# AIMMS 3.8

*Release Notes for Win32 Build 3.8.9*

Visit our web site www.aimms.com
for regular updates

Paragon Decision Technology

September 7, 2009

# Contents

# Chapter 1

# System Requirements

*System requirements*

This chapter discusses the system requirements necessary to run the various components of your Win32 AIMMS 3.8 system successfully. When a particular requirement involves the installation of additional system software components, or an update thereof, the (optional) installation of such components will be part of the AIMMS installation procedure.

## 1.1 Hardware and operating system requirements

*Hardware requirements*

The following list provides the of minimum hardware requirements to run your AIMMS 3 system.

- 1.6 Ghz or higher x86 or x64 processor
- XGA display adapter and monitor
- 1 Gb RAM
- 1 Gb free disk space
- A parallel or USB port (only when the hardware dongle is used)

Note, however, that performance depends on model size and type and can vary. It can also be affected by the number of other applications that are running concurrently with AIMMS. In cases of a (regular) performance drop of either AIMMS or other applications you are advised to install sufficiently additional RAM. If you are exceeding the limits of x86, you are advised to move to x64.

*Supported Windows versions*

The Win32 version of AIMMS 3.8 is designed to run under

- Windows 2000,
- Windows XP,
- Windows Vista, and
- Windows Server 2003.

The Win32 version of AIMMS 3.8 Unicode is supported under

- Windows 2000,
- Windows XP,
- Windows Vista, and
- Windows Server 2003

Running the Win32 version of AIMMS 3.8 versions under the 32-bit Windows versions 95/98/ME/NT 4.0 is not supported.

## 1.2 ODBC and OLE DB database connectivity issues

*Database connectivity*

Before you can start using the database connectivity features built into the Win32 version of AIMMS 3.8, the following ODBC and OLE DB components need to be installed on your computer:

- the Microsoft Data Access Components (MDAC) version 2.5 or higher, and
- an ODBC driver for each database from which you wish to retrieve data from within your modeling application through ODBC.
- an OLE DB provider for each database from which you wish to retrieve data from within your modeling application through OLE DB.

This section discusses the compatibility issues between the various available ODBC and OLE DB components and AIMMS 3.

*Microsoft Data Access Components*

The MDAC components, as well as the Microsoft ODBC Jet drivers (which allow you to connect to, for instance, Microsoft Access databases), and OLE DB providers for the most commonly used databases, are available through the Microsoft web site. Please note that early versions of the Data Access Components (or of the OBDC Jet drivers) may exhibit problems that prevent AIMMS 3 from successfully connecting to certain databases. If you are using the ODBC connection layer, you are therefore strongly advised to install the latest version of the Microsoft Data Access Components.

*Data Access Components installation*

The AIMMS 3 CD-ROM contains the installation executable of both Microsoft Data Access Components version 2.7 and the Jet 4.0 ODBC drivers and OLE DB providers. You can install these using the CD-browser which automatically starts up when you insert the AIMMS 3 CD-ROM into your computer. Alternatively, if you have downloaded the AIMMS 3 installation executable from the AIMMS 3 web site www.aimms.com, you can download the installation executable for the latest Microsoft Data Access Components from the Microsoft web site www.microsoft.com.

*Oracle ODBC drivers and OLE DB providers*

If your model needs to retrieve data from an Oracle database, the following information may be relevant to you. The Oracle provided ODBC drivers for Oracle version 7 only support ODBC version 2.0. If you are using Oracle version 7, you should therefore use the Oracle ODBC driver that is distributed with the Microsoft Data Access Components. The Oracle provided ODBC drivers for Oracle versions 8 and higher support OBDC version 3.0, and can therefore use the ODBC 3.0 connection layer. Oracle versions 8 and higher also provides a native OLE DB provider. You can download the installation executables of the most recent version of the ODBC drivers and OLE DB providers for Oracle 8 and higher from the Oracle web site www.oracle.com.

*Prefer native Oracle over MDAC drivers*

We discourage the use of the ODBC drivers and OLE DB providers for Oracle that are distributed with the Microsoft Data Access Components, as these drivers only support a subset of the functionality provided by the native Oracle drivers. In addition, in our internal tests, they appear to have issues with certain ODBC/OLE DB functionalities that are used by AIMMS.

*OLE DB issues*

If you are using OLE DB for database connectivity, the following issues apply:

- Access stored procedures cannot be called (yet) with the Jet 4.0 OLE DB provider. There is no work around either. Try to replace stored procedures by an AIMMS `DATABASE PROCEDURE`. Access queries, which are not marked as 'stored procedure' by Access, are perfectly well possible through the AIMMS `DATABASE PROCEDURE`.
- Oracle's own OLE DB provider for Oracle generally performs better than Microsoft's OLE DB provider for Oracle.
- Using `long` column types (such as `text` or `ntext`) with SQL Server, may reduce your application's performance dramatically. Therefore (and because those column types hardly make sense in an AIMMS context), try not to use them. If you feel you really need them, it may be a wise choice to specify an ODBC data source for the database tables that contain these columns.
- Using the OLE DB provider for ODBC may result in errors. Try to use the database manufacturers' own OLE DB provider as much as possible to get better performance and more stable behavior.

## 1.3 Viewing help files and documentation

AIMMS *documentation*

The AIMMS 3 User's Guide, the AIMMS 3 Language Reference, the AIMMS 3 Function Reference, and the AIMMS book on Optimization Modeling are available online as Adobe Portable Document Format (PDF) files. In order to view or print PDF documents, Adobe Acrobat Reader version 4.0 or higher needs to be installed on your computer.

*Acrobat Reader installation*

The AIMMS 3 CD-ROM contains the Acrobat Reader 6.0 setup executable, and you can install it using the CD-browser which automatically starts up when you insert the AIMMS 3 CD-ROM into your computer. Alternatively, if you have downloaded the AIMMS 3 installation executable from the AIMMS 3 web site www.aimms.com, you can download the Acrobat Reader 6.0 installation executable from the Adobe web site www.adobe.com.

# Chapter 2

# Installation Instructions

*Installation instructions*

This chapter discusses all relevant issues related to the setup of AIMMS 3 on your computer. To install AIMMS 3 on your computer, you can

- run the AIMMS 3 setup program from the AIMMS 3 CD-ROM, or
- download the setup program for the latest release of AIMMS 3 from the download area of our web site, and run it from your hard disk.

The latter is the standard procedure for obtaining and installing AIMMS 3 Software Updates. Note that in order to run AIMMS properly, AIMMS requires some specific 3rd- party library versions to be installed (e.g. Microsoft Visual Studio runtime library, Microsoft .NET Framework). In the rest of this text, these libraries are called the prerequisites for running AIMMS.

## 2.1 Installation instructions

*The AIMMS 3 CD-browser*

When you install AIMMS 3 from the AIMMS 3 CD-ROM, a CD-browser will be started automatically when you insert the AIMMS 3 CD-ROM into your computer. Using this CD-browser, you can

- start the AIMMS 3 or AIMMS 3 Unicode setup program (these setup programs take care of installing the prerequisites as well),
- install any other third-party software or system components available on the AIMMS 3 CD-ROM which are required to use certain components of AIMMS 3 (as described in Chapter 1), and
- start the AIMMS 3 License Server setup program (for network licenses only).

*Obtaining AIMMS 3 from the web*

The download area of our web site www.aimms.com provides the AIMMS 3 setup program of the latest AIMMS 3 releases. If you have selected the AIMMS setup program that you want to install, the corresponding prerequisites are selected as well. Note that pressing ’Download.msi’ does not include the prerequisites. Pressing the ’Download’ button does result in all selected files (including prerequisites) being downloaded.

*The AIMMS 3 setup program*

The AIMMS 3 setup program will guide you through the various steps that are necessary to successfully install AIMMS 3 on your computer. The AIMMS 3 setup program requires some prerequisites to be installed. When you install AIMMS 3 from an AIMMS 3 CD-ROM or from a downloaded package file, the AIMMS 3 setup program automatically takes care of of installing the prerequisites.

*Component selection*

The AIMMS 3 setup program requires that you provide the specific selection of AIMMS 3 components you wish to install (the setup will select all common AIMMS features by default (Typical), which is recommended)

*Solvers in installation*

In Table 2.1 you can find an overview of the solvers that are installed during a Typical and a Complete AIMMS installation on the several platforms on which AIMMS is supported. Note that it depends on your license whether you can use these solvers.

*Administrative privileges required*

Certain parts of the setup of AIMMS 3 may require administrative privileges. More specifically, you will need administrative privileges for

- the installation of the device drivers required to access the AIMMS hardware dongle (Windows only),
- the installation of, or updates to, any system components which access areas of the Windows registry in which you have no write access, or which write to the Windows system directory, if write access to that directory has been restricted, and
- the creation of writable directories in the common application directory on your computer where AIMMS will store its license configuration and any nodelock files associated with your AIMMS system, if write access to the common application area of your computer has been restricted.

## 2.2 AIMMS 3 licensing

*AIMMS 3 licensing*

AIMMS offers the following two types of licenses:

- single-user licenses, and
- network licenses.

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

*Single-user license protection*

Single-user licenses can be used by a single user on a single computer. To enforce the single-user character, AIMMS 3.8 requires that single-user licenses be protected by either

- a hardware dongle, which, depending on the type of dongle, must be connected to a USB or parallel port of your computer, or

| | | Win 32 | | Win 64 * | Linux 32 * | Linux 64 * |
|---|---|---|---|---|---|---|
| **Solver** | **Version** | **Typical** | **Complete** | | | |
| AOA | | √ | √ | √ | √ | √ |
| BARON | 7.5.1 | - | √ | - | - | - |
| | 7.5.2 | - | √ | - | - | - |
| | 7.5.3 | √ | √ | - | - | - |
| CONOPT | 2.071 C | - | √ | - | - | - |
| | 3.11 B | √ | √ | - | - | - |
| | 3.14 A | √ | √ | - | √ | - |
| | 3.14 G | √ | √ | - | - | - |
| | 3.14 M | - | - | √ | - | - |
| | 3.14 Q | - | - | - | - | √ |
| CPLEX | 8.1 | - | √ | - | - | - |
| | 9.0 | - | √ | - | - | - |
| | 9.1 | - | √ | √ | √ | - |
| | 10.0 | - | √ | √ | √ | √ |
| | 10.1 | √ | √ | √ | √ | √ |
| | 11.0 | √ | √ | √ | √ | √ |
| | 11.1 | √ | √ | √ | √ | √ |
| | 11.2 | √ | √ | √ | √ | √ |
| KNITRO | 5.1 | √ | √ | √ | √ | √ |
| | 5.2 | - | √ | √ | √ | √ |
| LGO | 1.0 | √ | √ | - | - | - |
| MINOS | | √ | √ | √ | √ | - |
| MOSEK | 3 | - | √ | - | - | - |
| PATH | 4.4a | - | √ | - | - | - |
| | 4.6 | √ | √ | - | - | - |
| SNOPT | 6.1 | √ | √ | √ | √ | √ |
| | 7.2 | - | √ | √ | √ | √ |
| XA | 13 | - | √ | - | - | - |
| | 14 | √ | √ | √ | √ | √ |
| | 15 | √ | √ | √ | √ | √ |
| XPRESS | 15 | - | √ | - | - | - |
| | 16 | - | √ | - | - | - |
| | 17 | - | √ | - | - | - |
| | 18 | - | √ | - | - | - |

* Typical and Complete installations are the same for Win 64 and Linux 32/64.

Table 2.1: Solvers included in the several types of AIMMS installations

- a nodelock file, which must be activated to match the hardware characteristic of your computer.

*You can choose*

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 supports dongles.

*Pros and cons of dongles*

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 nodelocks*

Nodelock files are stored on the harddisk of your computer, and are, therefore, much less vulnerable to loss. Only if you 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.

*Physical shipments*

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.

AIMMS *dongles*

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 3.8 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.

*Network licenses*

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.

### 2.2.1 Personal and machine nodelocks

*Two types of nodelocks*

Aimms offers 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 nodelock*

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 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 nodelock*

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 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.

*Privacy*

Although a personal nodelock makes a regular connection to our internet license database for nodelock renewal, we do respect your privacy and will not register patterns in your 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.

*Internet connection and proxy settings*

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.

*Automatic configuration scripts*

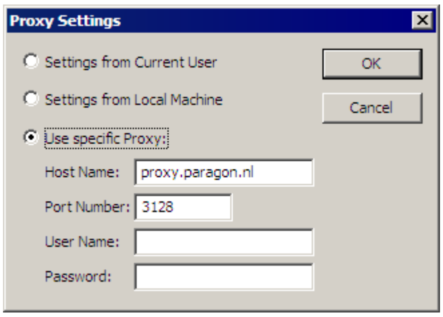
It should be noted that the use of auto-configuration scripts in determining the proxy server will fail if these use any other scripting language than Javascript. This is due to the libraries underlying the SOAP library used by AIMMS to connect to our license server. If you are in this situation, you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

*Manual proxy setting*

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 2.1. In this dialog box you can choose between

- 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.



○ Settings from Current User
○ Settings from Local Machine
● Use specific Proxy:
Host Name: proxy.paragon.nl
Port Number: 3128
User Name:
Password:

Figure 2.1: The AIMMS **Proxy Configuration** dialog box

### 2.2.2 Installing an AIMMS license

*Managing your AIMMS licenses*

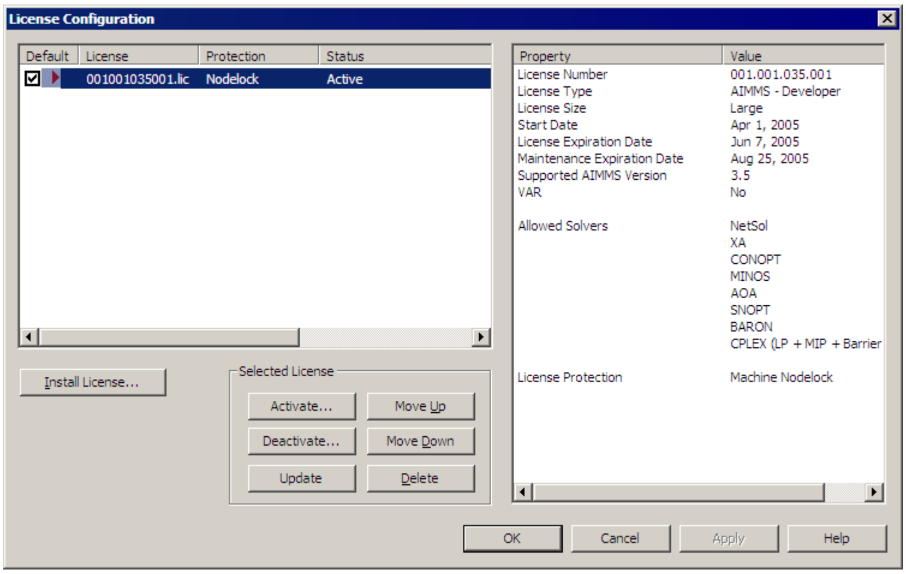
When you start up AIMMS 3.8 for the first time after installation, AIMMS will open the **License Configuration** dialog box illustrated in Figure 2.2. Through this dialog box you can install new AIMMS licenses and manage all AIMMS licenses that already have been installed on your computer.

*Installing a new license*

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

- existing AIMMS 3.7 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.



| Default | License | Protection | Status |
|---|---|---|---|
| ☑ | 001001035001.lic | Nodelock | Active |

| Property | Value |
|---|---|
| License Number | 001.001.035.001 |
| License Type | AIMMS - Developer |
| License Size | Large |
| Start Date | Apr 1, 2005 |
| License Expiration Date | Jun 7, 2005 |
| Maintenance Expiration Date | Aug 25, 2005 |
| Supported AIMMS Version | 3.5 |
| VAR | No |
| Allowed Solvers | NetSol<br>XA<br>CONOPT<br>MINOS<br>AOA<br>SNOPT<br>BARON<br>CPLEX (LP + MIP + Barrier |
| License Protection | Machine Nodelock |

Figure 2.2: The **License Configuration** dialog box

*Single-user nodelocked licenses*

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

- 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 2.2.1 for a more detailed introduction to personal and machine nodelocks.

*Single-user dongled licenses*

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

- 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.

*Network licenses*

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

- 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).

*Evaluation licenses*

To install an AIMMS evaluation license you need the following information

- 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.

*Student licenses*

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

- 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.

### 2.2.3 Managing AIMMS licenses

*Managing multiple* 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:

- 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.

*Default licenses*

In the **License Configuration** dialog box displayed in Figure 2.2, 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.

*Transferring 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

- 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.

*Emergency nodelocks*

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.

### 2.2.4 Location of license files

*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 2000, 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, 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.

*Do not move nodelock files*

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.

*User specific configuration*

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 3.8 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.

*Any User*

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.

*Current 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 2000, Windows XP and Windows Server 2003, this local folder is usually located at:

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

On Vista it is:

```
C:\Users\<UserName>\AppData\Roaming\Paragon Decision Technology
```

*Accessing configuration files*

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.

*Version dependent configurations*

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.

*Network License Client Files*

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:

- 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.

## 2.3 OpenSSL license

*OpenSSL included - in separate RPC installation only*

The separate *RPC* installation file, which you can download in addition to the main AIMMS software, contains the OpenSSL and SSLeay libraries to provide the HTTPS transport for AIMMS web services. The following licenses apply to OpenSSL and SSLeay.

*OpenSSL license*

Copyright (c) 1998-2006 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:
   ”This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)”

4. The names ”OpenSSL Toolkit” and ”OpenSSL Project” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core @openssl.org.
5. Products derived from this software may not be called ”OpenSSL” nor may ”OpenSSL” appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment: ”This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@ cryptsoft.com). This product includes software written by Tim Hudson (tjh@ cryptsoft.com).

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved. *SSLeay license*

This package is an SSL implementation written by Eric Young (eay@cryptsoft. com). The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young’s, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

# Chapter 3

# Getting Support

*Getting support*

This chapter discusses the procedures that apply when you encounter problems in AIMMS 3. Please refer to the list of known and reported issues at the end of this chapter to verify whether a problem which you wish to report has already been reported before.

## 3.1 Reporting a problem

*Problem fixing*

When you encounter a problem in AIMMS 3, Paragon Decision Technology (PDT) will investigate the problem, and if applicable provide a fix for it. PDT always provides this service if your AIMMS version is the latest functional AIMMS release. In all other cases, you are only eligible to obtain a fixed release if your AIMMS license is in maintenance. Any problem fix will always be performed on the latest functional AIMMS release only, and may require that you upgrade your AIMMS system to that version.

*Reporting a problem*

Whenever you encounter a problem, PDT needs the following information to process the problem.

- **Computer information**
  - Brand and model
  - Operating system and version number (including any installed Service Packs)
  - CPU type and speed
  - Amount of installed RAM
  - MDAC and ODBC/OLE DB driver versions if the problem is database-related
- **AIMMS information**
  - License number (**Tools-License-License Configuration** menu)
  - AIMMS build number (**Help-About AIMMS** menu)
- **Problem details**
  - A detailed description of the problem
  - The type of the problem
    - Crash
    - Incorrect functionality
    - Cosmetic

        - ∗ Feature request
    - The severity of the problem
        - ∗ Crash
        - ∗ No work around available
        - ∗ Work around available
        - ∗ Cosmetic
- **Reproducibility** (if applicable)
    - A description of the steps required to reproduce the problem
    - A copy of your project files along with any other files used by your project, if such is necessary to reproduce the problem

*Problem processing*

You should e-mail your problem report containing the above information to Support@aimms.com. Upon receipt, PDT will investigate your report and notify you of its status, as well as the actions that will be undertaken to fix the problem. You will be notified when an AIMMS 3 version will be released, in which the problem is fixed.

*Modeling support*

Support requests other than bug reports will, in principle, only be dealt with at our regular consulting fee. This is especially true for when you request us to provide extensive modeling support. If you are new to AIMMS 3, and need some quick pointers to help you tackle a particular modeling problem, we may decide to honor such requests at our discretion. When you send support requests to our e-mail account Support@aimms.com, please always include your AIMMS license number.

*Automatic Error Reporting*

Reports collected via the AIMMS Error Report Service are actively used by Paragon Decision Technology to continuously improve the quality and reliability of AIMMS. This service is automatically called when a severe/fatal error appears. At Why Report Errors you can read why you should report these errors. At Data Collection Policy you can read what information is collected by the AIMMS Error Report Service and what we do with this information.

## 3.2 Known and reported issues

*Known issues*

The following issues are known in the present AIMMS 3 release.

*RPC*

- Due to a change in the AIMMS 3.8.6 RPC layer, just after the installation of the AIMMS 3.8.6 RPC layer, you might encounter an error message about the AIMMS handler service. Please see this Knowledge Base article.

*Windows Vista*

- Deploying a web service with attachments in Windows Vista, with a manually started Aimms agent project, may result in temporary attachment files not being removed from the system's temporary directory. This happens because the `AIMMS HTTP/RPC Handler Service` runs under the `SYSTEM` account, while a manually started Aimms agent project usually does not. If the non-`SYSTEM` account under which the Aimms agent project runs does not have deletion rights on the `SYSTEM` temporary directory (which usually is the case), the temporary attachment files cannot be deleted. There are two ways to work around this problem:
  1) By using an automatically started Aimms agent (which, in a deployment scenario, is the preferred way of starting up Aimms agents).
  2) By stopping the `AIMMS HTTP/RPC Handler Service` and instead starting the `AimmsServiceHandler.exe` executable. This file is located in the Aimms `Common\Bin`-folder.
- On Windows Vista, the screen (progress window in particular) is not updated during execution once Aimms lost focus, by switching to a different application.

# Chapter 4

# Release Notes

*Release notes*

This chapter contains a description of the new features developed for AIMMS 3.8. In addition, it contains the release notes of all AIMMS 3.8 releases. For each release, the following items are listed:

- the build number of the release,
- the release date, and
- a description of the changes and the issues resolved in this release.

*AIMMS 3.8 versions*

By default, the installation procedure will install the single-byte AIMMS executable. This version is sufficient for most application developers. However, if you need to distribute a localized version of an AIMMS application to, for instance, Asian or Russian end-users, use of the common single-byte AIMMS version may not be sufficient anymore, as many languages in these regions cannot be represented by means of single-byte characters. To support localization to such languages, a Unicode version of AIMMS is available, in which all strings are represented internally through double-byte characters. More details on the AIMMS Unicode version can be found in Section 23.2 of the User's Guide.

An overview of all known and reported issues which are still outstanding in the current release is given in Section 3.2.

## AIMMS 3.8 build 9, release date 2009-09-07

*Build 3.8.9 2009-09-07*

In this AIMMS 3.8 Software Update, the following issues have been fixed or improved.

*Issues . . .*

- After a recent Windows Update, AIMMS displayed a duplicate UTC time zone info warning upon project start-up.
- It was not possible to open the AIMMS Help file anymore on systems with Internet Explorer version 8 installed.
- Sometimes, during a long run, AIMMS gradually became slower.
- Using the sparsity modifier ($) on the iterative operator `Mean`, could lead to a severe internal error during compilation.

- When using an end-user license tied to a specific VAR license, starting AIMMS with the project specified as a command line argument, did not work.
- Using a procedure upon double-click in a **Composite Table**, with an element parameter as argument, this argument would always keep the value it had upon opening of the page. Therefore, the procedure was not run when double-clicking on another element.
- The function `GMP::SolverSession::AsynchronousExecute` could solve an incorrect model during a re-solve after doing GMP modifications.

## AIMMS 3.8 build 8, release date 2009-06-23

*Build 3.8.8 2009-06-23*

In this AIMMS 3.8 Software Update, the following issues have been fixed or improved.

**Important** ...

- Due to a fix, it is possible that you encounter the warning *There is a difference in execution between* AIMMS *3.8 and former versions of* AIMMS when running your model in this new AIMMS version, followed by an explanation that an element is not in a domain set. This warning occurs because the AIMMS execution engine now updates the definitions of domain sets before and during execution of for loops. Be aware that this change can result in a different output of your model. In case this warning occurs we advice you to read the help on the options `Execution_respects_declaration_domain` and `Warning_difference_execution_respects_declaration_domain` and make a conscious decision on how to resolve the warning.

*... GUI*

- If the content of a **Pivot Table** is given via an implicit set of identifiers, then clicking on checkboxes in the grid area caused a fatal error.
- If an updated minimum or maximum date in the **Date Time Picker** object resulted in an invalid current date, the selected date was updated automatically. However, the underlying AIMMS identifier for the selected date was not updated in such cases.
- In rare cases, when using a High/Low/Open/Close chart, switching between properties could cause a fatal application error.
- Suffices for arcs and nodes, like the `.level` suffix, were not colored correctly by the syntax editor.

*... Solving*

- A defined parameter that was used as a reference in a **Nonvar**, **Priority**, **Relax**, or **Range** attribute of a variable might not be updated if the associated suffix of that variable was referenced in an assignment.
- The performance of consecutive calls to the procedure `GMP::Instance::CalculateSubGradient` has been improved.

- The Iterations Callback was not passing the correct number of iterations as long as no integer solution was found yet by **CPLEX**.
- The procedures `GMP::Solution::SendToModel` and `GMP::Solution::SendToModelSelection` did not calculate the level values of inline variables.

*. . . Databases*

- AIMMS did not handle MS SQL Server database columns of type `varchar(max)` and `nvarchar(max)` well, resulting in truncated strings being returned from the database.
- A non-empty database table could be emptied when the primary key(s) did not match with the database mapping of indices in AIMMS, and using the `replace rows` write mode.

*. . . Compiler and Execution*

- Using `%e and/or %I` in the first (format) argument of `StringToMoment` could lead to ignoring the hours/minutes/seconds information.
- For some time zones the DST transition in fall was reported one hour too late.
- A severe internal error could occur when a single element of an indexed parameter was displayed in a `Display` statement using an indexed unit parameter as unit.
- The performance of assignment statements of the following form has been improved:
  `a((t+1,i,j,k)|cond(t,i,j,k)) := ... ;`
  whereby the condition `cond(t,i,j,k)` is true for the first possible values of `t,i,j` and `k`.
- AIMMS could give a severe internal error on evaluating expressions of the form `sum( j in IndexedSet( ..., 'quoted element', ... ), ... )`.

*. . . Linux*

- The Linux version could complain about non-existing log files when the option `Number_of_logfile_backups` was moved from its default of 0 to a higher value.
- On Linux machines a number might be rounded differently than on Windows machines.

## AIMMS 3.8 build 7, release date 2009-04-21

*Build 3.8.7 2009-04-21*

In this AIMMS 3.8 Software Update, the following issues have been fixed or improved.

*Issues . . .*

- The option `Memory_garbage_collect` has been removed, since it decreased performance significantly and the memory gain was only minimal. In case of problems caused by allocating huge amounts of memory, you are advised to tune AIMMS with the various `rebuild` options in the `Memory_management` option category.

- In the **Profiler**, the maximum number of expression evaluations of assignments inside loops, could be computed erroneously, causing the sparsity to be incorrect.
- When using nodelock licenses, `NdLckSrv.exe` could hang when closing Aimms.
- `ExcelAssignTable` did not work correctly in case of ordered sets.

*. . . GUI*

- The **GIS** background in the **Network** object did not filter invisible layers during printing.
- In the **Network** object, **GIS** backgrounds that were retrieved from Yahoo or Microsoft Virtual Earth did not print correctly.
- When using a minimum and/or a maximum date in a **Date Time Picker** object and updating both the date value and the minimum and/or maximum value at the same time, could lead to the date value to be compared with the old minimum or maximum value. This could lead to a non-updated date value.
- When using dual monitors with the second monitor at the left of the main monitor and Aimms running in the second monitor, then a right mouse popup menu did not always appear at the correct location.
- When during startup the **Data source** attribute contained an invalid data source name, the **Data source** wizard could not be opened.

*. . . Pivot table*

- Columns in the **Pivot Table** for which no explicit column width has been specified but for which a default column width was present (in the form of an updatable identifier) could not be resized.
- In some situations, loading a large data set into an existing sorted **Pivot Table**, could lead to a large amount of memory being used, possibly resulting in a fatal application error.
- The option **only show rows/columns with differences** did not always filter out identical rows/columns correctly.
- Sorting in large **Pivot Tables** could sometimes lead to a stack overflow.

*. . . Solving*

- The **Math Program Inspector** did not show node numbers on the **Search Tree** tab when the option `Show_branch_and_bound_tree` had value `No`.
- Interrupting solver sessions using `GMP::SolverSession::Interrupt` from within a `for` loop could cause severe internal errors.
- Compiling the statement `Empty AllVariables.Stochastic;` could result in a severe internal error.

*. . . Compiler and Execution*

- The `GMP::Instance::CalculateSubGradient` function did not work for variables declared inside a library.
- Aimms did not do a unit conversion before checking data in the **Initial data** attribute of indexed parameters.
- Aimms did not do a range check on scalar parameters after modifying the data in the **Initial data** attribute.

- Name changes to identifier names that also have been deleted, could lead to situations in which that name could not be reused for a new identifier.
- In AIMMS 3.8.5 and AIMMS 3.8.6, working with cases containing string parameters and assigning different strings to the same parameter could eventually lead to a severe internal error.
- AIMMS now ensures that the definition of a parameter in the **Assert limit** attribute of an assertion, is evaluated before the value is used.
- The unnecessary **Protected** attribute of the multi-agent module could cause one-way encrypted projects that use this module, to become corrupted.
- Using the iterative operator `Count` in a macro could lead to a severe internal error.
- AIMMS 3.8.6 could do a duplicate unit scaling of the default value.

*... Linux*

- The environment variable, used to determine the search path to find the shared object (`.so` file) on **Linux** containing the external procedure, is now `LD_LIBRARY_PATH` instead of `PATH`.

## AIMMS 3.8 build 6, release date 2009-03-12

*Build 3.8.6 2009-03-12*

In this AIMMS 3.8 Software Update, the following issues have been fixed or improved.

*Improvements*

- The **NodeLockTool** for Linux can now perform online activation.
- The performance of Hessian evaluations used by **CONOPT** and **KNITRO** has been improved.

*Issues ...*

- The **Language Reference** used the non-existing function `ElementToString` in a few code examples. The occurrences of this function have been replaced by the equivalent `FormatString("%e", elementpar)`.
- Installing the **License server** on a Windows 2000 machine sometimes failed with an error message during the installation.
- A **View Window** in the **Model Explorer** was accidentally closed upon the **File - Save** command.
- The **MultiStart** and **StochasticDecomposition** modules were missing in the AIMMS Viewer installation. This caused projects including these modules to give an error upon opening it with the AIMMS Viewer.
- When running AIMMS with `--as-server`, the **Error Report Service** dialog box could still be displayed.
- The AIMMS **Excel Add-In** needs two support files of which one was not always correctly registered, sometimes leading to error messages while using the Add-In.
- Closing an AIMMS project via the AIMMS API could result in AIMMS not responding for a while when a previous procedure run resulted in a run-

time error.

*. . . GUI*

- If an **Assertion** failed in a **Gantt Chart** object, the page was not refreshed correctly.
- When using multiple identical GUI expressions on a page, now only one instance of this expression is created and evaluated, resulting in better performance.
- When a click label was displayed in a **2D Chart** ActiveX object, and the model was changed such that no data remained to be displayed, a fatal application error could occur.
- Sometimes, when having a **2D Chart** object with a click label displayed open on a page, changing any underlying data which affects the number of points/series displayed in the chart, resulted in a fatal application error.
- In the **Colors** tab of the page object properties, the command **set as default** did not work correctly if the color was not changed first.
- Clicking on a check box in a **Composite Table** while the value is read-only, no longer generates an error dialog.

*. . . Pivot table*

- The **Pivot Table** could be drawn incorrectly after the row or column tree had been re-sorted.
- After changing the implicit identifiers subset of a **Pivot Table**, some properties (like the text color, background color and number of decimals) were not always displayed correctly.
- In some situations, loading a large data set into an existing sorted **Pivot Table**, could lead to a large amount of memory being used, possibly resulting in a fatal application error.
- AIMMS could come up with an internal error when trying to update element parameters corresponding to the selected cell in the row or column tree of a **Pivot Table**.

*. . . Libraries*

- Using the same specified name for a local declaration section (different from `Declaration`) of procedures in a library, could cause problems during the adding of that library to another project.
- In a `read from table` statement without explicitly mentioning the identifiers to read, AIMMS tries to match the column names in the database with identifiers in the model. The searching process for identifiers in the model now starts with the namespace in which the database table has been declared. Similar behaviour holds for the `write to table` statement.
- A corrupted Template Duplicate could be the cause that the **Cross Library Reference** dialog box could not be opened.

*. . . Solving*

- When regenerating a non-linear program after a variable had been frozen, superfluous elements could appear in the constraint listing and in rare cases this could lead to infeasibilities.
- When you have a mathematical program with a specified variable set and a specified constraint set and between two solves a change is made in the variable set, but no change is made in the constraint set, then this could result in the second solve considering all constraints instead of the specified constraint set.
- If the **CPLEX** option `Global_thread_limit` is greater than 1 and the option `MIP_search_strategy` is set to `Automatic` or `Apply dynamic search`, then AIMMS will always set the option `Parallel_mode` equal to `Deterministic` for **CPLEX 11.2**, **11.1** and **11.0**. This is done to work around a **CPLEX** issue.
- The function `GMP::Instance::GenerateStochasticProgram` now issues an error message when the set of stochastic variables is empty.
- Calling the function `GMP::Column::Add` on a GMP created via the function `GMP::Instance::Copy` could lead to a severe internal error.
- AIMMS maintains a mapping between variables and actual columns in a Generated Mathematical Program, say `g`. When `g` is regenerated, this mapping is cleared by visiting all variables mentioned in the constraints of the mathematical program from which `g` is generated. In addition, AIMMS allows `GMP::Column::Add` to add columns from any variable, including variables not yet mentioned in the constraints of the mathematical program. Regenerating `g` and then again adding columns from variables not mentioned in the constraints of the mathematical program could lead to inconsistencies in the mapping and consequently to severe internal errors.

*. . . Compiler and Execution*

- Saving a model via the **File - Save as** command, could corrupt the model when the model uses VAR encryption.
- Using the `Exists` operator in the definition of inline variables could lead to an unnecessary compilation error.
- A commented out activating condition of an indicator constraint could cause a severe internal error.
- Setting the option `Initial_column_reservation` to an extremely high value (higher than 10485760), could cause a severe internal error.
- In debugger mode, a severe internal error could appear when a data page of a local parameter with dimension three or higher was opened, and subsequently the execution was halted.
- During a `read from file` statement, AIMMS should stop with an error when it encounters an undeclared identifier in the file. During a `read id1, id2, ... from file` statement however, AIMMS should continue when it encounters an undeclared identifier in the file. In AIMMS 3.8.5, AIMMS erroneously continued in both situations.
- AIMMS removed the prefix of a string parameter used inside the **Dll name** attribute of an external function when the string parameter and external function were declared inside different namespaces.

- When the function First is the first use of a locally declared set without **Index**, **Parameter**, **Subset of** or **Initial data** attribute, the AIMMS compiler issued a superfluous error message.
- The range (-1e150,1e150) was not accepted for objective variables.
- Applying the ElementCast function to an element of a compound set caused a severe internal error.
- When the index domain of an indexed identifier referenced an index in a horizon and did not have an index domain condition, changing the index domain to reference another index in a horizon could lead to a severe internal error.
- Using a period as placeholder for an index of a parameter inside an empty statement could lead to a severe internal error.
- Unnamed sections in .aim files caused AIMMS to infinitively search for a line end.
- Passing scalar data to external functions requiring runtime unit checking, could cause severe internal errors after (partial) recompilation of a model.
- Exporting a section to a .amb file, deleting that section and then re-importing the .amb file, could confuse AIMMS regarding deleted domain sets used in the index domain of identifiers.
- Re-opening a project after it had execution errors during data read, could cause a severe internal error.
- A syntax error in the **Arguments** attribute of a single argument internal function, could lead to a severe internal error.
- The result of FindNthString and StringOccurrences could be incorrect when the IgnoreWhite argument was set to true and the Key argument was prefixed with whites.
- Superfluous semicolons in the activating condition of a constraint were not filtered out when writing to a .aim file.
- AIMMS should only flag the use of functions inside constraints as an error when the function is an internal function (functions with a body declared inside the application), while it should not flag the use of intrinsic functions such as ElementCast inside constraints (functions predeclared by AIMMS) as an error.
- Using a single quoted element as the definition of an indexed set, could cause a severe internal error.
- A variable with a definition could prevent the creation of another variable with the same name but in another namespace. In addition, when both variables have a definition, changing one definition would also change the other definition.
- AIMMS could crash on assertions with comment only definitions.
- Reading an empty file with FileRead could cause a severe internal error.
- The intrinsic function FileRead could not be called more than 505 times in one AIMMS session.

*. . . Units of Measurement*

- The compiler could issue a superfluous error message when using a unit cast inside the domain condition on the LHS of an assignment.
- The values assigned to parameters with an integer range and a unit are now verified for integrality after converting the data with regard to the unit.
- A data definition of a unit parameter should be of the form `data{ 'e1' : [a_unit] }` where `[a_unit]` is a unit declared via a quantity. AIMMS now gives an error when the data definition of a unit parameter references to another unit parameter.
- The numeric values after scaling and translation of default values with temperature units, are now displayed in **Tables** and **Pivot Tables**.
- Zero temperature values with base unit `[K]` but displayed in `[C]` or `[F]` in **Tables** or **Pivot Tables**, are now displayed.
- Summing over an `if-then-else` expression in a linear constraint, could lead to temporary intermediate linear coefficients with value 0. These zero coefficients are eliminated at the end of generation, but during their lifetime they could confuse the runtime unit checking system.

*. . . Linux*

- AIMMS on Linux could not load all components when Linux was started in Security Enhanced Mode (SELinux).
- Accessing databases via ODBC could cause a crash on Linux when closing the project.
- On the Linux platform, backslashes in the filenames of files opened by the `write` statement, were not translated into forward slashes.
- In some situations, AIMMS projects run via the RPC service on Linux could crash.
- A `.amb` file with a developer password could not be opened when using `AimmsCmd` under Linux.

## AIMMS 3.8 build 5.11246, release date 2009-01-07

*Build 3.8.5.11246 2009-01-07*

In this AIMMS 3.8.5 Patch, a database issue has been fixed. We strongly advise you to upgrade to this Patch.

*Issue*

- Sometimes, when reading data from an **SQL Query** in a **Database Procedure**, not all rows of data were read from the database.

## AIMMS 3.8 build 5.11136, release date 2008-12-17

*Build 3.8.5.11136 2008-12-17*

In this AIMMS 3.8.5 Patch, the following two issues have been fixed. We advise you to upgrade to this Patch.

*Issues*

- Dumpfiles generated with the Error Report Service of AIMMS 3.8.5.11005 contained a problem, which caused them to be of no use for our development team.
- AIMMS 3.8.5.11005 crashed when the **AIMMS COM** object was called and the `as-server` argument of the `ProjectOpen` call was set to 1. This problem also appeared when starting AIMMS via the command line using the `as-server` command line option.

## AIMMS 3.8 build 5, release date 2008-12-03

*Build 3.8.5 2008-12-03*

In this AIMMS 3.8 Software Update, the following issues have been fixed or improved.

*Improvements*

- There is a new version of **CPLEX**: **CPLEX 11.2**. This version offers more control for solution polishing.
- There is a new version of **XA**: **XA 15**.
- **Profiler** information is now also available for the **index domain** attribute.

*Issues . . .*

- The links to **WSE 2.0** in the **Web Service Caller** and **CD Playing Time Optimizer** examples have been updated.

*. . . GUI*

- The (auto-suggest) dropdown list for editing the value of an element parameter did not disappear when using **Alt-Tab** to navigate to another Windows application.
- Within a **Library**, when using expressions in the inner page of an indexed page object, the expression was not evaluated correctly.
- Specifying an invalid proxy in the properties of a **GIS** source could result in a fatal application error.
- In a **2D Chart** with a click label displayed in it and in which data is displayed for a linked element parameter, changing the element parameter such that a chart containing no data should be displayed, could cause AIMMS to close itself.
- Disabling a layer (through the **Show Layer** property) on the background of a **Network** object that was the only layer of a WMS source, did not remove the layer from the background.
- A **Navigation** object within a **Tabbed Page** object or within an **Indexed Page** object was not working correctly. It could even lead to a crash if the **Navigation** object contained a **Close Page** action.
- When using `PageGetActive` in a procedure that is called from an object on a **Tabbed Page** or **Indexed Page**, it now always returns the containing page.
- If AIMMS was last displayed on a dual monitor and AIMMS was restarted in a single monitor setup, then the position of the AIMMS window was outside the boundaries of the main monitor.

- The auto-suggestion list that appears when editing the value of an element parameter, is now limited to contain no more than 50,000 elements. In previous version, trying to show larger lists could cause a crash.
- In the **License Configuration** dialog box, the supported AIMMS version of a license is no longer displayed because this information was only confusing and not really relevant.
- In the debugger, the tooltip showing the data of an identifier did not take the module or library prefix into account, and could thus show data of another identifier.
- In some developer dialogs, a decimal number was read back using the current settings for decimal point and thousand separator. In AIMMS, all developer dialogs should work according to the English numbering convention (as in the AIMMS language itself).
- The option `Warning_incorrect_file_format_for_CaseReadFromSingleFile` has been added in order to allow for testing whether a particular file is an actual case file by the function `CaseReadFromSingleFile`. Please see the Help for details about this option.
- When using multiple identical GUI expressions on a page, now only one instance of this expression is created and evaluated, leading to better performance.

*. . . Pivot table*

- In some situations, a **Pivot Table** that contains multiple identifiers with different dimensions and in which the user changed the order of these identifiers by drag-and-drop, a fatal application error could occur.
- A change in the current convention only changed the *unit* in the **Pivot Table**, not the actual values.
- In the **Pivot Table**, a right-mouse click in the grid area does not change the focus cell anymore.
- Aggregators were not updated when data displayed in a **Pivot Table** was changed from outside the **Pivot Table**.
- In some situations, a **Pivot Table** for which the sort order had been stored in the state file, could incorrectly be flagged as being changed.

*. . . Solving*

- In rare cases **XA 14** could return an inferior or even infeasible solution for a MIP model. This has been fixed in **XA 15**.
- **XA 14** could sometimes hang after an interrupt. This has been fixed in **XA 15**.
- Running the stochastic Benders algorithm with **XA 14** could result in a crash. This has been fixed in **XA 15**.
- There were some memory leaks in the functions `GMP::Instance::Copy` and `GMP::Instance::CreateDual`.
- AIMMS could hang if parallel **CPLEX** was used in combination with callbacks and inside that callback the function `PageRefreshAll` was called.
- For some models, the **Math Program Inspector** incorrectly reported a value of 0 for the density (on the **Matrix Statistics** tab).

- In some cases the bound analysis command from the **Math Program Inspector** could report incorrect results.
- When re-solving a nonlinear model with units, using the normal `solve` statement, an incorrect starting solution was passed to the solver for the second solve.
- The constraint listing file sometimes showed `Infeasible but not active` for infeasible rows, instead of ****.
- The constraint listing of a mathematical program, for which columns were protected by violation penalties, still marked these columns as frozen in some situations.
- The error message for not having specified at least one element in the objective column of a mathematical program, has been improved.
- In the special case that the model solved during a **CPLEX** MIP postsolve appears to be infeasible, the constraint level values are not available and they should have been set to 0.
- **BARON** now also checks whether **CONOPT 3.14G** is available.
- Interrupts from the interrupt dialog were not handled for asynchronous solves.
- Solving a model with **BARON** could result in a crash if one of the constraints was too large. AIMMS will now generate an error.
- For columns that were deleted by `GMP::Instance::CreatePresolved`, the following functions can now also be used:
  - `GMP::Column::GetLowerbound`, and
  - `GMP::Column::GetUpperbound`.
- `GMP::Instance::CreatePresolved` can now also be used for linear models.
- Performing sequences of actions in the **Math Program Inspector**, like Determine IIS, Determine Substructure causing Infeasibility, Resolve, etc., could yield unexpected results.
- The level values of variables frozen at level -1 were removed from the variables at a subsequent solve.

*. . . Databases*

- Expressions of data type element in the **Sql query** attribute of database procedures could lead to undefined results.
- Executing an AIMMS `SQLDirect` statement that includes a `DELETE` or an `UPDATE` SQL statement, erroneously returned the value `0` when the action did not affect any rows in the database (which is a perfectly valid situation).
- In rare cases, performing a `SELECT` **Sql query** on a database table or view with a number of long string columns in the column selection, yielded an out of memory error in AIMMS.
- When saving an Excel workbook with the function `ExcelSaveWorkbook`, using a relative path for the `SaveAsName` argument did not work properly.

*. . . Compiler and Execution*

- Aimms was not always able to load the DLL's for external functions properly, when dependent DLL's of Aimms changed the internal DLL search path. For example, this has been an issue with the combination of Aimms and Oracle 10.
- Aimms now detects the use of a domain set in the definition of a set as a union.
- Applying the function `NonDefault` to an index could cause a fatal application error.
- The compiler did not accept calls to functions with an index domain but without arguments.
- Getting an overview of the put files could lead to a fatal application error when Aimms was unable to obtain the filename for a particular file.
- On some installations of Windows XP, the localized names of time zones might be missing. Aimms was not designed to handle this eventuality.
- In the **Model Explorer**, if you select a single set that is a subset of `AllIdentifiers`, and `Ctrl-Shift-Drag` it within the **Model Explorer**, then you can now search for all identifiers contained in that subset instead of the set itself.
- A fatal application error could occur when running a procedure with indexed parameters as arguments and subsequently removing the **Subset of** attribute of a set that is used in the domain of the indexed parameter.
- Errors in data for identifiers not read in during a `read from file` statement are now issued as warnings and execution/data reading continues.
- Aimms used an expression optimization on expressions of the form: `index = ElemPar(otherIndex + 1)`, while it should only use that kind of optimization on expressions of the form: `index = ElemPar(otherIndex)`.
- The function `FindUsedElements` did not search identifiers whose domain was a true superset of the set searched.
- Running a just copied procedure from within the model editor did not always work.
- In the **Syntax Editor**, if the text was scrolled to the right, clicking the right mouse button somewhere in the text could accidentally show the popup menu that belongs to the yellow breakpoint area on the left.
- When executing a statement of the form `write sectionName to file...;`, Aimms used to evaluate the definitions of all sets and parameters instead of just the definitions of the sets and parameters in the named section.
- The performance of the execution could become unnecessarily slower, when having very sparse data and multiplying several identifiers with each other, which contain both unused and used running indices.
- Aimms could issue an unnecessary error message when the data type of a sub expression of an `if-then-else` expression had different but convertible data types.
- Setting a source file of a module to another `.amb` file whereby the contents of the new `.amb` file had less comment attributes, could result in superfluous error messages.
- Adding a **Library** with a root name or prefix name that clashed with that

of another library, could lead to a severe internal error.
- Expressions like `i in { a..b }` inside index domain conditions were not always properly handled.
- Closing and immediately re-opening a project, without closing AIMMS in between, could corrupt the internal data structures of AIMMS if the project contained a stochastic programming model. This caused a fatal application error.
- Repeated calls to `CaseCreateDifferenceFile` could lead to erroneous results for indexed sets.
- **From** and **To** attributes of **Arc** identifiers may now contain `if-then-else` expressions, provided that both the `then` and the `else` clause refer to a **Node** identifier.
- It is now ensured that the folder that stores the **Syntax Editor** configuration always exists.
- Comparing a set element which is the result from a function (like `last`, `first`) with an element parameter without a range, could cause AIMMS to issue a severe internal error.
- Importing a `.aim` file into a **Library** could add identifiers into the wrong namespace.
- During a section import, nested source files were not loaded.
- When the limit on integer variables is exceeded, this could confuse the memory manager of AIMMS, leading to unneccessary memory limit exceeded error messages and even to severe internal errors.
- The AIMMS compiler erroneously incorporated the index domain condition of variables into the **Sos weight** attribute of constraints.
- Displaying an error message that contains a %-sign sometimes resulted in a fatal application error.
- Applying the function `NonDefault` to an index could cause a fatal application error.

*. . . Units of Measurement*

- An unnecessary warning regarding different but commensurate units could arise under special circumstances.
- Calling a function, in which some of the arguments have units, could lead to incorrect unit consistency warnings.
- In the **Unit** tab of the properties dialog of a **Table** or **Composite Table**, you could get an incorrect error message: `Not a correct unit`.
- Values entered into a table were range checked according to the unit of the identifier instead of according to the unit used on the table.
- Using run-time unit execution on definitions could generate superfluous unit inconsistency warnings of the form: `The unit [-] is not consistent with the unit []`.
- Sparse execution of expressions involving individual unit analysis, nonzero defaults, and indexed element parameters as arguments of other identifiers, could lead to erroneous unit consistency errors.

*IT related ...*

- The installation has been updated to include the registration of the **Excel Add-In** components: `Folder.dll` and `MD5.dll`.

## AIMMS 3.8 build 4, release date 2008-09-16

*Build 3.8.4 2008-09-16*

In this AIMMS 3.8 Software Update, the following issues have been fixed or improved.

*IMPORTANT*

- **Particular error conditions on the internet license server could cause the automatic nodelock update process for personal nodelocks to end up with an empty, unusable, nodelock file resulting in a non-workable license. To minimize further interruptions, we higly recommend this Software Update.**

*Improvements*

- The life time of an emergency nodelock has been extended from 3 to 7 days, such that you have more time to arrange for someone to deactivate the license on the computer containing the active nodelock.
- **CPLEX** 11.1 has been added to AIMMS.
- The function `GMP::Instance::CalculateSubGradient` has been added. This function can be used to calculate advanced sensitivity information. It is only supported by **CPLEX** 9.1 and higher.
- When creating `aimmspack`-files, restricting access to a specific license/license range has been made easier.
- The `AimmsMAPI.dll` file, in the **E-mail and Sound example** has been updated to work with more versions of Microsoft Outlook.
- It is now possible to read from a **database table** identified by an element parameter. For example: `read from table elParDbTab;`

*Issues ...*

- AIMMS 3.8.3 added incorrect entries into the `.nch` file for copying procedures.
- Setting the visibility of an **Excel** workbook to `Off` (with the function `ExcelSetVisibility`) sometimes left the workbook still visible.
- The AIMMS 3.8 **Unicode COM Object example** erroneously had a reference to the Unicode COM Object of AIMMS 3.7.
- The error message file `log\aimms.err` is now properly closed upon project close.
- It is possible again to access dongles through a remote desktop connection.

*... GUI*

- The function `PageSetCursor` did not work correctly when applied to **Pivot Tables** to which artificial indices had been added.
- While running AIMMS on the right-side monitor in a dual monitor setup, dropdown lists of a **Selection** object could accidentally appear on the left-side monitor.

- In a **Scalar** object with a transparent background, it was not possible to see which value was selected.
- Sub-pages underneath a **Tabbed Page** object or an **Indexed Page** object did not respond correctly to data changes in their corresponding **Page Entry Procedure**.
- There is a new option `Use_subpage_Resize_Delay`. If you set this option to `On`, then it solves resizing problems on pages containing complicated sub pages (using **Indexed Page** and/or **Tabbed Page** objects).
- Changing the contents of a **Curve** object with an identifier on the X-axis, could result in errors (or a fatal application error) later on.
- In the **Gantt Chart**, when using a **Text** specification in which indices should be replaced by actual element names (for example: `"This is element i"`), the current library scope was not taken into account. If in this example the **Gantt Chart** was defined in the same library as the index `i`, you needed to prefix `i` with the library prefix.
- A ”Class not registered” message appeared upon trying to place an **ActiveX 3D-Chart** object on a page in the Unicode version of AIMMS.

*. . . Pivot Table*

- Aggregating in a **Pivot Table** over identifiers with a nonzero default value, didn’t always yield the expected results.
- Some changes have been made to the suffices that are initially shown on data pages of variables and constraints, that have not been saved before.
- When making changes using a multiple selection list box of a **Selection** object (as, for example, in the data page of a subset), for each individually selected item a data update was triggered, which could be inefficient when other objects (like a **Pivot Table**) were open as well.
- The developer state file of a **Pivot Table** in a **Tabbed Page** object was not saved when the containing page was saved.
- Making a subselection in a **Pivot Table** saved the changed status correctly into the state file. However, the page did not get a changed (*) mark.
- As a result of a change in AIMMS 3.8.3 to the sort order of elements in a **Pivot Table**, elements that were not present in a certain view of the **Pivot Table** but which were added as a result of some data change, were added to the *end* of the existing list of elements. This may not always be what you want, so an exception to this case has been introduced for the case in which the set itself had an ordering while no manual ordering (by means of drag-and-drop) had been specified. In this case, the elements will be ordered according to the set order.
- AIMMS sometimes displayed erroneous element names in the drop down list of a **Pivot Table**, when running a procedure from the menu that resulted in a change of data while such a drop down list was open.

*. . . Stochastic Programming*

- The **Benders Decomposition** algorithm for stochastic programming could fail if the model contained ranged constraints.
- In 64 bits Windows systems, an error in the mathematical program generator for Stochastic Programming could occur.
- In some situations, Stochastic Programming applications which save cases and have been edited, could lead to a severe internal error.
- There was an error in the generation of stochastic mathematical programs like:

```
VARIABLE
    identifier   : x
    index domain : (t,i,j) | xdom(i,j)
    property     : stochastic ;
```

  and

```
CONSTRAINT
    identifier   :  restr
    index domain :  ...
    definition   :   ... x(t,i,j-1) ...
```

  In the stochastic equivalent of this constraint AIMMS constructed the stochastic term:

```
    x.stochastic(sc,t,i,j-1)|xdom(i-1,j)
```

  While it should have constructed the term:

```
    x.stochastic(sc,t,i,j-1)|xdom(i,j-1)
```

*. . . Solving*

- The **NETSOL** help file has been added to AIMMS.
- A severe internal error could occur after first solving a mathematical program and then making two changes to the index domain of variables and/or constraints.
- AIMMS could become non-responsive, if **CPLEX Parallel Solve** was used with the option Show_branch_and_bound_tree switched on. AIMMS now ignores this option, because the MIP Search Tree in the **Math Program Inspector** is not supported for a parallel solve.
- The sparse execution of the function Max could get confused in case one of its arguments is the number 0.
- If the program status of a MIP model was IntegerInfeasible, the value stored in the suffix NumberOfInfeasibilities of the math program could be incorrect.
- The **Multi Start** module has been updated such that it can be used directly inside the **GMP Outer Approximation** module.
- If the function GMP::Column::Unfreeze was called for a GMP that was created using GMP::Instance::Copy, AIMMS could give a severe internal error.
- The calculation of variable ranges is now also affected by the options Shadow_price_range_relative_tolerance and Shadow_price_range_absolute_tolerance. AIMMS will give a warning if one of the problems solved during the calculation of sensitivity ranges is infeasible.

- When solving mathematical programs with violation penalties, AIMMS now removes excess variables, for which the corresponding entry in the violation penalty identifier has been set to 0.
- If the function GMP::SolverSession::Execute was called for two solver sessions created for the same GMP, or if GMP::SolverSession::Execute and GMP:Instance::Solve were both called for a solver session and its corresponding GMP respectively, AIMMS could pass incorrect updates to the solver.
- Callbacks are now called if the **CPLEX** option Parallel_mode is set to Deterministic, and the **CPLEX** option Global_thread_limit has value 1.
- The option Iteration_limit was not activated if **CPLEX** 11 used dynamic search to solve a MIP problem.

*. . . Math Program Inspector*

- AIMMS could give a fatal application error after selecting the action **Irreducible Inconsistent System** inside the Math Program Inspector for a math program with indicator constraints.
- For some models the Math Program Inspector incorrectly reported a value of 0 for the density (on the **Matrix statistics** tab).
- In some cases, the Math Program Inspector could lead to a fatal application error when used in the 64 bits version of AIMMS.

*. . . Databases*

- When trying to connect to a **system/user ODBC datasource** from AIMMS, while having a .dsn file with the same name in the project directory (e.g. datasource MySource and .dsn file MySource.dsn), a connection would be made to the datasource as specified in the .dsn file, instead of the intended one (i.e. the **system/user ODBC datasource**).
- In rare cases, a fatal application error could occur after executing a too complex database query from within AIMMS.
- Sometimes, when trying to connect AIMMS to a **MySQL ODBC datasource**, which did not contain a ”Default schema” reference in its properties, a fatal application error occurred.

*. . . Model Explorer*

- Copying a declaration section from one procedure to another could corrupt the information in the original declaration section.
- A severe internal error could occur after changing a **Calendar** into a **Set** and back.
- When moving an identifier in the model tree, while an **Attribute window** for that identifier was open, the **Attribute window** was not referring correctly to the new location in the tree. A similar thing happened when the identifier was included in a **View window**.
- Subsequent changes to the attributes of an **assertion**, of which the number of indices had been changed, could lead to a severe internal error.
- The syntax editor erroneously recognized a stand-alone suffix inside definitions of variables, constraints and parameters, while suffices should always be preceded by their appropriate identifier.

- In some cases, AIMMS did not respond anymore when you entered faulty `option` or `property` statements in the syntax editor.
- When a function or a procedure had zero arguments, using **argument completion** resulted in unreadable characters to be inserted at the cursor position.
- Sometimes, in the mapping attribute of a **database table**, a defined scalar element parameter as an argument of an identifier was not evaluated in time.
- When moving a procedure to another location in the model tree, AIMMS always converted tab characters into 8 spaces.

*. . . Compiler and Execution*

- Erroneous results could occur when two different cases are being executed sequentially (with the second case containing less data than the first) and using the operator `First` inside a `FOR` loop to assign a value to an identifier without additional free indices.
- Creating a `.aim` file via the source file wizard (by choosing **Write...** and selecting type `.aim`), could corrupt the accompanying `.amb` file.
- If the option `Maximal_number_of_warnings_reported` has been set to 0 programmatically (with the intrinsic function `OptionSetValue` or an `option` statement), AIMMS will now hide the **Errors/Warnings window** when there are compile time warnings generated.
- Using an `IN indexedSet` clause inside the index list in an **Index domain** attribute, was not always processed correctly by the compiler. This could lead to erroneous results.
- The `IN` clause used in iterative expressions such as `sum(i in SomeSubset, ...)` could affect the results of subsequent statements in which that same index was also used.
- The value `0` for the option `Put_Number_Width` was not properly handled.
- The suffix `.DefinitionViolation` on a variable without definition, could cause AIMMS to generate a fatal application error. Now, a proper compilation error message is generated.
- **Macro** identifiers now properly handle expressions of the form `X(ep(i)-1)` (where `ep` is an element parameter).
- AIMMS now verifies that the parameter `ep` in the expression `P(ep-1,j)` indeed has dimension 0.
- The character `ç` (= ASCII character 231) is now accepted as a currency symbol and as (part of) a unit name.
- AIMMS now issues a proper error message when the `..` operator is used in the data defined over compound sets.
- The statement `Empty P(i);`, where `i` is an index into an empty set, inadvertently emptied the entire parameter P.

*. . . Units of Meassurement*

- The computation of runtime computed units in **if-then-else** expressions involving variables with the inline property, did not always work properly.
- Indices without a scope in the **unit** attribute of parameters or in a **convention** attribute now result in a compiler error message.
- Expressions that force a runtime unit evaluation via unit casts, were not properly handled by AIMMS **macro's**, which could result in superfluous runtime error messages for inline variables.
- The sparse execution of numeric casts via **unit parameters** could be confused with conversions containing non-zero additions during the conversion. For example, with base unit K (Kelvin) and derived unit C (Celsius) and a unit parameter UP with value [C], the cast 0[UP] was computed incorrectly.
- A **calendar** containing a non-singular time unit such as 7 * days or 15 * minute for the **unit** attribute, which becomes empty at run time because the end date is set before the start date, could cause a severe internal error.
- AIMMS used to support only the units tick, second, minute, hour, day, month, year and century in the function StringToMoment. Now AIMMS supports all units that are commensurate with these units.

*. . . Cases*

- In the **Data Manager**, the **Case Properties** and the **Dataset Properties** dialog boxes did not show the modification date.
- The optional argument respectDomainCurrentCase has been added to the function CaseCreateDifferenceFile.
- The option Case_save_inactive_data has been added to the section AIMMS - Case management. The range of this option is {off, on}. The default for AIMMS 3.8 is on. When this option is set to on, inactive data is stored in cases.
- The intrinsic function CaseCompareIdentifier did not properly compare the contents of indexed sets.
- The function CaseReadFromSingleFile now returns 0 when the case file does not adhere to the AIMMS case format. CurrentErrorMessage is also filled.

## AIMMS 3.8 build 3, release date 2008-05-15

*Build 3.8.3 2008-05-15*

In this AIMMS 3.8 Software Update, the following issues have been fixed or improved. Please note that almost all issues that have been fixed in the AIMMS 3.7.109 Software Update, have also been fixed in this AIMMS 3.8 Software Update.

*Improvements*

- Licensing support for Sentinel dongles has been added to Windows 64 bits (x64) AIMMS versions.
- When updating a license in the **License Configuration** dialog box, you now get more detailed information on what has actually been updated.
- There are new versions of **KNITRO** and **SNOPT**, i.e., **KNITRO 5.2** and **SNOPT 7.2**.
- **CONOPT 3.14G** has been added to the typical installation of AIMMS.
- The **XPRESS 18** link has been upgraded from 18.00 to 18.10.
- The following new GMP functions have been introduced:
  - `GMP::Tuning::TuneSingleGMP`,
  - `GMP::Tuning::TuneMultipleMPS` and
  - `GMP::Tuning::SolveSingleMPS`.

  The first two functions can be used to activate the tuning tool of **CPLEX 11**. The last function can be used to solve an MPS file.
- There is a new component in the XML mapping tree: **Virtual Attribute**. See chapter ’Reading and Writing XML Data’ in the AIMMS Language Reference for more information.
- Some default formatting properties for **Pivot Table** identifiers can now also be specified through identifiers (in addition to a fixed keyword).

*Issues . . .*

- In case of a startup error, AIMMS Web Services does not show a *Could not startup* AIMMS dialog box anymore.
- The **Multi Agent Setup** dialog box created Send-, Handler-, and Filter-procedures for agent roles that were not marked as ’to be implemented’ in the current project.
- A better error message is given when the **AIMMS Launch Utility** is unable to open the specified file.
- A fatal application error could occur at startup of the AIMMS ASCII version for users in Eastern Europe or Asia, due to a coding error in the matching of the local time zone name with the set `AllTimeZones`.

*. . . GUI*

- Using **File-Exit** in a popup menu on a page could result in a fatal application error.
- In the **Network** object, small nodes were not aligned correctly with the attached arcs.
- Using long GUI expressions could result in a fatal application error. If an expression is longer than approximately 256 characters, then the expression is automatically broken up in multiple lines. This could lead to an error when reopening the containing page.
- When a set AxB (subset of (A,B)) is displayed in the GUI, AIMMS displayed the *root sets* of A and B, instead of A and B themselves.
- The tooltip for options was incorrect.
- A newly created **Table** with an identifier that uses compound indices was not correctly displayed.
- An execution error during the initialization of a **Scalar** object could result

in a fatal application error.

- In some situations, disabling a toolbar item through an AIMMS identifier was not always reflected immediately.
- Copying a template or page as a child of a **Print Template** that uses a different paper size, could lead to strangely sized pages.
- The functions `PageGetNext`, `PageGetPrevious`, `PageGetChild`, and `PageGetParent` did not check correctly whether a page was hidden or not. These functions now have an optional argument: `IncludeHiddenPages`. See the AIMMS Function Reference for details.
- Since the **Data Manager** can have its own user specified menu, some menu commands seem to appear multiple times in the actions of a button/menu item. Now a number of these commands are prefixed with "Data Manager" to make the distinction more clear.
- In some cases, in a **2D Chart**, changing the X-axis domain from a string parameter to a numerical parameter and back, made the chart disappear.
- In the **ActiveX** objects, using a font without the style bold, italic or underline, did not work correctly.
- Upon opening, an **Indexed Page** object did not update the underlying set(s) that determine the number of rows and columns to show.
- On **Windows Vista**, **Scalar** objects on dialog pages were sometimes not displayed correctly.

*. . . Pivot table*

- The **Pivot Table** did not show compound sets correctly, nor identifiers defined over a compound set.
- In some situations a **Pivot Table** for which the identifier index was located in the outer index area could crash when the data was refreshed.
- Some sorting issues in the **Pivot Table** have been resolved.
- Changing the existing element order of a set was not updated correctly to the **Pivot Table**.
- The alignment of parametric units in the **Pivot Table** has been improved.
- Making changes in a **Pivot Table** that contained aggregators did not trigger a recalculation of the aggregated values for aggregators that were not specified on the leaf index in the row or column tree.
- **Pivot Tables** on a page in a read-only library that store the developer state during page save, will not set the 'page changed' asterisk anymore.
- When multiple threads were accessing the model (e.g. when using Multi Agent technology), the **Pivot Table** could cause a fatal application error.

*. . . Cases*

- Newly declared subsets could retain inactive elements after a case load.
- A severe internal error could occur after loading a case with an empty indexed set and executing over that indexed set.
- A fatal application error could occur during the load of a corrupt case.
- Errors occuring during a case load were not always displayed in the **Messages/Errors** window.

*. . . Libraries*

- If a project contains references to libraries higher in the folder hierarchy (using a path name starting with ..\ ), then the **Project Export** to a destination folder did not work.
- Superfluous error messages inside a license protected library are now suppressed.
- A fatal application error could occur when switching from AIMMS ASCII to AIMMS Unicode for a project with libraries.
- Code that refers to members of modules inside libraries was indicated to be invalid by the syntax-editor.

*. . . Model Explorer*

- The speed with which sections can be moved using the model editor from one namespace to another has been improved.
- Because the meaning of an AIMMS application does not change by changing the order of declarations/nodes, identifier moves are no longer reported during the source file import check, unless the backward compatibility option `Notify_all_changes_during_amb_import_check` has been set to `On`. In addition, because attribute changes rarely influence the application significantly, they are also no longer reported unless the above mentioned option has been set.
- When changing a binary parameter into a string parameter, handling the range attribute is now done correctly.
- A message has been added to inform the project developer that changes made using the source file wizard are immediately saved to the model.
- Emptying the unit attribute of a function was in some cases not accepted by the model editor.
- It is now ensured that when the result of an external function is non-scalar, the implementation must pass the result as an argument.
- Auto-generated symbols like `MyVariable_definition`, could not be located in the Model Explorer when pressing the **Declaration** button in the **Find** dialog box.
- An incorrect error message about range mismatches could occur, involving multi-dimensional sets and indexed element parameters used in the condition of that set.
- AIMMS does not update defined parameters and sets anymore before a data backup.
- When a model edit in the **Order by** attribute of a set is followed by F5 (recompile model), this new ordering is now reflected on pages where this set is displayed.
- In some situations, a severe internal error could occur when loading a `.aim` file.
- In some situations the syntax editor could not be loaded correctly.

*. . . XML*

- When writing an XML file, a **Maps to** attribute in conjunction with a **Binds to** attribute is no longer ignored.
- AIMMS did not correctly read XML files containing an **ElementMapping** with a **Maps to** attribute, referencing an indexed set, that was used in conjunction with a **Binds to** attribute.

*. . . Name changes*

- The name change module did not ignore the text between `data`{ and } in some cases.
- Name change entries dated in the future are now discarded.
- The name change of a complementarity variable of which the **Complement** attribute referenced a constraint declared elsewhere also renamed that constraint.
- Name changes of defined variables could result in the constraints that represented the corresponding equalities to become invisible for pages and the **Math Program Inspector**.
- If a page contains an identifier reference in which one of the indices is fixed on the 'empty element', then **Commit Name Changes** failed for that page.
- For *copying* a node from one namespace to another namespace, the name change module is now used in a similar way as when *moving* a node from one namespace to another.

*. . . Solving*

- When the solver did not accept restarts AIMMS would completely regenerate a matrix even when partial regeneration would suffice.
- AIMMS gave a severe internal error if the **AOA** procedure `MasterMIPSetCallBack` was called outside the Outer Approximation module.
- The function `GMP::Instance::Copy` no longer copies the solver that was selected (if any) by the function `GMP::Instance::SetSolver`. Otherwise unexpected side effects could occur.
- The GMP Outer Approximation Module now deletes the GMP's for the MIP and NLP sub problems at the end of the algorithm.
- A memory leak in `GMP::Instance::Solve` and `GMP::SolverSession::Execute` for quadratic models has been corrected.
- Under rare circumstances regenerating a quadratic mathematical program could crash.
- The value assigned to the `.iterations` suffix of a Math Program could be incorrect for a MIP model in case **CPLEX** was used to solve the model.
- Using the functions `GMP::Instance::Delete` or `GMP::Instance::DeleteSolverSession` could result in a severe internal error.
- The performance of regenerating a mathematical program when many rows of a constraint are discarded (because of zero elements in a generated row or because the generated row is of the form `<= inf`), has been improved.
- Using the proxy stub interface for a nonlinear solver could fail.
- The Best LP bound shown in the progress window could sometimes in-

correctly increase (decrease) for a maximization (minimization) problem, if the **CPLEX 11** dynamic search algorithm was used.

- The initialization of **XPRESS 18** could fail.
- Arcs were not properly initialized for generation after a recompile.
- AIMMS now suppresses an unnecessary warning on multiple SOS Weights for a single column when one of the weights is 0.0.
- A variable will no longer be included in the SOS set if the weight of that variable specified in the **Sos weight** attribute equals 0.
- The upper bound on the option `Equality_relative_tolerance` has been reduced from 1.0 to 0.9.
- AIMMS would take unnecessary long to generate a constraint with definition of the form: `if ... then ... else 0 <= 0 endif`.
- The function `Sign` is now allowed in arbitrary non-linear expressions involving variables in constraints.
- AIMMS now stops execution when the option `Warning_non_optimal` is set to `error` and there is an infeasible model.
- The evaluation of inline variables could be incorrect in some cases where the `IN` operator was used.
- A severe internal error could occur when generating variables that are only referenced in activating conditions of constraints.

*. . . Math Program Inspector*

- The objective value displayed in the **Math Program Inspector** could be incorrect after a resolve.
- When using **CPLEX 11**, the branch & bound tree was not displayed correctly in the **Math Program Inspector**.
- In some situations the **Math Program Inspector** did not report dual degeneracy correctly.

*. . . Stochastic Programming*

- A superfluous compilation error could occur, regarding stochastic parameters for which in the definition a distribution was used.
- Commenting out definitions in stochastic parameters is now also reflected in commenting out the definition of the corresponding `.Stochastic` parameter.
- Non-binding stochastic rows were not properly emptied which could result in erroneous stochastic programs being generated.
- An error in the model editor could occur when adding the **Stochastic** property to variables. This might cause a fatal application error in large applications.
- A fatal application error could occur when both stochastic programming and violation penalties were used.
- A superfluous error message could occur regarding integer columns issued by the Stochastic Benders algorithm solving an RMIP problem.
- When a parameter contains stochastic data but the stochastic property of that parameter has been cleared, an error message could occur when saving a case.

*. . . Compiler and Execution*

- The performance has been improved in case the value of a summation was empty whilst the condition on the summation could require a significant amount of time.
- The performance has been improved for the function `Max`, used over very sparse data.
- The options that can be both assigned a keyword and a corresponding integer code are now also accepted by `OptionGetValue`.
- The Excel functions did not give proper error messages when trying to read a non-ASCII string from a spreadsheet.
- In some situations, repeated compilation could drastically increase the amount of memory used.
- The AIMMS debugger does no longer try to jump into external procedures, database procedures and intrinsic procedures (e.g. `Sin`, `Cos`). Of course, user procedures are still stepped in.
- Index domain conditions consisting of just a set reference, e.g. `(i | someSet)`, are now considered as a test on whether that set is non-empty.
- Multiple (two or more) unreferenced indices in the index domain of a parameter were not always handled properly. Consider e.g. the set `S` with indices `i,j,k,l` and `m`, and `IS(i)` an indexed subset of `S`. Then for parameter `P(i,j,k,l,m)` with definition `1 $ ( j in IS(i) )`, the data was not evaluated correctly, because the definition of `P` has unreferenced indices `k,l` and `m`.
- In AIMMS 3.8.2 the $ operator started to use the sparsity of the left hand side operand, but this could lead to an unnecessary evaluation of the left hand side.
- The slicing of the result of vector or matrix valued functions is not supported. The compiler now ensures that this feature is not inadvertently used.
- Calling an external function in a DLL that crashed, because the function or DLL itself was corrupt, resulted in a fatal application error in AIMMS.
- A severe internal error could occur when cleaning a slice of a parameter.
- If a definition used a call to another procedure or function, then when the GUI requested for an update of the definition, the GUI temporarily regained control which may lead to all kinds of problems (most likely: recursion).
- A set assignment involving lags and/or leads and copying an element of an indexed set (which is also a set) to another set, was not done correctly.
- The name change module did not work properly in some cases when an identifier was declared both in the global scope and in a module.
- A fatal application error could occur during compilation, due to a missing mandatory range attribute of an element parameter.
- After compilation, AIMMS initializes by assigning the data of the initial data attribute and executing the `MainInitialization` and `LibraryInitialization` procedures. At the end of this initialization, the data consistency of sets and parameter is verified. In order to avoid confusing error messages from this verification step that are simply a consequence of statements

not executed by the `MainInitialization`/`LibraryInitialization` procedures, AIMMS now skips this latter verification step if there are execution errors halting the execution of one of these procedures.

- AIMMS now requires all identifiers in a binding domain of a `for` loop to be indices. As this may give errors, an option has been introduced in the backward compatibility category: `Require_binding_domain_for_to_be_indices_only` with range { `On, Off` }. For AIMMS 3.8 the default is `Off`. For AIMMS 3.9 the default will be `On`.
- The option `Warning_numerical_range_violation` has been renamed to `Warning_range_violation`.
- The range checking on output arguments of AIMMS procedures has been improved.

*... Viewer*

- Menu items in the **Data** menu were not enabled/disabled correctly when switching from or to **AIMMS Viewer Test Mode**.
- If the currently active case is in a folder of the **Data Manager**, then its name was not correctly filled in in the **Export Viewer Project** dialog box.
- When running a project in which the VAR license attribute (of some module) had been specified by means of a folder in the AIMMS Viewer, the AIMMS Viewer was not able to find the corresponding VAR license (i.e. a VAR license file with name 255255003008.var in the specified folder).
- When using the AIMMS Viewer with the 95-style menus (standard in the 64 bits version of AIMMS), the menu contained many irrelevant commands.

## AIMMS 3.8 build 2, release date 2008-02-20

*Build 3.8.2 2008-02-20*

In this AIMMS 3.8 Software Update, the following issues have been fixed or improved.

*Improvements*

- The assignment operators `--=` and `++=` have been added to the language.

*Issues ...*

- Writing to a **Database table**, which contained a non-required numeric column, could lead to the writing of the value `0` instead of `NULL` in this column, if the mapped AIMMS identifier was an element in a subset of `Integers` and the value `''` was passed.
- It is now possible to map an XML tag that has both a required attribute and text content.
- The Linux version of AIMMS now supports `.aimmspack` files. The file `/home/name/MyProject.aimmspack` will be unpacked to the (newly created) folder `/home/name/MyProject/`
- Crash dump files can now really be sent to Paragon using the dump file dialog (which appears when AIMMS issues a fatal application error).
- The **Fatal Application Error** dialog box appeared unjustly when AIMMS could not allocate enough memory.

- The text editor behind the **File - Open - Text File...** menu item now displays scrollbars when needed.
- Added **Used Identifiers...** menu command to right mouse menu of the **Page Manager** and the **Template Manager**.
- The menu command **Find** did not work in the **Data Manager** and the **Data Management Setup**.
- In rare circumstances the AIMMS debugger could produce a strange page containing just the word 'Illegal'.
- In the AIMMS 3.8 Maintenance Release, the **Web Service Caller** example did not include the files to actually call the web service.
- The unicode version of AIMMS showed the error "ClassFactory cannot supply requested class" when trying to place an ActiveX **chart** object on a page.

*...Model explorer*

- Moving variables and constraints to a declaration section that is local to a procedure, is not allowed anymore. Is is however still possible for **external functions**.
- A fatal application error could occur when removing the **Unit** attribute of a local variable of an **external function**.
- Variables, which are declared as arguments of external functions, are not allowed to have a definition anymore.
- The model explorer now properly handles the **Convention** attribute of **database procedures**.
- A warning during compilation could result in an incorrect error message during a **Project Export** action.
- Model explorer actions, in particular saving `.amb` files, interchanged with the execution of procedures, are now less sensitive to an accidental change of the current working folder.
- Performing autocompletion inside the **Comment** attribute of a namespace prefixed identifier, could cause AIMMS to crash.
- AIMMS now checks whether properties, which are set in the **Property** attribute of identifiers, are consistent with each other, and issues a warning when they are not.
- Name changes used to be applied to cases with respect to the date when these cases had been created. This, however, is not sufficient when cases from older instances of a project are copied to newer instances of that project. Now, not only the creation date of cases is stored inside the case, but also the date of the last name change of the project in which the case is stored. This date will now be used for the application of name changes to cases.
- When using the **Declaration** button on the **Find** dialog box to jump to the location of an identifier in the **Model Explorer**, the **Model Explorer** menus and toolbar did not always have the right read-only status to reflect the currently selected identifier.
- The unused node names `gui`, `trash` and `configuration` are now accessible

from the model tree.
- For a variable X with a specified **Definition** attribute, AIMMS automatically creates a constraint X_definition. When the developer created both these identifiers himself, AIMMS still allowed the user to specify the **Definition** attribute of the variable, which could lead to conflicts. Therefore, the **Definition** attribute is removed in these cases. In order to avoid future confusion, AIMMS now forbids the declaration of X_Definition when X is already declared.
- The attribute windows have been changed, such that they correctly display large attribute names, such as **Timeslot format**.
- Changes made to procedures in the **Model Explorer** were not immediately applied to attribute windows that were still open.
- Name completion in the **Subset of** wizard now takes the scope in which the set has been declared (i.e. a library) into account.

*. . . Libraries*

- When a **Library** was missing during project startup, the **Library Manager** produced incorrect error messages when trying to specify the correct location of it.
- The **Library Manager** did not check whether two libraries referred to the same model file.
- AIMMS now allows a string parameter that is used in the **User data** attribute of modules or libraries to be declared within modules or libraries.
- Sometimes, private module/library members showed up in the public autocompletion list, when having typed the module/library prefix and showing this autocompletion list.

*. . . GUI*

- In the **Network** object, deleting the last source or layer specifying a GIS background, did not delete all underlying properties.
- In developer mode, the **Selection** object will not create more than 2000 individual radio buttons or check boxes. Because of the set constructions in AIMMS you could easily create a larger number of radio buttons or check boxes, but that would let Windows run out of resources.
- The width of the drop down list of a **Selection** object was not always correct.
- Several dual monitor position issues have been fixed with respect to the position of dialogs and the drop down list in, for example, the **Pivot Table**.
- The **Used Identifiers** dialog box is now initially sorted on the name of the identifiers.
- The range of special values (e.g. inf) was not properly checked when updating values from within a page object.
- Including a copy of the menu command **Tools - Diagnostic Tools - Debugger** or **Tools - Diagnostic Tools - Profiler** in a user menu, causes that this copy of the command did not work after reopening the project. Now, new copies do work, so to make these menu items work in the user

menu, you need to delete the old item and create a new copy.

- The last vertical grid line in a **Table** object was not always drawn.
- In a **Curve** Object, the **Upon Selection** procedure was not always invoked when clicking on points in the curve. Also, the value in the status line was displayed using the background color and was thus invisible.
- When deleting the only identifier in the **Contents** tab of a **Scalar** object, it was not always allowed to add a new identifier immediately.
- In properties dialog boxes, when using the **Wizard** button to select an identifier after already having typed (part of) an identifier name in an edit field: if the content of this edit field was semantically incorrect, this content was not shown at all in the wizard.
- When a GUI expression was specified in one of the properties of a page object, it was not always possible to change that expression back to a normal identifier reference, using the **Identifier Selection** dialog box.
- The menu command **Tools - Commit Name Changes...** will now remove any exclamation marks on pages in the **Page Manager**, if the corresponding page was opened and re-saved correctly during the process.
- On the **Text** tab of the properties dialog box, literal strings were sometimes rejected for no valid reason.
- In the **Network** object properties dialog box, several properties of the arcs and nodes did not allow expressions to be entered.
- In the **Network** object properties dialog box, specifying identifiers with definitions did not always lead to the immediate evaluation of these definitions.
- When using segmented arcs in a **Network** object, arrow heads were drawn incorrectly when their position value was greater than 100.
- In a **2D-chart** object, the **Y(1)-axis annotation** values were not displayed in rare cases.
- The **Upon Double-Click** procedure of the **Selection** object did not update the data on pages afterwards.

*. . . Pivot table*

- In the **Pivot Table**, it was not always easy to move outer indices to the row or column tree area when the current grid area displayed a single scalar.
- Setting the focus in the **Pivot Table** through the function `PageSetCursor` sometimes resulted in a new selection larger than the new focus cell.
- Using a GUI expression to specify an artificial index relation in the **Pivot Table** did not work.
- Reading back a state file for a **Pivot Table** which had been sorted on an aggregator row or column could lead to a fatal application error.
- Data changes that result in rows or columns being added to a **Pivot Table** could sometimes result in a strange ordering within the **Pivot Table**.
- Adding an identifier with a suffix to a **Pivot Table**, while also having specified some explicit suffices for the suffix index, would lead to a fatal application error. For identifiers with a suffix, the explicit suffices as

specified for the suffix index will now be ignored.

- When editing an element-valued cell value in a **Pivot Table**, the drop-down list could sometimes overlap with a scrollbar.
- Setting a reverse link in a **Pivot Table**, in which some of the domain sets were empty, did not work.
- Reverse links defined in a **Pivot Table** were not always updated correctly when initially opening the **Pivot Table**.
- Some issues with sorting in the **Pivot Table** have been solved.
- **Pivot Table** objects on **Tabbed Page** objects were not updated correctly if the identifier for the column width was changed.
- When data was changed in a **Pivot Table**, in which elements had been sorted alphabetically, it could happen that the updated layout did not follow the specified sort criterion.

*. . . Solving*

- AIMMS now issues a normal error instead of a severe internal error, if **BARON** is used as NLP solver by **AOA**.
- The **CPLEX** status file was written to the project folder while it should have been written to the `log` folder.
- AIMMS has been extended with the functions `GMP::Row::GetScale` and `GMP::Col::GetScale`. See the Function Reference for details.
- When using AIMMS with several users on the same machine, the solver configuration files could become inaccessible for some users.
- A severe internal error could result from combining violation penalties with model explorer actions and sensitivity analysis.
- AIMMS has been extended with the new option `Display_solution_statistics` for **CPLEX 10.1** and **CPLEX 11.0**. This option can be used to display statistics that give information about the quality of the solution. See the **CPLEX** help file for more information.
- When, in the first solve, a constraint generates unbinding or empty rows, a structural data change (the creation or deletion of a data element) to be incorporated in the third or subsequent solve could be ignored resulting in too few or too many rows generated.
- Multiple MPS files were generated even if the option for writing MPS files (option `Mps`) was set to `At_the_first_solve`.
- During generation, AIMMS moves the level value of frozen columns inside the bounds of that column, including the bounds of frozen columns, because AIMMS guarantees to the solver that the level value is inside the bounds (with the exception of semi-continuous variables). This was done silently when the difference between the level value and the nearest bound was less than the value of the option `Bound_tolerance`. A warning was issued when the difference was more than the value of this option. The value is now only moved when a corresponding warning message is issued. See also the options `Warning_move_frozen_into_bounds` and `Bound_tolerance`.
- `GMP::Solution::IsDualDegenerated` and `GMP::Solution::IsPrimalDegenerated`

can now also be used for QP models.

- A severe internal error could occur when AIMMS had to create a derivative evaluation error message involving special numbers and the power operator.
- The combination of stochastic programming and violation penalties has been improved.

*. . . Compiler and Execution*

- Changing the value of the option Equality_absolute_tolerance could lead to incorrect output when writing identifiers to files.
- The function SetElementRename does not rename elements from the predeclared identifier AllIdentifiers, or one of its subsets, anymore.
- The syntax checks on unit names have become more strict.
- The pre-declared index in the set CurrentAutoUpdatedDefinitions has been renamed from IndexCurrentAutoUpdateDefinitions to IndexCurrentAutoUpdatedDefinitions.
- A fatal application error could occur when combining the operators or and in with two set expressions, as, for example, in the expression

```
i in ( SetA or SetB )
```

- Index domain conditions are now handled properly, when one of the indices is an index into a **horizon**.
- A compiler check has been added which checks that a procedure does not call itself from within an expression in its body.
- Now the compiler checks that procedures with non-scalar output arguments are not called with scalar arguments.
- The $ operator now uses the sparsity of the left hand side operand when that operand is 'safe' (expression without side effects, little or no chance of errors during evaluation).
- Indexed sets with inactive data read from a case and written to a text file, could cause a syntax error when read back again from that text file.
- In some circumstances NBest could consider elements which did not satisfy the condition on its iterating index.
- The index domain condition of inline variables is now used during the substitution into constraints.

*. . . Units of Measurement*

- The function ElementRange now converts its first three arguments to the unit of that argument. For example:

```
QUANTITY:
   identifier   :  SI_Time_Duration
   base unit    :  s
   conversions  :  minute -> s : # -> # * 60
   comment      :  "Expresses the value for the duration of periods." ;

PARAMETER:
   identifier   :  pNumPeriods
   unit         :  minute
   initial data :  3 ;
```

```
SET:
   identifier   :  somePeriods
   index        :  iPeriod
   definition   :  ElementRange(1, pNumPeriods) ;
```

now generates 3 elements (based on minutes) instead of 180 (based on seconds).

## What is new in AIMMS 3.8

*What is new in AIMMS 3.8?*

This documentation reflects the state of AIMMS version 3.8. Compared to AIMMS 3.7, the following major new and extended functionalities have been added to the system:

*Benders decomposition for stochastic programming*

Following the introduction in AIMMS 3.7 of language features to formulate and generate the deterministic equivalent of multi-stage stochastic recourse models, AIMMS 3.8 includes an algorithm to solve stochastic models using a stochastic Benders approach. The nested Benders decomposition algorithm is a popular algorithm for solving (linear) multi-stage stochastic programming problems. AIMMS offers nested Benders decomposition as a user-customizable procedure inside the AIMMS library for operations on Generated Mathematical Programs (GMP library), similar to the AIMMS Outer Approximation solver.

*Benefits*

The Benders decomposition algorithm can be faster for stochastic models than solving the deterministic equivalent with an LP solver. The open implementation of the algorithm allows modelers to fine-tune it to their specific stochastic model and to use advanced features of the GMP library such as parallel/distributed solver sessions to solve multiple sub-problems in parallel, which may drastically reduce the total solution time.

*Nonlinear presolve*

AIMMS 3.8 offers a powerful presolve procedure for Nonlinear Programs and Mixed-Integer Nonlinear Programs, analogous to the presolve procedures employed by LP/MIP solvers. The AIMMS nonlinear presolve procedure applies techniques like:

- Inversion of nonlinear expressions.
- Solving the variables in the pre-triangular part of the matrix.
- Tightening the bounds of variables based on the linear and nonlinear constraints.

*Benefits*

The presolve procedure may simplify the model sent to a nonlinear solver (CONOPT, SNOPT, KNITRO, LGO, BARON, MINOS, or the AIMMS Outer Approximation solver). With the presolve, nonlinear problems may be solved faster and the solution may be better. The solver may even find a feasible solution for problems declared infeasible without the presolve, and in other cases provide proof that it really is infeasible.

*Multi-start solve*

AIMMS 3.8 includes a multi-start algorithm for nonlinear programming problems. The multi-start algorithm calls an NLP solver from multiple starting points and reports back the best feasible solution as its final solution. You can also obtain all feasible solutions found. The AIMMS multi-start algorithm has been written as a user-customizable procedure within the GMP library, allowing modelers to adapt the algorithm to their own needs.

*Benefits*

The multi-start algorithm may increase the chance of finding a good final solution. Multi-start increases the total solving time, but this may be limited by using the advanced features of the GMP library such as parallel/distributed solver sessions to solve problems with different starting points in parallel.

*Network object extensions*

The AIMMS graphical object to display networks has been extended to include:

- curved arcs,
- segmented arcs with either 2 or 3 segments, and
- more intuitive navigating and zooming (see picture).

*GIS extensions*

The link to Geographic Information Systems (GIS) has been extended to support:

- More map formats, such as:
  - ESRI shape files.
  - Yahoo! Maps.
  - File image based sources.
- Indexed sources and layers to allow data-driven specification of multiple sources and layers in a map (see picture).
- Geocoding language support (to find latitude/longitude coordinates).

*Pivot table extensions*

The versatility of the AIMMS pivot table has been further enhanced with:

- Sorting indices in the row and column areas.
- Nested sorting in the grid area.
- Column widths controllable through a model parameter.

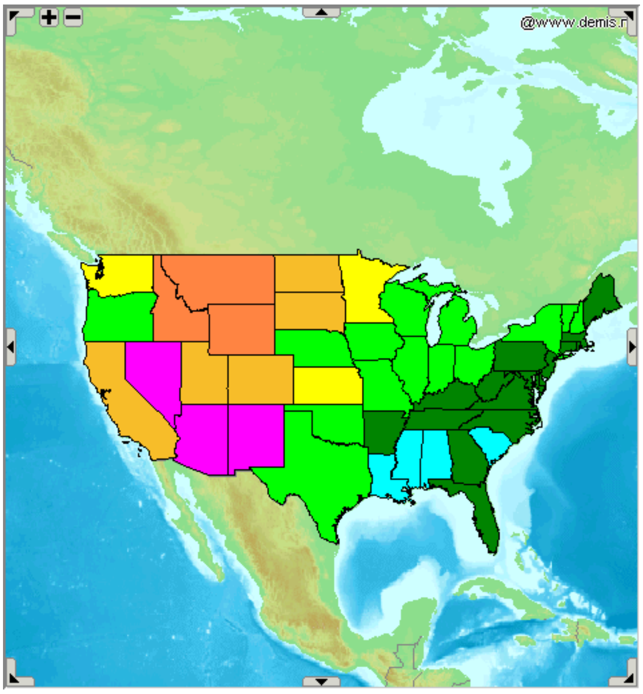


Figure 4.1: Example of a network flow object with a GIS-generated background

*Webservices-based RPC framework*

AIMMS 3.8 contains a portable, web services-based, RPC framework to replace a number of Windows-only COM-based RPC functionalities used by AIMMS 3.7 and before. The following services are now based on this new framework:

- AIMMS agent technology,
- AIMMS web services,
- proxy-stub solver interface, and
- a limited replacement of the current AIMMS COM object.

The RPC framework comes in a separate installation file.

*Benefits*

The services listed above, which were previously only available on Windows, will, in the Maintenance Release, also be available on Linux, and be truly interoperable between these platforms. For instance, a web service handler running on a Windows server can then dispatch requests to AIMMS agents on a Linux server. Also, the new framework allows solver sessions to run on a remote computer. This may provide a convenient way to solve stochastic Benders problems in a distributed manner, on an existing hardware infrastructure.

*`.aimmspack` files*

AIMMS now offers the possibility to pack all files related to an AIMMS project into one single (compressed) `.aimmspack` file. This functionality is not limited to AIMMS specific files (like `.amb`, `.prj`, etc.), but to all files and subdirectories that are needed for the project to work properly.

*Benefits*

The packing functionality makes it much more straightforward to distribute AIMMS projects with all related files included.

AIMMS *Viewer*

At the same time of the release of AIMMS 3.8, we will also release a freely distributable AIMMS Viewer, which allows anyone to browse the results of an AIMMS-based optimization application through the graphical user interface of the application. The Viewer does not allow entering or modifying data or performing optimizations. Developers using AIMMS 3.8 can opt to make their applications viewable for the AIMMS Viewer, while confidential data in a viewable project can be protected by means of a password protection scheme or through the common VAR licensing scheme of AIMMS.

*Benefits*

Combined with a newly introduced single project file, the viewer provides a convenient way to distribute a single file containing a complete interactive report, ready for anyone to view. The AIMMS Viewer thus enables you to easily share the results of your optimization model with colleagues and customers, to create sophisticated management dashboards, and to demonstrate your AIMMS applications to prospects.

## AIMMS 3.8 build 1, release date 2007-12-04

*Build 3.8.1 2007-12-04*

In this AIMMS 3.8 Maintenance Release, the following issues have been fixed or improved. Please note that almost all issues that have been fixed in the AIMMS 3.7.107 Software Update, have also been fixed in this AIMMS 3.8 Maintenance Release.

*New features*

Apart from the new features mentioned in this paragraph, you can find all new AIMMS 3.8 features listed in the previous chapter and in the next section, in the release notes for build 3.8.0.

- **CPLEX 11** can be used to generate a pool of solutions for a MIP problem. This is controlled by the **CPLEX** option **Do Populate**. See the **CPLEX 11** help file for more details.
- A separate RPC installation file has been added to AIMMS, which provides portable remote procedure call capabilities used by AIMMS components such as AIMMS web services, agent technology and remote solver sessions. This RPC installation includes the *OpenSLL* and *SSLeay* libraries, to provide the HTTPS transport for AIMMS web services.

*Improvements...*

- Several improvements have been made to make AIMMS 3.8 more robust with respect to issues which could lead to corrupt `.amb` files. This includes improvements in the field of importing and exporting source files.
- The option `Max_errors_during_file_read` has been added to the option category **AIMMS - Progress, errors & warnings - Stop criteria**. The range is

{1, maxint} and the default value is 1. When greater than 1, AIMMS will try to recover during a read from file statement and read subsequent data from the specified file.

- The versioning of the files messages.log and aimms.err in the log folder, as controlled by the option Number_of_logfile_backups, now ensures that the files created during the same AIMMS session get the same version number.
- The command line option --end-user (or -e) now acts as if you have an end-user license for AIMMS. Previously, this was only the case when AIMMS started a project from the command line as well.
- During a project export, Windows specific binary files are now converted to a format that is also recognized under Linux. Files that used a Windows format are: project files (.prj/.libprj), data manager files (.dat) and the multi-agent community setup file (usually .cfg).
- The **Index Of All Examples** has been extended to show which examples are new or have been extended in the latest Maintenance Release of AIMMS.
- The file _extractInfo that was generated by extracting from a .aimmspack file, has been renamed to _AimmsPackInfo. Additionally, it is now a hidden file. The file itself contains an explanation of its purpose.
- You can now define whether license and solver configuration settings are stored on a per-user *(CurrentUser)* basis or for all users *(AnyUser)* of a computer. The default for AIMMS 3.8 is AnyUser, for 3.7 it is CurrentUser.
- The behavior of the dumpfile sender has been altered, such that failure of transmission will not automatically remove the dumpfile anymore, leaving the possibility to send the dumpfile manually (by e-mail).
- If the %environment_variable% specification for the **Directory - Backup files** or **Directory - Listing and temporary files** options cannot be substituted, then the default folder names are used (Log and Backup).
- The **Messages/Errors** window now has a menu command to copy the text of the selected errors and/or warnings to the clipboard.
- AIMMS 3.8 (and future versions) will use their own list of recent projects, which is stored in the Windows Registry. Initially, this list of recent projects is taken from earlier AIMMS versions.
- AIMMS now offers an option to automatically create a backup of the entire project, when the original model has compiled successfully, directly after starting it. This file is called LastSuccessfullyStarted.bak and is placed in the backup folder. Per default, the option to create this file is on. It can be changed from the menu **Settings - AutoSave & Backups - Projects**.

*..Solving*

- The **progress window** now shows the number of generated integer variables.
- The option Print_presolve_status has been added to **CPLEX 10.1** (and higher). With this option you can display presolve status information for the variables and constraints in the listing file.
- AIMMS will add a new version of a solver automatically to the solver configuration if it detects a new version at startup. Also, a warning about this is displayed (once). Old solver versions are not removed from the solver configuration and the solver-to-model-type assignments are not changed.
- The procedure GMP::Instance::SetCallbackAddCut is now also supported by **XPRESS**.
- A direct reference to the solvers overview in the help file has been added under the new **Help - Additional Documentation - AIMMS Solvers Overview** menu item.
- The option Display_infeasibility_analysis has been introduced to show nonlinear presolve infeasibility information in the listing file.
- The following functions have been added to the GMP library:
  GMP::SolverSession::GetNodeNumber,
  GMP::SolverSession::GetNodeObjective,
  GMP::SolverSession::GetNodesUsed, and
  GMP::Solution::GetNodesUsed.
  For details, please consult the function reference.
- The **MINOS** link has been upgraded. **MINOS** can now be used in combination with GMP, **AOA** and the nonlinear presolver.
- The procedure GMP::Row::Generate now generates an execution error if it encounters a column that is not part of the GMP. The new optional argument autoAddColumn has been introduced to the procedure, which can be set to 1 to prevent these errors.

*...Math Program Inspector*

- The **Math Program Inspector** has been extended to support indicator constraints.
- The **Math Program Inspector** now supports stochastic programming models.

*...Model explorer*

- The **Ins** key can now be used to create a new entry in a tree-style property dialog.
- The **Database Mapping Wizard** has been made resizable.
- The **Tab** key can now be used to switch between attributes in the attribute form.

*..GUI*

- In **Network** objects with a GIS map as background, unnecessary retrieval of data is not performed anymore, in case some source has become totally invisible because all its layers have been set to *invisible*.

*...Pivot Table*

- The widths of the headers in the **Pivot Table** can now also be controlled through an AIMMS identifier.
- The **Pivot Table** properties have been extended with the possibility to specify a default column width (for all indices).

*...Multi-user*

- The **Font** selection dialog now allows you to copy and/or move fonts. In addition, the various buttons are disabled if the corresponding main project or library is read-only.
- The dialog which told you that the project and all its libraries are read-only while opening it in developer mode, has been removed.
- It is not possible anymore to have two separate **sections** and/or **modules** referring to the same source file.
- If you put the **localization** identifiers in the interface of a library, you can now select the localized text from that library in another library or the main model.

*Bugs...*

- The AIMMS XML functions now handle name changes properly.
- Sometimes, an end-user project could not find the **Menubar for project window**, as specified in the option category **Project - End-user menus**.
- The `Control-Shift-S` shortcut key for the **Run-Stop** menu command could disappear from the default menu when the `Control-Shift-S` key was assigned to another user menu item in a different menu.
- After loading a project containing `.aim` source files, AIMMS does not automatically mark this project as being modified anymore.
- AIMMS now properly retains all development passwords across model file exports.
- An **Indexed Page** object on a print page that contains sub pages larger than the object itself, was not printed at all.
- Switching focus from an *auto-hide* dockable window (for example the **Model Explorer**) did not always set the correct menus and toolbar.
- The **Database Mapping** wizard now properly handles suffices for the AIMMS identifiers.
- Exporting a `.amb` file after a model edit does not result in a corrupted `.amb` file anymore.
- The **Multi Development** example has been changed, such that the correct read-only attributes for the files are set upon extracting the `.aimmspack` file of the example. These read-only flags are used to illustrate the use of libraries.
- Displaying the total number of pages to print on a **print page** itself (for example a string like: `page 2 of 5`) did not always work correctly, or could even lead to a situation in which an infinite amount of pages would be printed.
- Clicking on **OK** directly after changing the queue name in the **Web Service Properties**, did not save the change.
- Identifiers in read-only modules were editable in a **View** window.

- If using VAR-licensing in combination with an encryption seed of 0, the **Protected** attribute of a **module** could get corrupted.
- Specifying a *page tree reference* as the startup page, did not open the referred page.
- When making a project backup, the `.prj` file was not always backed up correctly.
- The use of the **File Data Source** wizard could reset the project folder resulting in the model file being saved to the wrong location. This could also happen if AIMMS opened the corresponding dialog by executing a database statement like `read from table`.
- AIMMS will automatically create the solver configuration file (`.slv`) if AIMMS cannot find it at startup. Previously, AIMMS first had to be restarted to do a solve, otherwise it would generate an error that no solver was available.
- AIMMS did not update the **progress** window and could not be interrupted if **CPLEX 11** used dynamic search for solving a MIP model.
- Running the following action sequence: `MainExecution` - **Load Case** - `MainExecution` could cause a severe internal error at the end of the model generation.

*...GUI*

- In various page objects, a double-click did not always work correctly. This was the case when handling the new selection or the reverse links of the object, took longer to execute than the double click speed.
- For resizable pages, the scroll bars were not always displayed correctly if the size of the page was smaller than the minimum allowed size.
- When creating a `.aimmspack` file, the read-only status of files before they are packed is now taken into account properly.
- In a **Composite Table** object, it was not always possible to drag a column before the first column (so that it would become the first column).
- The drop down list in a **Selection Object** did not always have the correct width.
- Specifying the button action **Goto Next Page** (cyclic), while all sibling-pages are inactive, could result in an infinite loop.
- In the **Network** object, the option **Equal X & Y Scale** could update the identifier of the corresponding bound, while that identifier was not updatable (e.g. had a definition).
- Decreasing the number of individual bars in one column of a **2D Chart** object of chart type `Stacking Bar`, while the click label of the topmost individual bar is still being displayed, resulted in a fatal application error.
- Decreasing the number of stacked bars in a **2D Chart** object of chart type `Stacking Bar`, while the click label of the rightmost stacking bar is still being displayed, resulted in a fatal application error.
- In a **2D chart** object, the value of the property **Y-axis - Annotation - X-axis origin** wasn't always applied correctly upon opening a the chart.
- Double-clicking in a **Composite Table** in which making a selection is disabled, could lead to a fatal application error.

- In all GUI objects, you can no longer add procedures with element valued arguments, which are not of type `input`.
- Buttons with the **Flat Transparent** style did not always respond correctly when the foreground color changed.
- The **Slider** object now allows a non-updatable identifier for its **Identifier** property value. The slider will then show as disabled.
- When using checkboxes in the **Pivot Table** object, any reverse links were updated *after* the procedure specified as **Procedure upon change** had finished executing.
- The function `GeoFindCoordinates` has been disabled in the Maintenance Release of AIMMS 3.8, because the third-party server that delivered the data for this function, has been blocked. If this is resolved, the function will be enabled again in AIMMS.
- Importing pages or templates could result in an *unknown error*.
- In a **2D chart**, when using the **Multiplier** and/or **Constant** properties of the Y2-axis, the Y2-axis scaling was not always updated immediately.
- As an effect of the order in which elements have been added to the set, the internal order of elements in AIMMS can be different from the order as specified by the user. The sort order in the **Pivot Table** sometimes incorrectly reflected the internal sort order of AIMMS, instead of the order specified by the user.
- In a **2D Chart**, specifying a one-dimensional parameter for one of the **Appearance** properties of a two-dimensional identifier, sometimes yielded a fatal application error upon re-showing the chart.

*...Multi-user*

- When selecting a **User Color** in the **Properties** dialog of an object, it is now visible which colors belong to which library.
- Copying identifiers from a **library** or a **module** into the main model could result in an unnecessary `copy_` prefix in the resulting identifier name.
- When moving identifiers between **modules** and/or **libraries**, currently opened **Attribute** windows will be adjusted according to the new prefix. However, edit fields that are being edited and which have not yet been committed, will remain unchanged.
- The **Cross Library References** tool now correctly skips identifier names that are mentioned between the curly braces following the `data` keyword.
- Renaming a library in the model explorer is now also reflected in the **Library Manager**.
- Importing `.aim` files into a library module, could result in a fatal application error.
- Doing a **File - Export** action did not work correctly in combination with libraries and/or developer state files. Also, **One Way Encryption** did not work properly with this action.
- AIMMS sometimes wrongfully reported a name clash when importing a `.amb` file in a section of a library.
- **Project User Files** of a read-only library could still be edited via the func-

tion `FileEdit`.

- The **Library Manager** did not handle relative path names of model files correctly.
- When the `.aim` file of a library file could not be opened for writing, it was closed anyway causing a fatal application error.

*...AIMMS Viewer*

- A single AIMMS project can now be exported multiple times.
- In the **AIMMS Viewer**, the **Case Properties** were available by double-clicking in the **Data Manager**.
- In an end-user project you can now use **File - Export Viewer Project...** even if the project was not originally opened through a `.aimmspack` file. If it was opened through a `.aimmspack` file, the content of this file is used to determine which files need to be exported again. If started from a `.prj` file, then by default all files in the project folder(s) are exported, unless the developer of the project has saved the selection of files during a possible previous export.
- For an end-user system, the **Export Viewer Project** dialog box does no longer have the **One way encrypted** option.
- A newly created project was not exported correctly to a `.aimmspack` file. When unpacking it, the `.prj` file did not open.

*...Compiler & execution*

- An error could occur, when a parameter was defined by the return value of a procedure.
- Too many lines in the name change (`.nch`) files sometimes caused the compiler to crash.
- Indexed element parameters, used as an argument in identifiers, could have invalid data up after running a `CleanDependents` statement.
- The derivative evaluation of the `/$` operator was not correct when the right hand side of that operator is 0.0.

*...Stochastic Programming*

- A warning against stochastic variables with stage 0 has been added. The behavior of this warning can be controlled by the newly added option `Warning_stage_0_stochastic_variables` under **AIMMS - Progress, errors & warnings - Warnings - Optimization**. The possible values are `Off`, `Warning` and `Error`. The default value is `Warning`.
- The model editor now better supports changes in the **Stochastic** property of variables after a stochastic problem has been solved using *Benders for Stochastic Programming.*

*...Model Explorer*

- Formal arguments were sometimes shown unjustly in the **auto-complete list** of procedures/functions, leading to compilation errors.
- The tooltips generated by the syntax editor sometimes remained visible after the focus had already been lost.
- It was allowed to enter multiple lines in fields in which only single line values should have been allowed. This could lead to compilation errors.

- If changes in an attribute window imply that either more or less attributes should be displayed (e.g. by setting the **Stochastic** property of a variable, a **Stage** attribute must be displayed), the current window is now properly adjusted after clicking the **Check** button.
- During a **Find & Replace** action, you now get a message if the search reaches the end.
- AIMMS now closes a source file properly before deleting a node in the model tree which is associated with it (a section, for example).

*...Examples*

- In the **GIS Support** example an error occurred during startup.
- Opening the **Web Service Caller** example resulted in the error: "`Cannot find model file:  ..\StepByStepGuideProject\WebServiceCaller.amb`"

*...Windows Vista*

- On 64-bits Windows Vista systems, an **Indexed Page** object was not always resized correctly.
- AIMMS does no longer rely on the **Virtualization** feature of Windows Vista. There is no need for AIMMS to write in the `Program Files` folder or protected area of the registry. This means that for AIMMS on Windows Vista, files in the `Program Files` folder are now recognized as read-only.

## AIMMS 3.8 build 0, release date 2007-08-08

*Build 3.8.0 2007-08-08*

In this AIMMS 3.8 Release Candidate, the following issues have been fixed or improved. Please note that almost all issues that have been fixed in the upcoming AIMMS 3.7 Software Update, have also been fixed in this AIMMS 3.8 Release Candidate.

*New features*

- A new function `TestInternetConnection` has been introduced. It checks whether a certain URL can be reached (see the AIMMS Function Reference for more details).
- As part of the new Viewer functionality, a new function `IsRunningAsViewer` has been introduced, to detect whether the current AIMMS session is a Viewer session or not.
- The following new GMP functions have been introduced:
  `GMP::Solution::IsDualDegenerated`
  `GMP::Solution::IsPrimalDegenerated`
  `GMP::Column::GetStatus`
  `GMP::Row::GetStatus`
- The function reference has been extended with a list of suffices.
- With the new menu item **Close All** in the **File** menu, you can close all opened windows within AIMMS.
- The **menu builder** now has an extra **Default Menu bar** for the **data manager** which contains specific menu commands that are used to manipulate the **data manager tree**.

- AIMMS now offers facilities to read, write and display the Comment property of a **case**, not only via the **Properties** dialog of the **Data Manager**. For this, the two functions DataFileGetComment and DataFileSetComment have been introduced. Next to that, you can now use the keyword Comment in the AIMMS option Project - Data Manager - Data Dialog columns.
- **Libraries** can now be protected with a VAR license and you can specify a developer password for a library.
- When a fatal application error appears, AIMMS now offers the possibility to send the dump files, which are created automatically in such situations, to Paragon. This increases the chance that we will be able to fix problems which are hard to reproduce.
- A new tool has been added: **Library Cross References** (in the **Tools** menu). With this tool, references across various libraries and the main model can be inspected easily. There are two main tabs in the tool dialog: one which shows an overview of all items that are declared in the selected library and used in another library, and one which shows an overview of all items that are used in the selected library, but declared in another.
- The AIMMS localization feature can now be used on single libraries.
- In the page objects, if an identifier has a unit, you can now override the unit string that is being displayed via a static string or scalar string parameter.

*Improvements*

- All AIMMS examples are now distributed in the form of .aimmspack files, allowing you to locate your own preferred extract location. This means that it is now also possible to modify your examples under Vista, since you can extract them to a different location than the Program Files directory.
- The pre-declared set DatabaseWriteModes has been added to AIMMS. The DENSE keyword has been added as a modifier of other modes of the write to table statement. See for further details section 23.2
- Several improvements have been made to the saving of .amb files.
- In the **Identifier Selection** dialog the special entry Numerical Value has been renamed to [Numerical Value] so that it appears at the top of the sorted list.
- When double-clicking on an entry in a **Composite Table**, AIMMS will first try to run the **Procedure upon Double Click**. If that is not possible, it will try to run the default right mouse menu command, and, if that is not possible either, it will set the cell in edit mode (which is a new feature).
- The edit field for entering an expression in a GUI object now has a non-proportional font.
- If an .amb file is read-only, this is now visualized in the **Model Explorer**, both for the main model file as for source files of section and module nodes.
- The **Project options** dialog (**Settings** menu) is now resizable.
- It is now allowed to make changes to the **Data Management Setup** while

the current data manager file is marked as read-only.

- After finishing a text find sequence, the message ”Pattern not found.” has been replaced by ”Search reached end. No (further) matches.”.
- When you type the name of a procedure and hit `CTRL - (SHIFT) - SPACE` twice, the auto-complete feature of AIMMS now automatically inserts the formal arguments for procedures and functions for you.
- To increase the flexibility in specifying the location of the user database that is linked to a certain project, the `.usr` file is now interpreted as a link to another user database, as long as it is an ASCII file that contains the location of another valid user database.
- The **Data Manager** dialogs for saving, loading and selecting cases or datasets have been enlarged.
- When editing element parameters in a **scalar**, **table**, **composite table**, or **pivot table** object, a **dropdown button** is displayed now, which allows you to select an element parameter instead of having to type it yourself.
- Keyboard shortcuts have been added for collapsing (`CTRL -` for current, `CTRL SHIFT -` for all) and expanding (`CTRL +` for current, `CTRL SHIFT +` for all) of outlining nodes in the editor.
- **Color schemes** can now be specified through subsets of `AllColors`. This applies to **2D charts**, **Bar Chart**, **Curve** and **Gantt Chart** objects.
- The thousand separator and the decimal point character, that can be used in the click label of **2D and 3D charts**, now take their value from the options which can be set in the options tree **(Project - Appearance)**.
- The solver configuration file is now editable. AIMMS will look first in the project directory for the file `solvers.slv` and use that file if it exists.
- The copy/paste functionality of multiple cells in the **(Pivot) Table** has been adjusted:
  - when you select 1 cell in the table, and then do a paste while there is a ’matrix of values’ on the clipboard, then the selection in the table is automatically extended to the sizes on the clipboard.
  - when you have selected a range of values in the table, and then do a copy while there is only 1 value on the clipboard, then the entire selected range is filled with that value.
- The element sorting in the **Pivot table** has been extended with the possibility to sort elements alphabetically.
- The number of decimals for an identifier in the **Pivot table** can now be specified by a multi-dimensional parameter (instead of a scalar only).

*Bugs...*

- The AIMMS startup window did not have the correct size when using the AIMMS startup option `ide-style-95` (which is the only supported style in the 64-bits version of AIMMS).
- Altering the IDE Style in one project could also affect a project that was opened subsequently (without exiting AIMMS in between).
- The **Print Setup** dialog now has the correct title: ”Print Setup”, and an **OK** button to confirm the selection (on some Windows systems this was

a **Print** button, while nothing was printed yet).

*...Libraries*

- It is no longer allowed to move a page or template to another library while that page is currently open.
- When moving an entire subtree of pages or templates to another library, AIMMS only checked for private symbols at the root page/template of that subtree. Now, all pages underneath are also checked.
- You can no longer move **menu builder** items between libraries if you have a page open that is referring to one of the selected menu bars, tool bars or popup menus. You get a warning that you must close the page first.
- If you rename or add an identifier with name `X` in the scope of a Library/Module while that name `X` is already present in the global scope, you will now get a dialog in which you can decide whether the already present occurrences of `X` should either reference the newly added `X` or should still reference `X` in the global scope.
- When adding a new library, AIMMS will generate a prefix for it. It now also ensures that this prefix is different from the prefixes of all libraries added before.

*...Syntax editor*

- Some default suffices like `.unit` and `.txt` were colored incorrectly.
- Sometimes, base units were not recognized correctly in editor fields, and, as a result, were shown incorrectly colored.

*...GUI*

- When the contents of a file were displayed in a **Scalar** object and had been scrolled down, the content would be scrolled back to the beginning each time the page regained focus.
- The **pivot table** will now automatically scroll to the focus position when the focus is changed from outside (using the function `PageSetCursor`).

*...Compiler*

- If `S` is a set, with index `i`, and `T` is a subset of `S`, with index `j`, and `P(j)` is a parameter, then the statement `A := sum( i, P(i) )` should only consider the elements in S that also exist in T. AIMMS now does this and two options have been added to support this change. The option `Execution_Respects_Declaration_Domain` has been added to the **AIMMS - Backward Compatibility** option category. The range of this option is { `Off`, `On` } and the default is `On`. When you want to use the old behavior, you should switch this option to `Off`. The option `Warning_Difference_Execution_Respects_Declaration_Domain` has been added to the options category of **Execution Warnings**. The range of this option is { `Off`, `Warning`, `Error` }, with default value `Off`. When there are differences in your model results, you may want to switch this option to `Error` or `Warning` in order to find the cause of those differences.