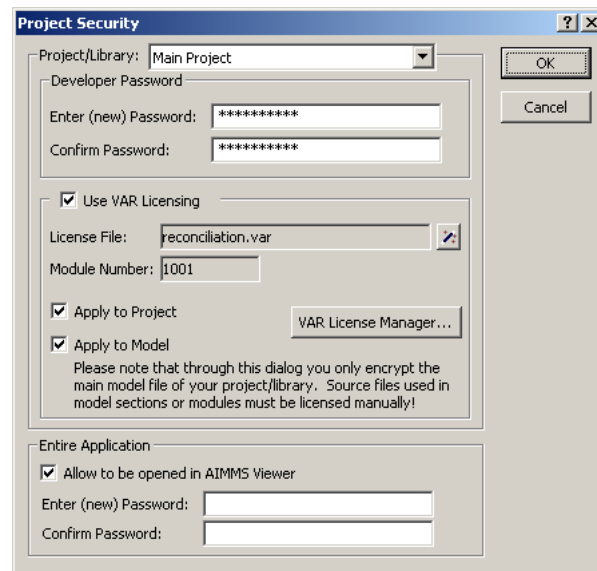


Figure 21.3: The **Project Security** dialog box

project or model as a whole, completing the appropriate sections of the **Project Security** dialog box will suffice. If you want to license separate sections of your model tree containing a fully functional source module ready for use by third parties, you must set up the licensing directly in the model tree, as explained in Section 21.2.2.

The simplest form of project security is by protecting your project and model file through a *developer password*. You can add a developer password to your project by completing the **Developer Password** section in the **Project Security** wizard. An existing developer password is removed by entering an empty password. Once a project is password protected, you need to enter this password every time you open the project in developer mode.

Password protection

Note that adding a developer password to a project will not encrypt the information stored in the model file per se. Although the model file stores its information in a binary format, parts of it may still be readable, potentially giving away proprietary information. To encrypt the information in the model file, you should protect it by a VAR license.

No encryption

AIMMS will add the developer password to both the project and the model file. To open a project in developer mode successfully, the developer password you enter must match the passwords stored in both files. If the password you enter does not match either (stored) password, full developer access to the project will be denied.

Project and model password must match

Through the **Use VAR Licensing** section of the **Project Security** dialog box, you can protect the use of an AIMMS-based modeling application by means of a VAR license. With such a license you can impose restrictions on

Licensing a project

- the expiration date of the application,
- whether the license is a stand-alone or a network license, and
- in the case of a network license, the number of network users that can run the application concurrently.


Before you can create VAR licenses yourself, you must register with Paragon Decision Technology (PDT) as a Value Added Reseller (VAR). After you have done so, you will receive a `.vid` file containing a unique VAR identification, which will enable the **VAR License Manager** in AIMMS (see Section 21.2.1). Using this tool you are able to create the VAR licenses necessary to protect your own AIMMS-based applications on the basis of your VAR identification code. The uniqueness of the VAR identification code issued by PDT ensures that licenses to protect your applications can only be created by you. *You should make sure that these .vid files are never distributed to your end-users, but stay strictly within your development organization.*

VAR identification code

In addition to protecting a project with your own unique VAR identification code, you can associate a unique (integer) module identification code with a particular project (or source module) itself. AIMMS will only allow the use of a particular module if *both* the VAR and module identification codes stored in the project and/or model files coincide with the module identification code

Module identification code

stored in the VAR license. In this manner, you can license several projects and source modules independently.

In the **Project Security** dialog box, you can license the use of your AIMMS-based application on the basis of an existing VAR license file. If you have not yet created a VAR license file, you can open the **VAR License Manager** through the equally named button on the **Project Security** dialog box. By selecting a VAR license file using the **License File** wizard , AIMMS will read the associated module identification code from the selected license file (if applicable) and display it in the **Module Number** field.

Existing VAR license file

You can license the use of the project file and the model file independently. If you license the use of the model file associated with your project, AIMMS will just add the appropriate licensing attributes to the main model node in the model tree (as explained in Section 21.2.2). If required, you can later modify these attributes manually to suit your particular needs.

License project and/or model

Whether you are granted developer or end-user access to a VAR protected AIMMS project depends on various factors, such as

Developer/end-user access

- the AIMMS license matches the VAR license,
- the VAR identification code stored in your organization's .vid file, and
- the entered developer password.

Table 21.1 which type of access is granted under which conditions.

	VAR protection & license not ok	VAR protection & VAR ID not ok	VAR protection & VAR ID ok no VAR protection	
Start end-user	no access	end-user	end-user	
Start developer			<i>passw. ok</i>	<i>passw. not ok</i>
- main project	no access	end-user	developer	end-user
- library	no access	restricted	developer	restricted

Table 21.1: Developer/end-user access for VAR protected projects

When your model contains a VAR protected library project for which you do not have an appropriate VAR license, AIMMS still grants you restricted developer access to the library. In restricted developer mode, you can still refer to the pages, templates and menus contained in the library, but you cannot edit these. More specifically, in restricted developer mode, the following actions are prohibited:

Restricted developer access

- Page & Template Manager:
 - No "Copy"
 - Restriction on "Used Identifiers" (excluding private identifiers)

- No “Open in Edit Mode”
- Menu Builder
 - No “Copy”
 - Restriction on “Used Identifiers” (excluding private identifiers)
 - No “Properties”
- Pages & Templates
 - No “Edit Mode”
- Data Management Setup:
 - No “Properties”
- Project User Files
 - The root for this project remains collapsed

If you have prepared multiple VAR license files each associated with a single AIMMS license, you can distribute your project along with all generated VAR license files as a single package by using AIMMS' *VAR license directory* facility. Instead of specifying a VAR license *file* in the **License File** attribute, AIMMS also allows you to specify a *directory* for this attribute. In that case, AIMMS will look inside the specified directory for a VAR license file (with the .var extension) that matches the AIMMS license number. For instance, if the AIMMS license number is 15.90.10.7, AIMMS will look inside the specified directory for a VAR license file called 015090010007.var.

VAR license directory

21.2.1 Creating a VAR license

When your AIMMS license is associated with a .vid file containing a unique VAR identification code, the **Tools-License-VAR License** menu will be enabled. Through this menu, AIMMS allows you to generate you own VAR licenses. It will open the **VAR License Manager** dialog box illustrated in Figure 21.4.

Creating a VAR license

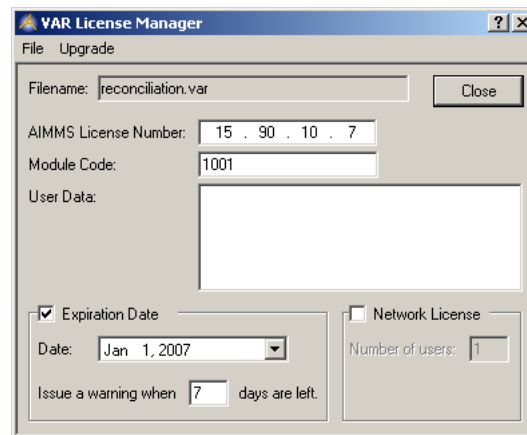


Figure 21.4: The VAR License Manager

In the **VAR License Manager**, you can enter the AIMMS license number of the end-user to whom you intend to distribute your module. This uniquely couples your VAR license to a single AIMMS license. To make your module available to a group of users, who all have similar AIMMS license numbers, you can replace the varying part of these AIMMS licenses by zero. When you enter a zero as one of the components of the AIMMS license number (all of which are separated by dots), AIMMS will accept any number for that component of the AIMMS license. Thus, if you license your module to license number 0.0.0.0, you permit your application to be used by *any* AIMMS license, while a VAR license with license number 15.90.10.0 allows your application to be used by all AIMMS licenses starting with 15.90.10.

*AIMMS license
number*

The module code that you specify for a particular VAR license must coincide with the module identification code stored in the project and/or model files associated with the module that you want to license. AIMMS will only allow an end-user to use your module if the module codes in the VAR license and in the module-related files are the same.

Module code

In addition to coupling your module to a single AIMMS license (or group of AIMMS licenses), you can also limit the period of its use. When you enter an expiration date in the **Var License Manager**, AIMMS will not allow any use of the module after that date. By default, AIMMS will warn your end-user about the expiration date if the application is started within 7 days of the expiration date.

Expiration date

AIMMS and VAR licenses can be stored on a stand-alone machine, as files, or can be provided by an AIMMS network license server to your local area network. You can turn a VAR license into a network license by checking the **Network license** check box and providing the number of concurrent users. In this case, the AIMMS license number must be the license number of the corresponding AIMMS network license.

*Stand-alone
versus network*

If your licensing needs go beyond the standard licensing features described above, AIMMS allows you to store a user-definable string in a VAR license file in which you can store whatever information you require for your particular licensing scheme. When a license has been activated for a particular module, this user data can be made available through a locally declared string parameter (see Section 21.2.2). In the **User data** area of the VAR license manager you can enter any string, of maximum 256 characters, that you want to pass on to your module.

User data

With the **Open**, **Save** and **Save as** items in the **File** menu of the **VAR License Manager** you can open an existing VAR license for modification, resave it, or save a license in a new VAR license file. When your end-users include your module into their model, the VAR license file must be available, either as a local file or through the AIMMS network license server, under the name that you specified in the **License File** field of the **Project Security** dialog box.

Opening and saving VAR licenses

21.2.2 Licensing model sections

As explained in Section 4.2, AIMMS allows you to associate a separate source file with every subtree of your model. Such a separation is not only useful in a multi-developer environment, but also for the separate storage of those parts of your model that can be considered as more or less independent modules. For instance, such a module could consist of

Multiple model files

- the AIMMS interface to a library of DLL functions providing functionality that is not easily modeled in AIMMS itself, or
- a model with well-defined input data to solve a particular problem class.

To create a licensed module, you must first store the source code of the section containing the module in a separate source file by completing the **Source file** attribute of the section. This step is not necessary if you want to VAR license an entire model. When you open the attribute form of the main model node or of the separated section, it will contain a number of additional attributes, as illustrated in Figure 21.5. These attributes allow you to turn the main model

Creating a licensed module

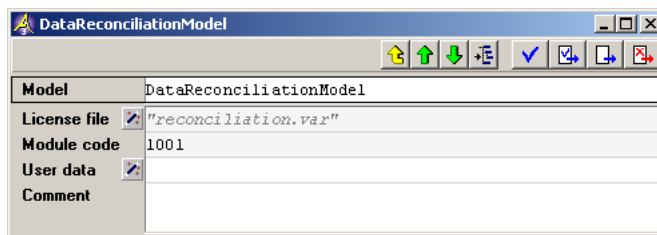



Figure 21.5: Licensing attributes of the main model node

node (or the section) into a licensed module. *Note that these additional licensing attributes are only visible when your AIMMS development organization possesses a .vid file containing a unique VAR identification code. Without the VAR identification code, licensing of a source module is not possible.*

To license a source module, you must select the name of a license file that you want to associate with the module through the **License file** wizard . AIMMS will automatically enter the appropriate **Module code** if applicable. If you enter a string parameter in the **User data** attribute, AIMMS will assign the user data stored in the VAR license file to that string parameter for further use in the module.

*Licensing
attributes*

After you have turned a section into a licensed source module, AIMMS will automatically encrypt the associated source file, making it impossible for your end-users to read its contents when you distribute it. In addition, when an end-user includes your licensed module in his model tree, the subtree containing your code can no longer be opened. These restrictions do not apply to you as the developer of the module.

*Module
encryption*

21.3 User authentication

When an application is set up for use by multiple users, it is usually considered desirable that users have access to only those parts of the application that are of interest to them, and can be given or denied the right of access to each others data. AIMMS allows you to set up such a controlled environment around your model-based application. This section describes the security features available in AIMMS.

*User
authentication*

21.3.1 Introduction

In a multi-user environment, a log on procedure is commonly employed to identify and authenticate the particular user who wants to make use of a system at a particular time. Users can own distinct resources within the system, and can control the access of other users to such resources. In addition, this scheme is often extended using the notion of *user groups* to categorize users who share a certain characteristic (e.g. who work in the same department), and for that reason should be able to access each others data.

*Users and
groups*

Complementary to the distinction of users and user groups and their associated rights to access *data*, is the question of which rights should be assigned to a specific user in accessing particular *functionality* within a system. For instance, in an AIMMS application, one might want to restrict the access to particular end-user pages, not allow a user to make changes to the values of certain identifiers within an end-user page, or disable his ability to execute particular parts of the model.

Access rights

Rather than defining these access rights for every individual user, or for every user group, at a particular installation site, it often makes more sense to distinguish the several *roles* an end-user can play within an application, and link the access rights of a user to his role within the application. The number of roles that need to be defined for a particular application and their associated level of authorization, is usually fixed and relatively small.

End-user roles

To help you set up a flexible environment for providing security to your model-based application, AIMMS supports the concepts of authorization levels, model users and user groups as discussed above. To help you accomplish this task, AIMMS provides a number of security tools for the definition of authorization levels during application development, as well as for adding users and user groups once the application is installed at a particular end-user site.

*Security in
AIMMS*

Although authorization levels, users and user groups all play a role in securing an application, the responsibilities for their creation, use and administration are quite different.

Responsibilities

- The creation and change of authorization levels can only be carried out if the AIMMS project is opened in developer mode, as the set up of authorization levels with their associated rights is part of the design of a model-based application.
- The creation and modification of users and user groups is a task for a site-specific user administrator, and can also be performed if the project is opened in end-user mode.

21.3.2 Setting up and using authorization levels

You can associate authorization levels with your modeling application through the **Settings-Authorization Level Setup** menu, which is only available in development mode. It will open the **Authorization Level Manager** illustrated in Figure 21.6. In this dialog box you can add new authorization levels to your application by adding nodes to the list of existing authorization levels.

*Setting up
authorization
levels*

By default, during log on end-users of a protected AIMMS application will obtain the authorization level that has been assigned to them by the (local) user administrator. For every authorization level in your application you can specify a password that allows end-users to obtain an authorization level different from their default level during an AIMMS session. By double-clicking on an authorization level (or through the **Edit-Properties** menu) you open the **Properties** dialog box displayed in Figure 21.7. In the **Password** tab of this dialog box you can specify the password required to switch to this authorization level during a session (see also Section 21.3.4). If you do not specify a password,

*Password
protection*

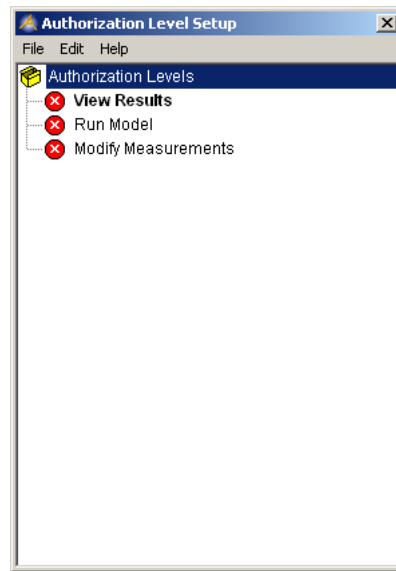


Figure 21.6: The **Authorization Level Manager**

end- users can switch to that authorization level as long as they have access to the **File-Authorization** menu.

Whenever you have authorization levels in your AIMMS project, one level is designated as the *default* level. In the authorization level tree, the default level will be shown in bold. Initially, AIMMS will make the first authorization level that you add to the tree the default level. You can modify the default authorization level using the **Edit-Set Default** menu. AIMMS uses the default authorization level for users to whom no authorization level has been associated (see also Section 21.3.3).

*Default
authorization
level*

Within an AIMMS model, you have access to all the authorization levels defined for the associated project through the predefined set AllAuthorizationLevels. In addition, the currently active authorization level is available through the predefined element parameter CurrentAuthorizationLevel in the set AllAuthorizationLevels. The value of this element parameter changes, whenever a user logs on to the application, or changes the authorization level during a session.

Use in the model

Using the predefined set and element parameter discussed above, you can set up your own customized authorization level based security scheme within your application. By defining your own subsets of, and parameters over, the set AllAuthorizationLevels you can specify conditions to check whether the current user is allowed to perform certain actions.

*Authorization-
based
security*

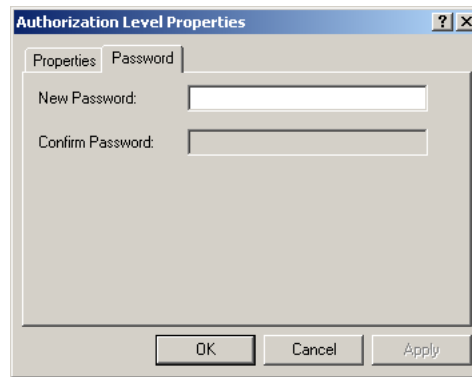


Figure 21.7: The authorization level **Properties** dialog box

Assume that `ExecutionAllowed` is a two-dimensional parameter defined over the set `AllAuthorizationLevels` and a user-defined set of `ActionTypes`. Then the following code illustrates the use of the element parameter `CurrentAuthorizationLevel` to allow or forbid a certain statement to be executed.

Example

```
if ( ExecutionAllowed(CurrentAuthorizationLevel, 'Solve') ) then
    solve OptimizationModel;
else
    DialogError( "Your authorization level does not allow you\n" +
                "to solve the optimization model" );
endif;
```

You can also use parameters defined over `AllAuthorizationLevels` to influence the appearance and behavior of the end-user interface. More specifically, the following aspects of an AIMMS end-user interface can be influenced through the nonzero status of (indexed) parameters:

Use in the interface

- the access to a page through the page tree-based navigational controls,
- the visibility of graphical (data) objects on a page,
- the read-only status of data in a data object, and
- the visibility and enabled/disabled status of menu items and buttons.

If such parameters are defined over `AllAuthorizationLevels`, these aspects can be directly linked to the permission appropriate for a specific authorization level by slicing over the element parameter `CurrentAuthorizationLevel`.

21.3.3 Adding users and groups

All user and group information associated with a particular AIMMS application is stored in a separate (encrypted) user database file. Before you can start adding users and user groups you must first link your application to an existing user database or create a new one. As users and groups are site rather

User databases

than model specific, all user management tasks can be performed from within both development and end-user mode of a project.

You can link to an existing user database, or create a new one, through the **Settings-User Setup-Link** menu. This will open a dialog box to let you select an existing or new user database file, with the `.usr` extension. If you select an existing user database which is password protected, AIMMS will only allow you to link to the user database after entering the correct password.

Linking to a user database

Through the **Settings-User Setup-Unlink** menu you can unlink a linked user database. If the user database is password protected, you can only unlink after entering the correct password. Thus, you can effectively prevent your end-users from circumventing the authentication procedure by unlinking the user database.

Unlinking a user database

When you distribute an AIMMS application to your end-user, it may be the case that your end-user wants to use a user database already linked to some other AIMMS application and share this user database among several AIMMS applications that he is using. The end-user however has no means to change the project. You, on the other hand, as a developer of the AIMMS application, can only establish a correct link if you have access to the user database of the end-user. Besides that, you may not want to store end-user specific information (i.e. location and name of the user database file) in the AIMMS project. To allow end-users of an application to specify another location for the user database without the need to change the AIMMS project itself, the existing user database file (e.g. local to the project directory) may be replaced by a single line ASCII file (with the same name and extension as the original user database file) containing a reference to the actual user database file.

User database forwarding

You can edit a user database linked to your application through the **Settings-User Setup-Edit** menu. After a password-check (if the user database is password protected), AIMMS will open the **User Manager** window illustrated in Figure 21.8.

Editing a user database

You can add a new user administrator password or change an existing one via the **File-Change Password** menu in the **User Manager**. Adding a user administrator password has the following effects that linking, unlinking and editing the user database is password-protected.

Password protecting a user database

You can add new user groups and users as new nodes in the user manager tree. As each user must be a member of a unique user group, you must first add one or more user groups to the user manager tree before you can add users. Figure 21.8 illustrates a user group and user configuration. Insertion,

Adding users and groups

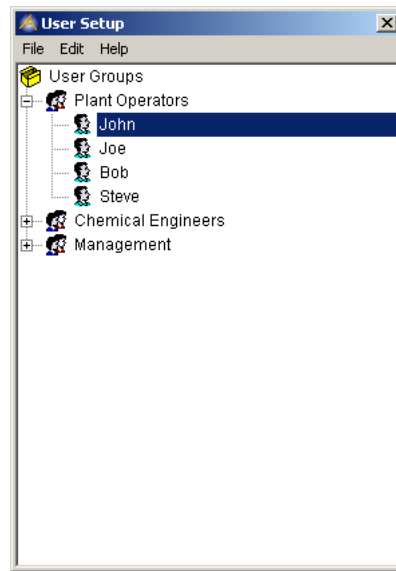


Figure 21.8: The **User Manager** window

deletion and modification of user and group nodes within the user manager tree is carried out in the usual fashion (see also Section 4.3).

The user group in which you position a user will become the *default user group* of that user. When a user logs on to an AIMMS application, he will automatically become a member of his default group. During a session, group membership can be modified through the **File-Authorization-Group** menu (see also Section 21.3.4). Group membership is only relevant in determining the access rights to case data (see also Section 21.4).

Default user group

The user manager in AIMMS lets you set up a hierarchical group structure. You can use it to set up a hierarchical protection scheme for case data by assigning the relevant access rights to members of parent and child groups (see Section 21.4).

Hierarchical group structure

For every new user or user group added to the user database, you can set its properties in the associated **Properties** dialog box illustrated in Figure 21.9. You can open the dialog box by double clicking on a user or user group, or through the **Edit-Properties** menu. In the dialog box you can specify properties such as

User and group properties

- the authorization level associated with an account, its expiration date and password,
- whether the user or user group concerned is allowed to enter the project in development mode,

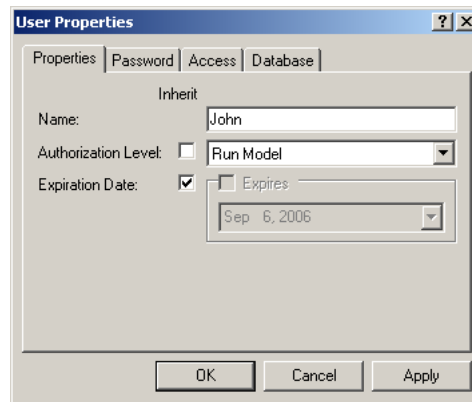


Figure 21.9: The user or user group **Properties** dialog box

- the default access rights for cases and datasets (see Section 21.4), and
- the default ODBC or OLE DB name and password associated with a user or group of users.

In the **Expiration Date** field in the **Properties** dialog box of a user account you can select an expiration date for the account using a calendar control. To remove an existing expiration date, you should uncheck the **Expires** check box. AIMMS will not allow the user to log on to the application when his account has expired.

Expiration date

In the **Authorization Level** field in the **Properties** dialog box of a user or user group you can enter the authorization level assigned to that user or user group. When you assign an authorization level to a user group, newly created user groups and users within that group will automatically inherit its authorization level. When you assign a default authorization level to a user, it will automatically be assigned to that user when he logs on to your application. During a session, a user can always override his current authorization level using the **File-Authorization-Level** menu (see also Section 21.3.4).

Authorization level

During user log on, AIMMS determines the authorization level applicable for that user in the following order:

Rules

- the authorization level of the user account itself,
- the authorization level of the first parent user group for which an authorization level has been specified, or
- the globally defined default authorization level (see also Section 21.3.2).

In the **Password** tab of the **Properties** dialog box of either a user or user group you can specify a password for that user or user group. If you check the **Password Required** check box, an end-user is not allowed to ‘enter’ an empty password through the **File-Authorization-Change User Password** menu. The user password is verified whenever a user logs on to your application. The user group password is verified when a user wants to change his user group during a session via the **File-Authorization-Group** menu.

Password

To every user or user group you can assign developer rights. Only users with developer rights are allowed to open a project, which is linked to a user database, in developer mode. If no user has been assigned developer rights, you can still open the project in developer mode by using the predefined “*User Administrator*” account. Note that restricting developer access by adding user authentication to your project does not prevent any of the project’s source components (like source files and/or libraries) from being used in another project. To make sure that all source components are also protected outside the context of the current project consider using user authentication in combination with one way encryption or VAR licensing.

Developer rights

In the **Database** tab of the **Properties** dialog box of users and user groups you can enter a user name and password which AIMMS will use for authenticating the ODBC/OLE DB connections of a user. The following rules apply.

ODBC/OLE DB user name and password

- If an ODBC/OLE DB user name and password have been specified for a user account itself, AIMMS will use these when authenticating an ODBC/OLE DB connection.
- Otherwise, AIMMS will inherit the ODBC/OLE DB user name and password of the first user group in which the user account is contained that provides an ODBC/OLE DB user name and password, and makes these available to all of its children (as indicated by the **Inherit** check box).
- If no ODBC/OLE DB user name and password are found in the previous steps, or when ODBC/OLE DB authentication fails, a log on dialog box will be presented.

Within the modeling language AIMMS provides access to the currently logged-on user and his user group. They are available through:

Use within model

- the string parameter `CurrentUser` holding the name of the user currently logged on, and
- the string parameter `CurrentGroup` holding the name of the currently active user group.

By default, AIMMS does not provide access to the entire set of users and user groups defined in the user database attached to a project, as this information is not necessary for most applications. However, if you need access to the set of all users and groups in your application, AIMMS offers the following two functions to obtain this information.

Obtaining all users and groups

- SecurityGetUsers(*user-set*[,*group*][,*level*])
- SecurityGetGroups(*group-set*)

The argument *group* is a string, the argument *level* an element of AllAuthorizationLevels, while the output arguments *user-set* and *group-set* are (root) set identifiers.

The functions SecurityGetUsers and SecurityGetGroups create new elements in the indicated sets for every user or group in the user database. When specified in a call to SecurityGetUsers the group and level arguments serve as filters, filling *user-set* with only those user names that are member of the given group and/or possess the given authorization level.

Calling semantics

21.3.4 Logging on to an AIMMS application

Whenever an AIMMS application has an associated user database, you must first log on before you can run the application. The **Logon** dialog box is illustrated in Figure 21.10. Initially, AIMMS will enter your Windows user name if

Logging on



Figure 21.10: The **Logon** dialog box

this name is also present in the user database. You can always close an application during the log on procedure by pressing the **Cancel** button, if you do not have a valid user account for the application. After logging on successfully, AIMMS will set the user group and authorization level to the values associated with the account of the currently logged on user.

The end-user will automatically be logged on to an AIMMS application if (1) the current Windows user appears in the application user database, (2) the application password of current user equals his Windows password, and (3) the application runs in end-user (or viewer) mode.

Automatic log on

Through the **File-Authorization** menu end-users can log off, or modify their current user group or authorization level if this is needed to read or write particular case data, or when additional authorization is required to perform particular tasks within the model. If you do not remove the (standard) **File-Authorization** menu from your application, you are strongly advised to password-protect all user groups and/or authorization levels to prevent unauthorized access by end-users.

*Switching
authorization or
group*

Through the **File-Authorization-Change User Password** menu end-users of your application can modify their password without needing the interaction of the user administrator. By checking the **Password Required** check box in the **Properties** dialog box of an end-user, you can prevent end-users from entering empty passwords.

*Changing
end-user
passwords*

21.4 Case file security

When your AIMMS-based application is used by multiple end-users all sharing the same data management tree, read and/or write protection of the individual datasets and cases may become a relevant issue. AIMMS offers such protection by associating cases and datasets with end-users in the user database.

*Protecting your
data*

As explained in Section 21.3, user groups in the user database can be ordered in a hierarchical fashion. All case file security in AIMMS is based on this hierarchy. More specifically, AIMMS allows you to assign different access rights to

Access rights

- the owner of the dataset or case,
- members of the group associated with the dataset or case,
- members of groups that lie hierarchically above the user group associated with the dataset or case,
- members of groups that lie hierarchically below the user group associated with the dataset or case, and
- all other users.

For each category of users you can separately specify read and write access to the case or dataset.

Only when you are the owner of a dataset or case, or are the local user administrator, will AIMMS allow you to modify the access rights previously assigned to a case. You can perform this task through the **Properties** dialog box of the dataset or case in the data manager. In the **Access rights** tab of this dialog box, which is displayed in Figure 21.11, you can change the associated user group, as well as the access rights for the each of the categories listed above.

*Modifying
access rights*

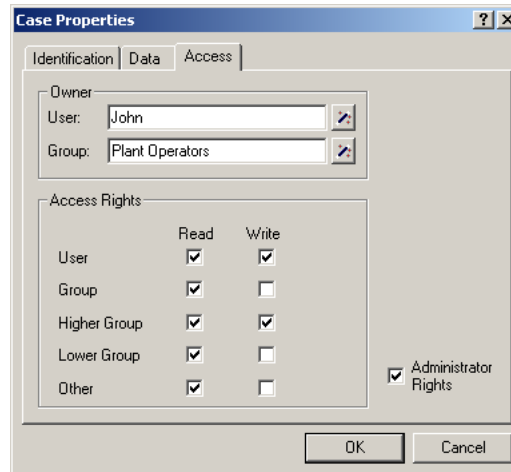


Figure 21.11: Access rights of a dataset or case

Normally, AIMMS will only allow you to modify the access rights of the datasets and cases you own. You can override this by checking the **Administrator Rights** check box displayed in Figure 21.11. This will open a password dialog box requesting the user administrator password associated with the end-user database. If successful, you can modify the access rights of any dataset or case as if you were its owner. With user administrator rights, you can even change the owner and user group associated with the case or dataset.

*Administrator
rights*

By default, any newly created dataset or case will be owned by the user that is currently logged on, and will be associated with the currently active user group (usually the group in which the end-user is placed in the end-user database). The access rights associated with such a dataset or case will be the default access rights assigned to the end-user in the end-user database by the local user administrator.

*Default access
rights*

You can specify the default access rights of an individual user or of an entire user group through the **Access** tab in the properties dialog box of either the user or user group at hand. In this dialog box, illustrated in Figure 21.12, you can either

*Specifying
default access
rights*

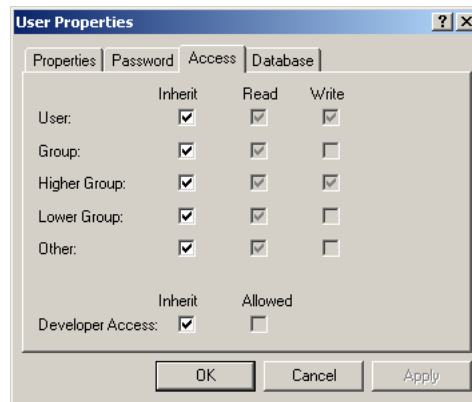


Figure 21.12: Specifying default access rights

- specify the specific access rights for a particular user category in a similar fashion as for a case or dataset itself, or
- indicate that you want to inherit the rights for a particular user category from the next higher user group.