

AIMMS 3.7

Release Notes for Win32 Build 3.7.110

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

Paragon Decision Technology

September 16, 2008

Contents

Contents	2
1 System Requirements	4
1.1 Hardware and operating system requirements	4
1.2 ODBC and OLE DB database connectivity issues	5
1.3 Viewing help files and documentation	6
2 Installation Instructions	8
2.1 Installation instructions	8
2.2 AIMMS 3 licensing	10
2.2.1 Personal and machine nodelocks	11
2.2.2 Installing an AIMMS license	14
2.2.3 Managing AIMMS licenses	16
2.2.4 Location of license files	17
2.3 OpenSSL license	19
3 Getting Support	22
3.1 Reporting a problem	22
3.2 Known and reported issues	23
4 Release Notes	24
AIMMS 3.7 build 110, release date 2008-09-16	24
AIMMS 3.7 build 109, release date 2008-05-15	26
AIMMS 3.7 build 108, release date 2008-02-20	28
AIMMS 3.7 build 107, release date 2007-11-20	31
AIMMS 3.7 build 106, release date 2007-08-14	39
AIMMS 3.7 build 105, release date 2007-06-25	43
AIMMS 3.7 build 104.4834, release date 2007-05-07	48
AIMMS 3.7 build 104, release date 2007-04-20	48
AIMMS 3.7 build 103, release date 2007-02-15	54
AIMMS 3.7 build 102, release date 2006-12-14	60
What is new in AIMMS 3.7	67
Integrated GIS support	67
Syntax editor	68
Stochastic programming support	68
Parallel solver sessions	69
Improved multi-developer support	70

Pivot table extensions	70
Web services extensions	71
Solver additions and updates	71
Supported platforms	72
Improved functionality and bugs	73

Chapter 1

System Requirements

This chapter discusses the system requirements necessary to run the various components of your Win32 AIMMS 3.7 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.

System requirements

1.1 Hardware and operating system requirements

The following list of minimum hardware requirements guarantees that your Win32 AIMMS 3 system will be able to run small- to medium-sized AIMMS projects in a reasonably comfortable manner. Note, however, that this may also depend on the number of other applications that are running concurrently with AIMMS.

Hardware requirements

- Pentium IV processor
- XGA display adapter and monitor
- 256 Mb RAM
- 1 Gb free disk space
- A parallel or USB port (for installing the AIMMS dongle)

When you try to run an AIMMS project for larger data sets on a PC with a small amount of installed or available RAM, you may find that the increased memory requirements of running large models give rise to extensive disk swapping. In general, this will have a dramatic effect on the overall performance of both AIMMS and the other applications running concurrently with AIMMS. In such a case you are advised to install sufficiently additional RAM to run all applications comfortably again.

The Win32 version of AIMMS 3.7 is designed to run under

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

Supported Windows versions

The Win32 version of AIMMS 3.7 Unicode is supported under

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

Running the Win32 version of AIMMS 3.7 versions under the 32-bit Windows versions 95/98/ME/NT 4.0 is not supported.

AIMMS 3.7 has been tested on the latest available beta versions of Windows Vista, and will be supported on Windows Vista as soon as it is officially released by Microsoft.

Windows Vista support

1.2 ODBC and OLE DB database connectivity issues

Before you can start using the database connectivity features built into the Win32 version of AIMMS 3.7, the following ODBC and OLE DB components need to be installed on your computer:

Database connectivity

- 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.

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 ODBC 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.

Microsoft Data Access Components

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.

Data Access Components installation

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 ODBC 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.

*Oracle ODBC
drivers and OLE
DB providers*

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.

*Prefer native
Oracle over
MDAC drivers*

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

OLE DB issues

- 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

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.

*AIMMS
documentation*

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.

*Acrobat Reader
installation*

Chapter 2

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

Installation instructions

- 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.

2.1 Installation instructions

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

The AIMMS 3 CD-browser

- start the AIMMS 3 or AIMMS 3 Unicode setup program,
- 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).

The download area of our web site www.aimms.com only provides the AIMMS 3 setup program of the latest AIMMS 3 release. To obtain the latest versions of the setup programs of third-party software and system components required in conjunction with AIMMS 3, follow the links provided on our web site (or in Chapter 1 of these Release Notes).

Obtaining AIMMS 3 from the web

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 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)

The AIMMS 3 setup program

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.

*Solvers in
installation*

Solver	Version	Win 32		Win 64 *	Linux 32 *
		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	-	-	√	-
CPLEX	8.1	-	√	-	-
	9.0	-	√	-	√
	9.1	√	√	√	√
	10.0	√	√	√	√
	10.1	√	√	√	√
	11.0	√	√	√	√
KNITRO	5.1	√	√	√	√
LGO	1.0	√	√	-	-
MINOS		√	√	-	-
MOSEK	3	-	√	-	-
PATH	4.4a	-	√	-	-
	4.6	√	√	-	-
SNOPT	6.1	√	√	√	√
XA	13	-	√	-	-
	14	√	√	√	√
XPRESS	15	-	√	-	-
	16	-	√	-	-
	17	-	√	-	-
	18	-	√	-	-

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

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

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

*Administrative
privileges
required*

- the installation of the device drivers required to access the AIMMS hardware dongle (Win32 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 offers the following two types of licenses:

*AIMMS 3
licensing*

- single-user licenses, and
- network licenses.

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

Single-user licenses can be used by a single user on a single computer. To enforce the single-user character, AIMMS 3.7 requires that single-user licenses be protected by either

*Single-user
license
protection*

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

When ordering the Win32 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 Win64-x64 platform does not support dongles.

You can choose

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

*Pros and cons of
dongles*

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

Pros and cons of nodelocks

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

Physical shipments

If you request your license to be protected by a dongle, an AIMMS dongle is sent to you along with your AIMMS 3 CD-ROM. If you upgrade from AIMMS 2.20, you can continue to use your existing AIMMS 2.20 hardware dongle. 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 3.7 setup program will only install the required device drivers for accessing the grey and purple Sentinel dongles. If you still use the green Activator dongles supplied with AIMMS 2.20, you can obtain the required drivers separately from our FTP site <ftp.aimms.com>.

AIMMS dongles

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

Network licenses

2.2.1 Personal and machine nodelocks

AIMMS offers two types of nodelocks:

- personal nodelocks, and
- machine nodelocks.

Two types of nodelocks

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

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

*Personal
nodelock*

- Personal nodelocks can be transferred to another computer 3 times per 24 hours. This allows you to take your AIMMS license home in the evening and back to work the next morning without any problems.
- Personal nodelocks have a limited lifetime of 60 days, and should be renewed within that period to extend the lifetime to its full 60-day period. If the nodelock is not renewed within its 60-day lifetime, this does not invalidate your AIMMS license in any way—you only have to renew your nodelock prior to being able to use your AIMMS system again. Note that the renewal process does not require any manual intervention, as AIMMS will try to automatically connect to our internet license database to renew your nodelock once every day you are using AIMMS.
- Both activation and nodelock renewal of personal nodelocks require a working connection to the internet. As a consequence, in the absence of an internet connection you can continue to work uninterrupted for a period of 60 days, before an internet connection is required to renew your nodelock.
- With every activation or nodelock renewal AIMMS will also update your license files if new license files are available (e.g. if your system is in maintenance), and will inform you of any messages that are available for you in our database.
- Because of their volatile nature, PDT will replace a personal nodelock without any questions asked in case of loss of or damage to your computer.
- You can switch your personal nodelock to a machine nodelock at any time.

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

*Machine
nodelock*

- Machine nodelocks can be transferred to a replacement computer 3 times per 365 days.
- Machine nodelocks have an unlimited lifetime (unless deactivated).
- Machine nodelocks can be either activated online if your computer is connected to the internet, or offline through the license activation area on the AIMMS website.
- License files will only be retrieved when the machine nodelock is activated, or by explicit request.
- In case of failure, PDT will, in principle, only replace machine nodelocks on the same computer.

- Once you have chosen for a machine nodelock, it is not possible to switch back to a personal nodelock.

Although a personal nodelock 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.

Privacy

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

Internet connection and proxy settings

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

Automatic configuration scripts

If AIMMS does not succeed in automatically detecting the proxy settings that apply in your network environment, AIMMS also allows you to manually set the proxy settings during the activation process. If the online activation process does not succeed directly, AIMMS gives you the option to either continue with an offline activation process, or to manually supply the proxy settings that apply to your network environment through the dialog box illustrated in Figure 2.1. In this dialog box you can choose between

Manual proxy setting

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

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

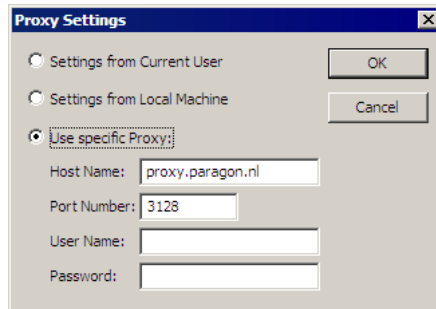


Figure 2.1: The AIMMS Proxy Configuration dialog box

2.2.2 Installing an AIMMS license

When you start up AIMMS 3.7 for the first time after installation, AIMMS will open the **License Configuration** dialog box illustrated in Figure 2.2. Through

Managing your AIMMS licenses

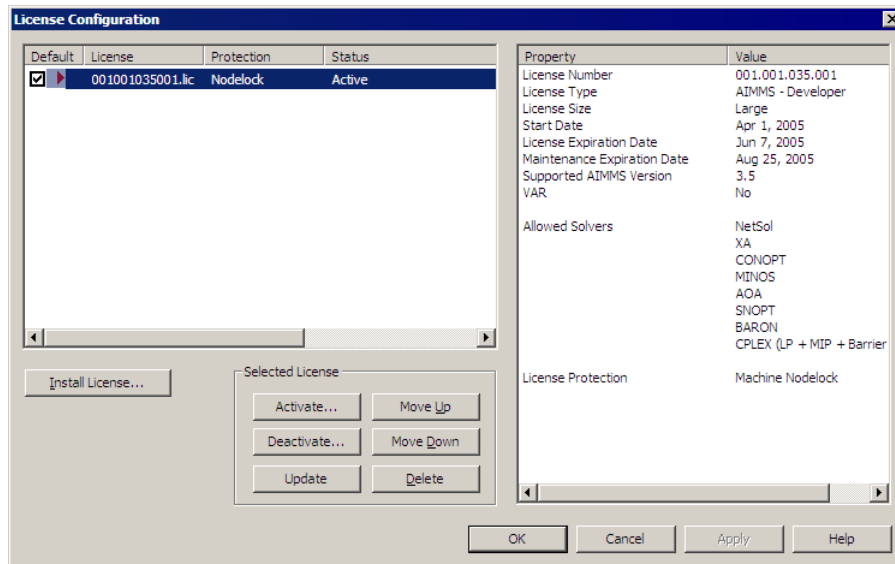


Figure 2.2: The License Configuration dialog box

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

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

Installing a new license

- existing AIMMS 3.6 licenses,
- nodelocked licenses,
- dongled licenses,
- network licenses,
- evaluation licenses, and
- student licenses.

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

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

*Single-user
nodelocked
licenses*

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

You have the choice to request a personal nodelock or a machine nodelock. A personal nodelock must be requested online, a machine nodelock can be requested online or offline. Refer to Section 2.2.1 for a more detailed introduction to personal and machine nodelocks.

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

*Single-user
dongled licenses*

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

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

*Network
licenses*

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

To install an AIMMS evaluation license you need the following information

*Evaluation
licenses*

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

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

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

Student licenses

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

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

2.2.3 Managing AIMMS licenses

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

Managing multiple AIMMS licenses

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

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

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

Default licenses

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

Transferring licenses

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

You can deactivate an active nodelock using the **Deactivate** button in the **License Configuration** dialog box. Deactivation will only succeed if there is no

conflict with the transfer limit for the given nodelock type. This makes sure that there will never be a problem activating a deactivated license. After successful deactivation the license will not be removed from the list but be marked as inactive. If the license is not active on any computer, you can reactivate the license through the **Activate** button.

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

*Emergency
nodelocks*

2.2.4 Location of license files

AIMMS keeps its license and configuration files in the folder

Paragon Decision Technology

*Location of
license files*

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.

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

*Do not move
nodelock files*

You can specify whether the license and solver configuration that AIMMS uses is the same for any user of the machine, or different for each individual user. To prevent problems when running AIMMS as part of a computer service, AIMMS 3.7 will by default use the same configuration for any user. To modify this behavior, you should edit the file `UserDistinction.cfg` in the common `Config` folder. In the file `UserDistinction.cfg.default`, straightforward directions are given on how to edit it. If no `UserDistinction.cfg` file exists, AIMMS will use the `UserDistinction.cfg.default` file instead. You can use this file as a base for setting up your own configuration.

User specific configuration

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

Any User

In the scenario where each specific user of the pc has its own license and solver configuration, the configurations that are modified by a user are stored in the local application data folder. On Windows 2000, Windows XP and Windows Server 2003, this local folder is usually located at:

Current User

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

On Vista it is:

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

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

Accessing configuration files

Inside the `Config` folder of the `(any)user` folder, each major AIMMS version (3.5, 3.6, 3.7, etc.) will create its own specific subfolder when it needs to write a configuration file. During an attempt to read, AIMMS will first look for the specific file in the folder that matches its own major version number, and otherwise it will subsequently search through the folders of previous versions. In other words, when you upgrade to a new AIMMS version, initially your configuration will be the same as the one you were using for the previous AIMMS version, but if you change something in the configuration, this will only affect the configuration of the AIMMS version you are working with.

Version dependent configurations

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

*Network License
Client Files*

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

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

2.3 OpenSSL license

The 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
included*

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

OpenSSL license

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.

SSLey 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.

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 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 acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgment: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS 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 AUTHOR OR 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.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Chapter 3

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.

Getting support

3.1 Reporting a problem

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.

Problem fixing

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

Reporting a 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

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.

*Problem
processing*

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.

*Modeling
support*

3.2 Known and reported issues

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

Known issues

- The Program Files directory is protected by Windows Vista. As a result, users cannot make changes to the files and folders located in this directory. In case you open an AIMMS example located in the Examples directory and you make changes to the example, Windows Vista saves these changes in the so-called *Virtual Store*. For AIMMS 3.8 the examples are stored in .aimmspack files and need to be extracted to a writable location first. *Virtualization* is turned off for AIMMS 3.8, because there is no need to change the files in the AIMMS 3.8 folder.
- On Windows Vista 64-bits systems, in AIMMS 3.7 an indexed page object does not always resize correctly.
- Web Services in AIMMS 3.7 and User Account Control on does not work, because it writes one configuration file to the AIMMS Bin directory (usually located under Program Files).
- The screen (progress window in particular) is not updated during execution once AIMMS lost focus, by swicthing to a different application.

Windows Vista

Chapter 4

Release Notes

This chapter contains a description of the new features developed for AIMMS 3.7. In addition, it contains the release notes of all AIMMS 3.7 releases. For each release, the following items are listed:

Release notes

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

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 22.2 of the User's Guide.

*AIMMS 3.7
versions*

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

AIMMS 3.7 build 110, release date 2008-09-16

In this AIMMS 3.7 Software Update, the following issues have been fixed or improved.

*Build 3.7.110
2008-09-16*

- **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 highly recommend this Software Update.**

IMPORTANT

- The **Library Manager** dialog box could crash if the model was not compiled yet. *Issues...*

- 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**. *... GUI*
- 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).

- In rare cases, a fatal application error could occur after executing a too complex database query from within AIMMS. *... Databases*

- 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. *... Compiler and Execution*
- The function `ReferencedIdentifiers` now also looks in encrypted source.
- 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.
- AIMMS now verifies that the parameter `ep` in the expression `P(ep-1,j)` indeed has dimension 0.
- Macro** identifiers now properly handle expressions of the form `X(ep(i)-1)` (where `ep` is an element parameter).
- The value 0 for the option `Put_Number_Width` was not properly handled.

- The computation of runtime computed units in **if-then-else** expressions involving variables with the `inline` property, did not always work properly. *... Units of Measurement*
- 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.

AIMMS 3.7 build 109, release date 2008-05-15

In this AIMMS 3.7 Software Update, the following issues have been fixed or improved.

*Build 3.7.109
2008-05-15*

- The syntax error message for expected identifiers in a subroutine call with a formal argument of type InOut, Output or Handle, has been rewritten.
- Upon a project close/project open action, the memory was not always properly initialized.
- A compiler error was unjustly issued when inline variables with an index domain condition were used in the righthand side of assignments.
- Some small memory leaks have been addressed in the areas of procedure calls, runtime evaluation of unit consistency, explicit zeroes in tabular data input and accumulating error messages.
- In rare circumstances the Ord function with a second argument was not accepted by the compiler.

Issues ...

- The option Run_only_once for the **Procedure upon change** in a **Gantt Chart** worked opposite to what may be expected.
- Deleting assertions in a **Gantt chart** could lead to errors.
- When using **assertions** in a **Gantt chart**, the **procedure upon change** was not always executed.
- If the duration identifier in a **Gantt chart** is integer valued, then trying to drag the left side of a bar only modified the start of the bar, and not the duration as well.
- The ActiveX **Calendar** object allowed a procedure with an argument to be specified for the On Change procedure, which it shouldn't. Executing this procedure led to a fatal application error. Only procedures without arguments are allowed now.
- In the **Scalar** object, displaying the value of multiple cases went wrong if the number of cases exceeded 20.
- When reopening a **Pivot Table**, in which the indices of a color parameter do not match the indices of the corresponding identifier in the table, a fatal application error could occur. Such a situation probably happened after changing the indices in the model explorer, and then reopening the page with the **Pivot Table** on it.
- Too small **Indexed page** objects could lead to a fatal application error.
- When opening the right-mouse menu of a page object, which refers to a default AIMMS menu, a fatal application error could occur.

...GUI

- Trying to change the default domain index display order in the **Math Program Inspector** (on the **Domain Index Display Order** tab of the **Variable Properties** or **Constraint Properties** dialog box) without having selected anything, resulted in a fatal application error. *... Math Program Inspector*
 - Using the **Variable Properties** or the **Constraint Properties** dialog box in the **Math Program Inspector** could lead to a fatal application error after the user had switched from the **Domain Index Display Order** tab to the **Analysis Configuration** tab and back again.
 - Infeasible models with program status 'Integer Infeasible' or 'Infeasible or Unbounded' did not always report the correct number of infeasibilities in the **Math Program Inspector**.

- An unneeded prefix of an identifier in a module inside a library is not generated anymore. *... Libraries*
 - The name change support for GUI expressions on pages inside libraries has been improved.
 - Entries in name change files that referred to deleted libraries could corrupt AIMMS' internal memory.

- AIMMS now empties the status file whenever **CPLEX** is initialized. *... Solving*
 - AIMMS sometimes incorrectly wrote messages to the **CPLEX** status file if the option `Solver.listing.messages` was set to `Never`.
 - Stochastic rows which are not part of the root scenario could have their elements in a non-increasing order. This was not according to the interface protocol to solvers.
 - In some rare cases the **Math Program Inspector** start values were not used directly by **CPLEX**, even if the start values defined a feasible integer solution.

- Running a Microsoft Access query with parameters from within AIMMS, could lead to a severe internal error or a fatal application error. *... Databases*
 - Using an SQL Query in AIMMS on a datasource which was not Oracle, MS Access, MS Excel, MySQL, MS FoxPro or SQL Server, could lead to a fatal application error.

- In rare cases, installing the AIMMS Excel Add-In from the `AimmsExample.xls` workbook, caused an error in the Visual Basic function `Right$`. *... Excel*
 - Using `ExcelAssignTable` with all optional arguments at their default value, in combination with an Excel workbook which contained at least one named range, resulted in a fatal application error.
 - Calling either of the functions `ExcelColumnName` or `ExcelColumnNumber` resulted in a fatal application error in the Unicode version of AIMMS.
 - In some cases, installing the AIMMS **Excel-Add In** raised an error message in a Visual Basic line regarding the function `Right$`.

AIMMS 3.7 build 108, release date 2008-02-20

In this AIMMS 3.7 Software Update, the following issues have been fixed or improved.

*Build 3.7.108
2008-02-20*

- The Excel functions `ExcelColumnName` and `ExcelColumnNumber` have been adapted, such that they now support Excel 2007, which has a maximum of 16384 columns instead of the 256 columns of previous Excel versions. *Issues ...*
- Data pages displaying variables and constraints will automatically add suffices if the underlying data is refreshed by, for example, a solve.
- The **Flat File View** of a model was not printed correctly.

- In rare occasions, changing the symbol appearance properties of the **Y/Y(2)-axis** identifiers in an **ActiveX 2D Chart**, could lead to a fatal application error. *... GUI*
- When not using the **Tabbed MDI** style nor the **Docked Windows** style, a minimized progress window did not reappear when pressing Ctrl-P during a run.
- Using the menu command **File - Open - Model** from within a right mouse menu on a page, could lead to a crash of AIMMS.
- Using a multi-line expression as a value of the **Hidden** option of a page, did not always work correctly. It also caused the list of **Used Identifiers** of a page to be incorrect.
- The **Order by** attribute of a set containing only one element, was never evaluated, resulting in an out of date status. All definitions depending on such a set were also considered to be out of date, even after evaluating. This could decrease the performance.
- The printing of **Indexed Page** objects has been improved.
- In some situations a **Pivot Table** containing multiple identifiers could incorrectly generate a message that the maximum number of entries was exceeded.
- When creating a multi cell selection in the **Pivot Table**, dragging the mouse from the grid area to either the row or the column area and then copying the selected cells to the clipboard, could result in a fatal application error.

- The case loading support for cases created with AIMMS 2.20 has been improved with regard to **ReducedCost** information of variables. *... Cases*
- `CaseCreateDifferenceFile` now returns 1.0 instead of 0.0 when it was run successfully.
- Previously, the shadow prices of defined variables were not stored in cases.

- The model explorer does not allow the creation of procedures with names equal to keywords (such as `for`, `if`, etc) anymore inside library modules. *... Model Explorer*
 - A severe internal error could occur when deleting a set declared in a module, where the set is also used as the range of an element parameter and that element parameter is used as an argument in an identifier (e.g. `MyIdentifier('ElementValue')`).

- The functionality of `DebuggerBreakpoint` is improved such that it allows the user to continue using the debugger even when he steps out of the procedure that called the intrinsic function `DebuggerBreakpoint`. *... Debugger*
 - Trying to compile (F5) with the debugger on a breakpoint, resulted in duplicate and incorrect dialog messages.

- Due to an error in the handling of data of external function calls, repeatedly solving generated mathematical programs, which have been created using `GMP::Instance::Copy` could result in a severe internal error. *... Solving*
 - The option `Feasibility_pump_heuristic` has been added for **CPLEX 10.1**.
 - Opening the **Math Program Inspector** could result in a fatal application error if a variable or constraint was emptied after the solve.
 - The computation of Hessian was not always correct when one of the functions called in the row was `DistributionCumulative` or `DistributionInverseCumulative`.
 - AIMMS did not respond anymore when **CPLEX 11.0** used dynamic search to solve a MIP model.
 - The listing file was not closed after creating a new project, causing a sharing violation when it was opened afterwards.
 - When the **CPLEX 11** option `MIP_search_strategy` was set to 'Automatic' and option `Show_branch_and_bound_progress` was switched on, then AIMMS would incorrectly set the option `MIP_search_strategy` to 'Apply branch-and-cut'.
 - There was a memory leak in the matrix generator where AIMMS interchangeably generated the matrix completely and incrementally, and during the incremental regeneration additional rows were generated.
 - In AIMMS 3.7 the dialog box for interrupting a run (menu command **Run - Stop**), did not by default have the checkbox **Only Interrupt Current Solve** checked, like previous AIMMS versions.
 - The derivative evaluation of the `/ $` operator was not correct when the right hand side of that operator was 0.0.
 - When a model became infeasible after adding variables and constraints via the model explorer to the application, a severe internal error could occur.

- After removing the index domain attribute of a newly created indexed set without a filled subset attribute, the data page was shown as if the set was a subset.
 - Profiling support has been added for the evaluation of inline variables at the end of a solve.
 - Recompiling a model with inline variables could result in a message from the compiler, about variables that are used in the index domain.
 - The data of parameters indexed over compound sets could not be read out of composite tables in a text file.
 - AIMMS now issues a compile time error message when assigning a one-dimensional set to an indexed set without using the arguments of the indexed set.
 - When identifiers were moved to another namespace, the corresponding elements were not properly renamed in the set `AllIdentifiers`.
 - When the range of a running for loop index is a defined set and that set is temporarily modified in a called procedure or via a **Case Load**, the defined set might not be updated in time when that for loop index is used as an argument in an identifier.
 - When indexed element parameters are used as arguments of other indexed identifiers, AIMMS uses an optimization where the element parameter is stored as a relation. In rare circumstances this optimization could lead to a performance degradation instead of a performance improvement.
 - Indexed element parameters used as arguments in identifiers, could have incorrect data after a `CleanDependents` statement.
 - AIMMS erroneously used matrix update technology in the second run of in the following action sequence: call to a procedure - **Load Case** - call to the same procedure. This could cause a severe internal error at the end of the generation.
 - A severe internal error could occur when the range of a running index is both adapted inside a constraint index domain and inside a for loop index domain, where a solve statement was called inside that for loop.
 - The computation of $(1 / \$ \text{ inf})$ erroneously returned 1 instead of 0.
 - Both the compiler and the execution engine have been improved with regard to the comparison of multi-dimensional sets (relations) when these sets are obtained via case referencing.
 - There was a memory leak in the function `ReferencedIdentifiers`.
-
- The Linux version of AIMMS now uses a mapping, to map the `tzname[0]` from Linux to a Windows time zone name.
 - Time zone conversion to dates will no longer be performed for the formatting of calendar elements to calendars of granularity century, year, month or day, because time zone conversions only make sense on an 'hour' granularity or smaller. Note: A calendar with unit [day] has granularity 'day', but a calendar with unit [24 * hours] has granularity 'hour'.

... Compiler and Execution

... Time Based Modeling

- The function `TimeslotCharacteristic` did not work correctly when a non-default time zone was used around a daylight saving time change.
- For the elements `UTC`, `Local` and `LocalDST` there is now data in the predeclared identifiers `LocaleTimeZoneName` and `LocaleTimeZoneNameDST`.
- The unit casting of a number inside a for loop was not always correct, when the unit contained an indexed unit parameter.
- Unit analysis of if - then - else expressions is now postponed from compile time until run-time. E.g. a parameter `A(i)` with unit `UP.A(i)` could have definition `if (ord(i)=1) then 1[m] else 2[kg] endif;`. Unit casting at compile time could give you a unit warning even though `UP.A` would be consistent with the result.
- The units of variables used as arguments in external functions, are now handled correctly by the AIMMS compiler.
- Assignment statements that required unit analysis for individual elements could result in a severe internal error.
- AIMMS now gives a proper error message when the unit '-' is used outside the quantity `SI.Unitless`.
- The GMP functions `GMP::Column::SetLowerBound`, `GMP::Column::SetUpperBound` and `GMP::Column::Freeze` passed an unscaled value to the solver if the variable had a unit.
- On Windows Vista, when editing the value of an element parameter in a page object, the AutoComplete-dropdown list of the edit field caused a fatal application error. If you have explicitly disabled this feature through setting the registry value `ElementAutoComplete` in `HKEY_CURRENT_USER\Software\Paragon Decision Technology\Aimms 3\SYSTEM`, you can now safely remove this registry value, such that the autocompletion is re-enabled.
- On Windows Vista, entering an invalid number (for an identifier without a range) in a **Pivot Table** could result in a fatal application error.

... Units of
Measurement

... Windows
Vista

AIMMS 3.7 build 107, release date 2007-11-20

In this AIMMS 3.7 software update, the following issues have been fixed or improved.

*Build 3.7.107
2007-11-20*

- The performance of the syntax editor has been improved by not creating unnecessary internal symbols anymore for suffices such as `.Stochastic`, `.ExtendedConstraint` and `.ExtendedVariable`.
- **CPLEX** version 11 has been added to AIMMS. It features dynamic search, a new search strategy for MIP models which proved to be very effective for many (large) models. Dynamic search is controlled by the **CPLEX** option **MIP Search Strategy**. See the **CPLEX 11** help file for more details.

Improvements

- AIMMS now warns when a stochastic parameter is used in the index domain of a variable or constraint. This warning is issued because AIMMS generates the stochastic mathematical program for only one scenario. In order to present the full stochastic mathematical program, it stores the differences with respect to the other scenarios. This is usually equivalent to generating all scenarios, except when the index domain of a variable or constraint contains references to stochastic parameters. In the option category **AIMMS - Warnings - Compilation** there is a new option `Warning_Stochastic_Parameter_In_Index_Domain_Variable_Constraint` that has range { Off, Warning, Error } and default Warning which controls the warning as explained here.
- `Card(IndexIdentifiers)` returned the cardinality of the set `AllIdentifiers`. Now however, it returns the cardinality of the identifiers referenced by `IndexIdentifiers`. This was already the behavior of `Card(IndexIdentifiers, 'Level')`. This also holds for `ActiveCard`.
- `Card(IndexIdentifiers, someSuffix)` has been adapted for the dependency definition graph. It now depends on all identifiers that can appear on the left hand side of an assignment statement (sets and parameters without definition, and variables and constraints). Please check the AIMMS Function Reference for more details.
- Both the AIMMS option tree and the AIMMS compiler now accept { off, on } as range for the off/on typed options in the category **Project**.
- The evaluation of the + (union) and * (intersection) operators has been improved in situations where these are used on set valued subexpressions, when the result is not assigned to a set.
- The suffix `.SolutionTime` now has the unit [seconds] if and only if this unit has been declared, and the (new) option `Solution_Time_Has_Unit_Seconds` has been set to On. Otherwise `.SolutionTime` is unitless. For AIMMS 3.7, the default value of the option is Off.
- The syntax for the right hand side of an assignment to unit parameters starting with a square bracket has been extended to include other unit parameters as well. For example: in previous AIMMS versions the assignment statement


```
unit_par_a := [ km ] ;
```

 was allowed, while the assignment statement


```
unit_par_a := [ km * unit_par_b ] ;
```

 was not allowed (but now it is). Note that in all AIMMS 3.7 versions the statement


```
unit_par_a := unit( km ) * unit_par_b ;
```

 was allowed. However in no AIMMS 3 version the statement


```
unit_par_a := [km] * unit_par_b ;
```

 will be allowed, because that mixes up the syntax for units with the syntax for expressions. The problems come from the following: internally AIMMS uses two symbol tables. One for identifiers and one for units. This makes it possible to use e.g. `s` as a unit as well as an index of a set. Furthermore, brackets can be used for both grouping of sub expres-

sions and indicating unit constants. When the compiler crosses such a bracket in an expression that consists of both an identifier and brackets, like `unit_par_a := [km] * unit_par_b` ; it does not know whether the bracket indicates the starting of unit constants, or the starting of a sub expression.

- AIMMS now supports runtime unit checking for the passing of arguments to procedures.
 - Unit consistency checks have been added for reading data from a file.
 - AIMMS does not require the use of free indices anymore when writing to indexed database tables without explicitly mentioning the identifiers to be written.
 - The `System` command has now been implemented in the `AimmsCmd` tool.
 - The robustness of case reading has been improved, after changing the domain of a set from a one-dimensional domain to a compound set.
-
- After a project export with one way encryption turned on, `.amb` files that were present in subfolders of the project folder were accidentally copied to the export folder.
 - **Auto Save** and **Project Backups** could go wrong while a **File Selection** dialog was on the screen.
 - The **Agent Role Comment** was not saved properly after making changes in the **Webservice Properties** dialog.
 - After adding/removing an index from a constraint or variable, a severe internal error could occur.
 - An error could occur when reopening the model explorer after a significant portion of that model had been deleted.
 - Editing an identifier in a **View** window, with the corresponding **Attribute** window of that identifier still open, resulted in a fatal application error.
 - When using a Network License, configuration files that were sent from the server to the AIMMS client could overwrite local configuration files. The files sent from the server are now copied to a special local folder named `NetworkCache`.
 - The deletion of a **Case Type** or **Data Category** was not properly saved in the project file. Upon reopening the project file the deleted item reappeared.
 - AIMMS now forbids copying of model explorer sub-trees containing source file references.
 - When calling procedures from within expressions, the profiler did not always give the correct results.
 - When using an environment variable reference in a value for the option `Backup_files`, the variable was not properly expanded. For example: `%Userprofile%\MyBackup`

Bugs...

- For a **2D Chart** with a set or index as the **X-axis domain**, changing the chart type to a **Polar** chart and back to the original chart type, replaced the original **X-axis** labels with numerical values. ...GUI
 - 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 showing the chart.
 - After pressing **Start** in a **Parametric Curve** object, all kinds of errors could appear.
 - When creating artificial identifiers to implement expressions typed in the GUI, there was a memory leak.
 - When showing a right-mouse menu from another library on a page while the menubar of that page had a navigation menu item, the items in the menubar were repeated several times.
 - Color changes made through the function `UserColorModify` were not propagated correctly to page objects that depended on it.
 - In the action list of a **Button** or a **User Menu** item, using a reference to one of the **File** menu commands: **Library Manager**, **Open - Model**, **New - Model** or **Backup - Restore - Project**, could result in a fatal application error. This is because all these commands may result in 'deleting' the current model, and thus implicitly the action list that is currently being executed. These specific menu commands are now filtered out, and executed as an extra command after the action list itself has finished its actions.
 - Closing the **Startup Page** could result in an error when selecting a menu command afterwards.
-
- Adding new identifiers to **Pivot Tables** that were already extended with artificial indices, has been improved. ...Pivot Table
 - In the **Pivot Table** it could occur that the scrollbars were not updated correctly after an external data change (from within a procedure).
 - In some situations, changing data in the **Pivot Table** for multiple cells simultaneously (block edit) did not work correctly.
 - When copying part of a **Pivot Table** with **including headers** selected, the header titles as well as the elements in the trees are now copied.
 - The **Pivot Table** was not able to display scalar values or identifiers for which all indices had been fixed (either explicitly or by moving them to the outer index area).
 - In the **Pivot Table**, the initial data was not retained when an invalid value was entered.
-
- In the **Library Manager**, replacing a library with another version (for example in another folder) did not work correctly. ...Libraries
 - In the **Library Manager**, selecting another model file for an existing library could result in file sharing errors later on.
 - Procedures with a module or library prefix did not work correctly in the

Data Manager Batch Run facility.

- In the **Library Manager**, selecting an ASCII .amb file in a Unicode AIMMS (or vice versa) resulted in a fatal application error.
- Mapping entries in database tables in a library module with explicit referencing to identifiers in a global scope (by using the :: prefix of the global scope identifier(s)), were sometimes interpreted wrongly as identifiers with the same name, but declared inside the library module.

- For non-AIMMS language attributes such as **Name**, the auto complete dialog will not appear anymore. ...Syntax Editor

- Solving a mathematical program twice with an explicit subset of constraints and a specified violation penalty could result in using an incorrect subset of constraints the second time. ...Solving
- The procedure `GMP::Solution::SendToModel` could pass an incorrect solution if multiple GMP's were used.
- The generation of a constraint did not work correctly when using the function `GMP::Row::Regenerate` in a FOR loop and the constraint definition contained one of the functions `ORD(running index)` or `VAL(running index)` or a running index that was compared with a quoted element or element parameter.
- AIMMS did not print the *major iterations* information if the `SNOPT` option `Print_Output_File` was set.
- A fatal application error could occur if the following sequence of GMP functions was executed:


```
GMP::Column::SetLowerBound(gmp1,col1,val1);
gmp2 := GMP::Instance::Copy(gmp1,"name");
GMP::Column::SetLowerBound(gmp2,col1,val2);
```

 The same could occur with the functions `GMP::Column::SetUpperBound`, `GMP::Row::SetRightHandSide` or `GMP::Row::SetLeftHandSide`.
- The \$ operator was sometimes interpreted wrongly, when used in a non-indexed constraint with a variable on the right hand side.

- AIMMS now also recompiles the property **stochastic** for parameters instead of only compiling that property at the initial compilation. ...Stochastic Programming
- The `MapVal` function was not supported when generating stochastic equivalents of parameter and constraint definitions.
- When the range of a stochastic variable references a stochastic parameter, the range of the `.Stochastic` suffix of the variable is now automatically adapted to use the `.Stochastic` suffix of the parameter. For example: consider a stochastic variable `X(i,t)` with range `[0, upb(i)]` where `upb(i)` is a stochastic parameter. Then the range of `X.Stochastic` will now be `[0, upb(i).Stochastic(IndexScenarios,i)]`.
- The wrong sign was used in the contribution of fixed variables to the right hand side of stochastic programs.

- if - then - else expressions were not properly handled for stochastic programming.
 - When a warning is issued based on the option setting `Warning_Stochastic_Programming_Scenario_Consistency`, the GMP was marked wrongfully as unsolvable. It is now only marked as unsolvable when this option is set to Error. The default of this option is Warning.
 - Combining terms in a constraint referencing the same stochastic variable with stochastic parameters and without stochastic parameters, was not properly handled.
 - For stochastic programming, the representative column may have lower bound equal to upper bound, but columns represented by that representative column may have lower bound significantly lower than their upper bound. Therefore, stochastic columns for which the lower bound equals the upper bound, now appear in the matrix and are not optimized away anymore.
 - An error occurred when generating a stochastic program that contains a constraint of the following form: $PD * X + PS * X + \dots \leq b$, where X is an ordinary variable, PD a deterministic parameter and PS a stochastic parameter.
 - An automatic optimization, consisting of a reordering of factors in a linear expression of the original deterministic formulation of a constraint, could confuse the subsequent construction of the stochastic definition of that constraint. This resulted in severe internal errors during the generation of the stochastic mathematical program.
 - Writing both the level and stochastic data of a variable to a file in a single composite table could lead to a severe internal error.
-
- When the mathematical program, objective, type, or direction in a solve statement are specified via defined parameters, it is now ensured that these defined parameters are up-to-date before executing the solve statement.
 - Previously, AIMMS did not accept expressions like `ord(first(Aset))` or `ord(last(Aset))`.
 - The value 0.0 can now be assigned to semi-continuous variables with a strictly positive lower bound. In addition, the matrix generator does not move semi-continuous variables initialized to 0.0 up to their lower bounds.
 - When the arguments of `MomentToTimeslot` have a reference outside the given calendar, AIMMS now consistently returns the empty element.
 - When commenting out the definition of a constraint, the old definition was still used until restarting the project. AIMMS now issues an error message when checking the constraint.
 - The `write where` suffix clause did not handle the `.Stochastic` suffix correctly.
 - The execution of scalar assignments with a condition, containing itera-

...Compiler and Execution

tive operators on the left hand side, did not always result in the correct assignment.

- If an indexed element parameter has a subset of a pre-declared root set as its range, and this indexed element parameter is used as argument of a function, AIMMS incorrectly issued a warning.
- A condition like `SomeIdentifier.Basic = 'NonBasic'` did not give the correct result.
- The **Run - Interrupt** shortcut key `Ctrl - Shift - S` did not work correctly when a dialog box was displayed during a run (for example: the dialog from the function `DialogProgress`). This also had the effect that `Ctrl - Shift - S` could not be used twice after the first interrupt dialog was cancelled.
- Adding an integer element outside the range of the set `Integers` to a nonempty subset of the set `Integers` could result in a fatal application error.
- Moving identifiers including at least one variable with non-empty definition attribute out of a declaration section and subsequently deleting that declaration section, could corrupt the internal data structures of AIMMS which could be followed by a fatal application error.
- A severe internal error could occur when using compound root sets declared in a local declaration of a procedure.
- When copying a procedure that included a local index, error messages regarding an undeclared local index could be generated.
- The functions `ArgMin` and `ArgMax` did not always give the correct result for string valued arguments.
- A fatal application error could occur in some cases in which parameters with a complex index domain, set definitions and procedures using these identifiers, were used.
- AIMMS accepted a single element as initial data for an indexed set, resulting in a fatal application error at compile time. A single element is no longer accepted for an indexed set.
- A compiler error occurred when a one-dimensional set was used in the condition of a for loop or an assignment statement, without any reference to the running indices.
- After an import of a `.amb` file or after the addition of a library, the compiler now properly adapts the pre-declared sets `AllDefinedSets`, `AllDefinedParameters`, `AllUpdatableIdentifiers`, `CurrentInputs`, `AllStochasticParameters` and `AllStochasticVariables`.
- The importing of `.aim` files which include modules, into sections, has been improved. In addition, the detection of name clashes has been improved by also detecting when a node in a `.aim` file has the same name as the section that is being imported into.

- When the `.Unit` suffix is used on the left hand side of an assignment, a proper error message is given instead of a severe internal error. *...Units of measurement*
 - When the unit in a unit attribute is prefixed with a constant and contains a unit parameter, the computation of the individual unit did not take that constant into account, resulting in a duplicate scaling to the solver.
 - In the unit attribute of a parameter, using an indexed unit parameter that has quoted elements in its index domain, AIMMS showed an error message. For example: `TheUnitParameter('Demand', j)`.
 - The expressions `Unit(-)` and `Unit([-])` now both result in the empty unit instead of an error message.
 - When the unit of a scalar parameter is a slice of an indexed unit parameter, rounding could result in incorrect values.
 - When the unit attribute of a calendar consists of a scalar parameter times a unit, the scaling was not always done correctly.
 - When reading input data from a file in which the units are specified at each individual element in a list, duplicate scaling could occur.
 - Using a parameter with an indexed defined unit parameter as its unit, could result in incorrect data.
 - During the run time analysis of units of measurement for constraints, severe internal errors could occur.
 - Using a sparse indexed identifier with a non-zero default and an indexed unit parameter, could lead to incorrect warnings.
 - When the unit attribute consists of an indexed unit parameter with all indices specified by quoted elements, AIMMS had difficulties in determining the uniqueness of the unit attribute, resulting in a different value for the unit attribute upon reopening the attribute window.
 - A severe internal error could occur when using literal units as function arguments.
 - The `.Unit` suffix can now be used in an argument value in calls to procedures that expect a unit valued argument.
 - The not operator now also accepts unit valued arguments. It already accepted numeric, element and string valued arguments. In all cases its result is numeric valued (always 0 or 1).
-
- In the database table mapping wizard the identifiers get their corresponding icon in front of them. This now also works for identifiers in a library. *...Databases*
 - Element parameters used in indexed database tables are now up to date before starting the `write to/read from table` statement.
-
- Under Linux, the file `message.log` was not emptied between two subsequent AIMMS sessions. *...Linux*
 - The Linux version of AIMMS did not properly start a project in End-User mode.
 - Some folders that are created under Linux did not get the correct access

flags.

- Reading/writing from Excel through ODBC sometimes resulted in an ODBC- ...*Windows Vista* error, stating that the table name could not be found. The table name was surrounded on both sides with a random ASCII-character.

AIMMS 3.7 build 106, release date 2007-08-14

In this AIMMS 3.7 Software Update, the following issues have been fixed or improved.

*Build 3.7.106
2007-08-14*

- Data pages displaying variables and constraints will automatically add suffices if the underlying data is refreshed by for example a solve. *GUI*
- In a **Scalar** object, a numerical value (e.g. '1') as the value for the **Hide If** property caused on error when reopening the page.
- In previous AIMMS 3.7 versions the GUI did not accept the **.Level** suffix for a parameter.
- In previous AIMMS 3.7 versions opening pages that used the AIMMS 2.20 page references (via the **.page.aliases** file) did not work.
- The **Argument Wizard** could cause problems in case the escape key had been used to cancel the creation of a new argument.
- In AIMMS 3.7, when in a **Selection** object on a page the property setting **Text Shown** was set to **Identifier Text**, the identifier name was displayed instead of the identifier text attribute.
- The behavior of the **Domain wizard** has been improved when selecting indices declared for a subset of the selected domain set or free indices as domain indices.
- In a table, having a **procedure upon change** that was changing other data in the same table, could result in a Fatal Application Error.
- In the properties dialog of a **Scalar** object, changing the **Element Description** to **Identifier Text** was not always applied correctly.
- The **Scalar** object accepted a value after complaining that the value violates the range.
- The stage attribute of a variable can now be modified via a wizard.
- Using more than 25 elements in a **2D Chart**, resulted in the default colors to become black after 25 normal colors. Now the **2D Chart** can handle up to 256 elements.
- The **Identifier Cardinalities** and **Math Program Inspector** menu commands did not function when they were copied to a user menu (in the **Menu Builder**).

- The **Pivot Table** can now also show the `.nonvar` suffix of variables.
- The **Pivot Table** was not able to show compound sets with dimension greater than one.
- A **Pivot Table** containing multiple identifiers (with different index domain) that shows the identifier index in the outer index area would not always display the headers in the row tree correctly.
- In a **Pivot Table** containing multiple identifiers (with different index domain) moving a header that was not shared among all identifiers could sometimes lead to a Fatal Application Error.
- When in a **Pivot Table** a subset of `AllIdentifiers` is used as the contents of the table, a Fatal Application Error could occur when this set was empty.
- When you have a **Pivot Table** of which the contents was determined by a subset of `AllIdentifiers` and the list of explicit identifiers also contained a reference to an identifier that had already been deleted, a Fatal Application Error could occur.
- In the **Pivot Table** properties dialog on the **Indices** tab, in the **Default** category the value that was used by AIMMS for the **Show All Elements** property was different than the one displayed.
- Moving the identifier node to the lowest level in the row (or column) tree of a pivot table could cause a Fatal Application Error. A page that already contained a pivot table with the identifier node in the lowest level, could not be opened.
- When changing tabs on the **Pivot Table** end user dialog, it could happen that AIMMS would cause a Fatal Application Error when the aggregators tab was selected again.
- Aggregator values in the **Pivot Table** were not always correctly updated after grid values had been manually changed.
- In the AIMMS Unicode version, **Pivot Table** state files were not always read in correctly.
- The **Pivot Table** did not respond correctly to the **Use (Shift-)Tab within Object** setting (on the **Misc** tab of the **Pivot Table** properties dialog).

Pivot Table

- **Commit Name Changes** now also applies name changes to library module `.amb` files.
- For identifiers that were referred inside their own library, the name completion unnecessarily prefixed these identifiers when used in a definition attribute.
- If a (library) project is opened in read-only mode (because the `.libprj` or `.prj` file is marked as read-only), the corresponding model file is treated as read-only as well.
- The **Selection via Identifier** of the **Linked Page** button actions did not work correctly in combination with **Libraries**.
- When using an identifier name quoted as an element in the definition of a variable that is declared in the same library as the quoted identifier,

Libraries

the quoted identifier got prefixed upon reopening the attributes of the variable.

- After setting the **Stochastic** property of parameters or variables, the syntax editor did not show the `.stochastic` property directly in the name completion. *Syntax Editor*
- Integer numbers are now colored similar to other elements when used inside `DATA{}`.

- **BARON** cannot be used for asynchronous execution and therefore AIMMS will now generate an error that `GMP::SolverSession::AsynchronousExecute` is not supported for the solver **BARON**. *Solving*
- Calling `GMP::SolverSession::Execute` after `GMP::Instance::Solve` could result in a Fatal Application Error.
- A Severe Internal Error or an incorrect solution could be generated if the procedure `GMP::Row::Generate` was called for a row that contained columns that were not part of the GMP.
- The value of `inf` assigned to the option `Solver.Workspace` was not handled correctly. AIMMS would give an unexpected error message if **XA** was used.
- In the last two AIMMS 3.7 versions, the generation of nonlinear constraints involving an AND-condition did not work correctly.

- After a forced recompile AIMMS will no longer initialize parameters with their initial data. Still identifiers with a large index domain and initial data or definition can use large amounts of memory. *Compiler and Execution*
- A Severe Internal Error could occur in a model that contains a module with source file, after recompilation and a commit name changes.
- A Severe Internal Error could occur when set valued expressions contained the `$` operator.
- During intensive use of distribution functions, the incorrect error **Input-value is a probability and thus should be in the range [0,1]** could occur.
- The `Min` function now also works for strings.
- `OptionGetString` now recognizes options with spaces in their names.

- Both files and conventions can be declared locally to a procedure and the convention attribute of a file may contain a reference to such a local convention. Saving and reopening the application could corrupt the convention attribute of such a local file because the convention name was prefixed by the procedure name, causing a compile error. *Name Changes*
- In some rare situation the **Commit Name Changes** functionality incorrectly reloaded source files causing a corrupt model.
- A Severe Internal Error could occur after renaming a variable that has just been copied and which contains a definition.

- An encrypted model could become corrupt when moving procedures to an unencrypted source file.
- The `.unit` suffix can now be used in unit casts. For example, the expression `(1)[MyParameter(i).unit]` is now allowed, where the unit cast `[MyParameter(i).unit]` is used.
- In some situations when using a unit of measurement, the conversion factor was not correctly evaluated and therefore the displayed values were not correct.
- A warning regarding unit consistency was given when one of the sub expressions of `+` was not a numerical expression. E.g. `MyString := "The value is " + TheValue ;` where `TheValue` is a numerical parameter that has a unit.
- Consider a procedure `Proc` with a local set `S` and index `i`, a local indexed parameter `P(i)`, and a local unit parameter `UP(i)`. The unit attribute of `P(i)` can be defined as `UP(i)`. Saving, closing and reopening the project will now show the unit attribute of `P(i)` as `UP(i)` instead of `UP(proc::i)`.

Units of measurement

- When executing a `DirectSQL` statement within a manually started transaction, AIMMS now does not execute an automatic rollback anymore when it fails. This gives the AIMMS developer maximum flexibility in using `DirectSQL` in combination with manual transactions. Note that you can use the return value (0 or 1) to see whether the execution was successful.
- Using a numerical column in a primary key in combination with writing in merge mode, sometimes yielded an unjust **duplicate key value** error from the underlying database.
- A Fatal Application Error could occur when an execution error appeared after a database read statement.
- Opening the mapping wizard for a database table/procedure/function in combination with an MS Access database and the "Microsoft OLE DB Provider for ODBC Driver" resulted in a fatal application error.

Databases

- The **X64** installation of AIMMS now has been extended with GIS support too.
- The **Math Program Inspector** did not work correctly using the **X64** version of AIMMS.

Windows X64

- Using **Web Services** under **Windows Vista** with User Account Control (UAC) active, does not work automatically since some files that are important for Web Services must be generated in the AIMMS bin-directory, which is located in the Program Files directory. You can work around this by either changing the rights of the AIMMS bin-directory, or by using AIMMS under the Administrator account, to generate the web service related files.

Windows Vista

- The icons in the **Library Manager** dialog had incorrect coloring under Windows Vista.
- The icons in the **Navigation** object had incorrect coloring under Windows Vista.
- In the AIMMS options, you can now use *environment variables* in the specification of the backup and log folders:
For example for the option **Project - Directories - Backup files** you can now specify %APPDATA%\MyProject\MyLogFolder. The value of APPDATA differs per user and is defined like:
On XP C:\Documents and Settings\John\Application Data
On Vista C:\Users\John\AppData\Roaming.
To see which *environment variables* are available, get a command prompt and type: set. The interesting ones that are usually defined on any windows system are: APPDATA, USERPROFILE, USERNAME.
- When another **Data Manager** file was opened while there was a page open that contains a **multiple case object**, and the CurrentCaseSelection was not empty, a Fatal Application Error could occur.
- Read-only modules could be displayed as sections in some rare situation.

Various

AIMMS 3.7 build 105, release date 2007-06-25

In this AIMMS 3.7 software update, the following issues have been fixed or improved.

*Build 3.7.105
2007-06-25*

- A link to **XPRESS 18** has been added to the **Complete** AIMMS installation.
- The following **BARON** options have been added: Node Selection, Probing Level Frequency, Probing Tree Start Level and Probing Tree End Level. More information can be found in the Help.
- A Stochastic Tree Library has been added to the Stochastic Programming example. With this library you can easily display your stochastic scenarios in a tree representation.
- The following GMP functions have been made more efficient, especially for quadratic constraints: GMP::Row::Generate, GMP::Coefficient::Set and GMP::QuadraticCoefficient::Set.
- The **Sentinel SuperPro (dongle) Drivers** have been removed from the AIMMS installation. They only need to be installed once if your license is bound to a dongle. They can now be downloaded from our website at http://www.aimms.com/aimms/downloads/aimms_37/download_37.html and can be installed separately.
- There was a limit of 32 arguments on multi-agent handler procedures. This limit has been increased to 256.

*New features**Improvements*

- The **Progress Window** now shows the number of threads **CPLEX** is using if **CPLEX** uses more than one thread for solving a MIP.
 - A unit consistency check has been added between a numeric parameter or variable and its bounds.
 - The width of the area to grab a vertical resize line in the **Pivot Table** has been increased.
- Bugs...*
- Starting AIMMS from a command line when a relative project path was specified, did not work.
 - The behavior of processes started from AIMMS (e.g. Acrobat Reader) has been adapted such that handles to AIMMS files are not 'inherited' anymore by these processes. Before, this was not the case which could result in AIMMS files staying locked while AIMMS was already closed.
 - Working with long file names near the Windows limit of 260 characters could cause a Fatal Application Error in AIMMS.
 - DataFileGetDescription was documented as a function in AIMMS, while the function did not exist.
 - References to the AIMMS tutorials in the AIMMS documentation index are removed, because **Acrobat Reader 8** would crash otherwise.
 - In case the **Interface** attribute is not empty and contains comment(s) before the interface attribute wizard is opened, a dialog will pop up informing the user that comments will get lost when the wizard is used to edit the attribute.
- ...GUI*
- When a saved data page contains an error or when the data page does not contain the identifier at hand, you now get the option to create a new data page for this identifier.
 - The default data page for indexed variables and constraints will now always show the **Level** suffix, even if all values are default. Note that to display the level of a constraint, you need to set the `Level` property of that constraint, or set the `Always_store_constraint_levels` option.
 - When the columns width of a column in the **Composite Table** was specified using a parameter, the width was not always set correctly.
 - Importing a bitmap file into the **Project User Files** could result in a corrupted bitmap.
 - When using `--ide-style-95` in the command line, and when using a user menu in which the shortcut key **Delete** has been set explicitly for the Edit - Delete menu command, pressing the **Delete** key in a **Table** or **Scalar** object would result in an error.
 - When deleting one or more entries in a **Table** (by pressing the **Delete** key), the entries are now set to their default values. This was not the case in all situations. For example, when deleting multiple cells, the value was explicitly set to '0' (even if the default value was not equal to '0').
 - In some situations nothing was shown in the dialog of `DialogGetElement-ByText`.

- In some situations the hours in a calendar of the **Gantt Chart** were incorrectly displayed as follows: ... - 20 - 21 - 23 - <empty> - 0 - 1 - ...
 - The curve in a **Curve** object was not always displayed (correctly) if the lower and upper bound of the Y-axis were both very large, but their difference relatively small. For example: lower 400,000,000 and upper 400,000,100.
 - When the right-mouse menu of a page object was displayed on the screen after which a different tab of a **Tabbed Page** object was selected (thus without selecting a command from the menu), a Fatal Application Error could occur.
 - Pressing F4 during a drag operation in **Page Edit** mode could result in a Fatal Application Error.
 - A Fatal Application Error could occur when using the **Arguments** wizard to delete all arguments.
 - The **Body Call** wizard has been improved for body call attributes that contain a tab character.
 - In AIMMS 3.7.104 the menu command **Project Save** explicitly asked to save any open page that had been edited.
 - In all previous AIMMS versions when closing the project, any open page that had been edited caused the "Page has changed" dialog to appear. When you now choose to save the project, these open pages will be saved automatically. When you close a page explicitly, you are still asked whether you want to save the page.
 - In the **Menu Builder** the command Edit - Export often resulted in a Fatal Application Error.
 - In the **Import** dialog in the **Menu Builder**, the checkbox **Use this option for all imported items** did not work properly. Furthermore, the option **Import as New** did not work.
-
- A **Pivot Table** that contains identifiers with suffices, possibly in combination with explicit suffices specified for the suffix index (on the **Indices** tab of the **Pivot Table** properties) was not always displayed correctly. ... *Pivot Table*
 - The **Pivot Table** could cause a Fatal Application Error when an error dialog was issued while selecting an element from a drop-down list in the outer area.
 - Changing the value of an outer index in the **Pivot Table** using the drop down list could lead to a Fatal Application Error.
 - The **Pivot Table** did some unnecessary updates when changing a value in the **Pivot Table** through a checkbox.
 - Specifying the **Save User State** menu command as an action behind a button or a menu command did not work.
 - When changing the font in the **Pivot Table**, the sample font area in the **Font** selection dialog is not filled initially with the currently selected font.

- After a case is loaded, non-predeclared subsets of `AllIdentifiers` are now adapted to reflect model edits since the case saving time. For example, in previous AIMMS versions, when you have deleted an identifier after saving the case, loading the case would add the identifier to the set again.
 - `CaseCreateDifferenceFile` did not take ordered domain sets into account which sometimes resulted in a Severe Internal Error.
 - Searching for the startup case could in some situations result in a Fatal Application Error.

... Cases
- The definition of element parameters used as arguments of identifiers inside the mapping attribute of **Database Tables** was not always up to date.
 - The **Excel functions** did also pass the inactive data of an identifier to a spreadsheet range.
 - When you have an Excel project that includes a module without any (Visual Basic) code, you could get the following error message when opening the wizard for the action `Run Excel Macro` in the **AIMMS Excel Add-In**: `Run Time Error '35': Sub or Function not defined.`
 - Writing huge amounts of data in bulk mode could lead to an out of memory error.

... Databases
- A Severe Internal Error could occur when filling in the `Violation Penalty` attribute and solving the mathematical program within the same session, or generating a mathematical program via `GMP::Instance::Generate`.
 - Indexed quadratic constraints, solved as part of a QCP problem, might have the contribution of the quadratic terms to the right-hand side inadvertently removed.
 - `GMP::Row::Generate` could generate incorrect coefficients for a quadratic objective constraint.
 - **MINOS** would give a Fatal Application Error if the amount of memory, set by the general solvers option `Solver Workspace`, was not available.

... Solving
- In some situations, recompiling a model that includes stochastic constraints resulted in a Severe Internal Error.
 - Stochastic data read in using the `read from file` statement is no longer ignored.
 - Consider a parameter `P` with both a definition and the property `Stochastic set`. The definition associated with the parameter `P.Stochastic` was not always up to date when being displayed on pages or being retrieved via the **AIMMS API**.

... Stochastic Programming

- Assignments to indexed identifiers with a domain condition, whereby the condition on the assignment contained a unit cast involving the operators / or * could confuse the compiler. For example, a parameter P with index domain (i)|PD(i) and unit \$/g, the following assignment contains a condition whereby the condition includes a unit cast that involves the operator /: $P(i\$Q(i)\langle\rangle 0[\$/g]) := 1 ;$.
 - The empty statement on a stochastic parameter will now both remove the deterministic data as well as the stochastic data. Furthermore, the compiler now allows you to empty suffices of identifiers separately.
 - A warning is now issued when %y is used in conjunction with %w instead of %Y in timeslot formats.
 - Using an expression in the range of a parameter resulted in a Fatal Application Error in some cases.
 - After compilation errors on procedure arguments, the error recovery has been improved.
 - A calendar was considered 'out of date' which reduced the performance drastically when updating the definitions directly depending on that calendar.
 - Reading sliced data into parameters with a non-atomic unit could lead to scaling problems in the data.
 - An error during the reading of a composite table from a file could confuse the reading of subsequent composite tables in the read statements that follow.

- When the Debugger was on a breakpoint, hovering the mouse over a math program, macro or other non-data identifier type, could lead to unexpected results like not displaying data of local identifiers. Also, hovering over a partly selected expression led to unexpected results.

- Identifiers that are declared public in a module, will now be better recognized/colored by the syntax editor. Similar for units.
 - In some situations incorrect syntax could result in a Fatal Application Error when the syntax editor was trying to recognize/color it.
 - The tooltip generation of functions or procedures that expect another function as an argument have been adapted to display as such.

- The **Add New Library** dialog now asks for confirmation if a name is entered that is already in use.
 - Within a **Library** and in, for example, the **Page Object Properties** dialog, the name completion (control-spacebar) did not work correctly for identifiers in the library itself.
 - When a (library) project is not changed, its (lib)prj file will now remain unchanged.
 - Conditions on breakpoints in a library did not handle the prefixes of the identifiers correctly.

... *Compiler*... *Model editor*... *Syntax editor*... *Libraries*

- Using the **Argument** wizard in a library did not always work correctly.
- In the model editor it was not possible to copy an item from a readonly library to (for example) the main model. It did work correctly using the Copy and Paste commands in the menu.
- It was not possible to copy a page from a readonly library.
- In the **Page Manager**, copying a page between libraries could result in an error, if the underlying tree of template pages could not be duplicated because of name clashes with existing items in the destination tree.
- In a View Window of the **Identifier Selector**, procedures within a library were not prefixed with the library prefix. Furthermore, the printed version of a View Window did not contain prefixes at all.
- In the **Project User Files** dialog, the button **Edit Text File** did not work properly for files in a library.
- The **Profile Listing** file did not contain information about libraries.
- The **Project Backup** could result in a Fatal Application Error after deleting libraries from your project
- In the **Pivot Table** and **ActiveX** objects, selecting a font from a library did not always work correctly.
- An error could occur when a set, which was used in a timeslot format, was declared within a module or library.
- When an external function and its call were spread over different libraries a Severe Internal Error could occur.
- Using artificial indices in a **Pivot Table** that contained identifiers for which some of the domain indices were declared within a library, did not work.

AIMMS 3.7 build 104.4834, release date 2007-05-07

In this AIMMS 3.7 software update, the following issue has been fixed.

- In AIMMS 3.7.104 there was an issue regarding the opening of child processes from within AIMMS (such as viewing PDF files). In specific situations, having such a child process still open while trying to save an .amb file, could result in the error message "Error reopening database: Error opening DB file (The file exists)".

Build
3.7.104.4834
2007-05-07

AIMMS 3.7 build 104, release date 2007-04-20

In this AIMMS 3.7 software update, the following issues have been fixed or improved.

Build 3.7.104
2007-04-20

- Two new functions are added to read and write cases from/to a separate file on disk: `CaseWriteToSingleFile` and `CaseReadFromSingleFile`. *New features*
- A new section called "Application to goal programming" is added to the AIMMS Language Reference in the chapter "Solving Mathematical Programs".
- There is a newer version of KNITRO, i.e. **KNITRO 5.1**.

- The option `Severe_Internal_Error_Contact` is added. The default is "AIMMS technical support via `Support@aimms.com`". This is the contact address mentioned in the severe internal error messages and fatal application error dialogs. *Improvements*
- The function `TestDataSource` is extended with the optional arguments `interactive` and `timeout`. The range of `interactive` is $\{ 0, 1 \}$ and the default is 1. The range of `timeout` is $[0, \text{inf})$ and the default is 30. The unit is seconds. When `interactive` is 1 and passwords are needed, a popup is presented to enter the password. When `interactive` is 0 while a password is needed, the `TestDataSource` returns immediately with the value 0. The function `TestDataSource` waits at most `timeout` seconds before returning.
- The **Pivot Table** has been extended with an extra property that allows the developer to specify the **Maximum Number of Entries** in the row and column tree. If the data that is contained in the Pivot Table is high-dimensional and the developer has not restricted the end-user access to the indices tab and/or the possibility to move indices around, the end-user could create a pivot table of unmanageable size (e.g. by making all indices dense and moving them to the same tree). The new property will not show the pivot table but show a message instead, when the number of rows in the row tree (or the number of columns in the column tree) will exceed the specified maximum.
- The Flow Shop example has been extended to illustrate how to use a MIP start solution.

- The name change module for changing the name of defined variables has been improved. *Bugs...*
- One new time zone has been added to the Linux version to match the latest Windows time zone information.
- In AIMMS 3.7.103 personal nodelocks were not automatically updated which resulted in the need to manually update the nodelock license after the grace period of 60 days.
- The Index of All Examples did not work properly when it was opened on x64 machines.
- The AIMMS 3.7 installer did not overwrite the examples by the ones in the software update installation package when the existing examples were modified.
- Localized Text in combination with a read-only project or model did not work correctly.

- On Windows Vista, calling `OpenDocument` with forward slashes in the file name did not work. Now, if the function fails, it automatically attempts a second call in which all forward slashes are replaced by backslashes.
- When Acrobat Reader 8 was opened from within AIMMS, closing it would lead to a crash in Acrobat Reader.

- When a saved data page contains an error, and it does not contain the identifier anymore, you now get the option to create a new data page for it. ... GUI
- In the property tree of a page object, the subproperties folder now automatically expands when the folder property is set.
- The **Element Text** tab of the page object properties dialog, sometimes displayed the same index twice.
- The 'default background' color in the **User Colors**, will always use Windows' background color, unless it is specifically set to a different color.
- A resizable page could sometimes scroll unnecessarily when the focus switched between objects.
- The keys **Delete** and **Backspace** can now be used as a shortcut in a user-defined menu. However, these shortcuts will **not** work when the focus is in a window that has its own interpretation of these keys. For example when a simple edit field has focus, then pressing Delete will delete a character and it will thus not invoke the linked menu command.
- Changes made to a copy of a default menubar or toolbar in a library were not always correctly taken into account when the actual menu was constructed. In some situations this could even lead to a fatal application error.
- Deleting entries in a menubar that is being used by one of the open pages, could result in an error.
- In tables with multiple identifiers in the rows, scrolling vertically did not always show the correct values.
- In AIMMS 3.7 the **Navigation** object option **Include hidden page as disabled** did not work.
- When deleting one or more identifiers from the Y- or Y2-axis of a **2D Chart**, the result was not immediately visible after clicking on 'Apply' or 'OK' in the **2D Chart** properties dialog.
- In the **2D Chart**, specifying a one-dimensional value label string for a two dimensional Y-axis identifier, with the index of the value label string matching the second index of the Y-axis identifier, resulted in a fatal application error.
- Sometimes, the click label of a **2D Pie Chart** was not completely readable, when clicking on a slice close to the edge of the pie object. Now, the click label is attached to the exact position where the user clicked.
- In the **Date Time Picker** object, when selecting a date/time without time values (or where hours, minutes and seconds are 0), AIMMS will check whether the current value of the string parameter contains time values. If not, and it only has date info, the new value of the string will also

contain date info only.

- The **Indexed Page** object did not respond correctly to a change in the number of columns or rows.
- In specific situations, the wizards for **Range**, **Index** and **Arguments** could result in a fatal application error.
- In recent AIMMS 3.6 and AIMMS 3.7 versions the description of the **integer** range in the **Range** wizard was incorrectly given as $\{-inf..inf\}$. This description has been changed back to $\{0..inf\}$, which has always been AIMMS' interpretation of the integer range. Furthermore the range nonnegative integer has been removed.

- The copying and pasting of blocks in the Pivot Table has been improved.
- In a Pivot Table, if the current combination of outer indices results in an empty display, then the table will try to initialize the outer indices to a combination for which there is data. This only happens after a data change that originates from outside the Pivot Table itself, and only those indices that are not linked to an element parameter are modified.
- The **Only show row differences** and **Only show column differences** options in the Pivot Table did not filter out identifiers for which all values were identical.
- Using the multiple case view when artificial indices have been specified in the Pivot Table gave unexpected results. This combination is no longer allowed.
- When the 'General' Pivot Table property **Save Layout/State by Developer** has been set to 'No' (and when no user state is present yet), the developer state file was not read in during initialization. It was also not read in when the state file was marked as read-only.

...Pivot Table

- The warning message regarding inconsistent structure information in cases is not unnecessarily issued anymore.
- The recovery of corrupt case reading has been improved.
- Cases with stochastic data can now be read in, even when the associated parameters are no longer stochastic.
- For a case type, the procedures **Prior to Load** and **Upon Load** were also called when the case was loaded from within a procedure using `CaseLoad-Current`. This was not in sync with the documentation and resulted in an error dialog.
- The function `CaseCreateDifferenceFile` could not properly handle Horizons.

...Cases

- Trying to insert a Microsoft Access query with arguments resulted in error messages stating that the arguments should not be enclosed in square brackets. Also, executing Access queries with the `UseResultSet` property checked, did not work.
- Writing to a MySQL database table in **replace rows mode** through an

...Databases

ODBC link, sometimes caused an invalid 'duplicate key' error message.

- When reading/writing an Excel workbook through ODBC, every sheet in the workbook is implicitly considered to be a "database table". The name of such a table is formed by appending a '\$' character to the sheet name. Those tables were not shown in the table wizard(s). Also, executing the function TestDatabaseTable with such a table as the second argument incorrectly yielded 0 instead of 1.
- When using a FoxPro/DBASE-database link in AIMMS, read or write commands sometimes yielded a severe internal error.

- There is a new xav14.dll file which fixes some rare bugs in **XA 14**.
- The program status FreeForFutureUse in the set AllSolutionStates has been replaced by InfeasibleOrUnbounded. Right now, **CPLEX** is the only solver that can return the program status InfeasibleOrUnbounded.
- AIMMS did not retrieve the level values from **BARON** when the model was infeasible.
- When some options are set for a specific solver and the solver is deleted from the solver configuration, these stored option settings will now be read in once that solver is added again.
- AIMMS could give a solver initialization error during startup when a solver version was not available.
- The GMP version of AOA could fail on models that have a fixed variable in a quadratic constraint.
- Repeated calls to GMP::Row::Generate and GMP::Instance::Solve could result in a severe internal error.
- The routines GMP::Instance::Delete and GMP::Instance::DeleteSolverSession did not free all memory that was allocated when creating the GMP or solver session.
- A fatal application error could occur after interrupting the solve for finding the substructure causing infeasibility in the **Math Program Inspector**.

... Solving

- Doing the action **Irreducible Inconsistent System (IIS) or Substructure Causing Infeasibility** from the **Math Program Inspector** for a stochastic model resulted in a fatal application error. For AIMMS 3.7 this action is no longer allowed for stochastic models.
- For a parameter P with a definition and the stochastic property set, AIMMS will now automatically derive a corresponding definition for the parameter P.Stochastic. The option DeriveDefinitions.Stochastic.Parameters in the backward compatibility category, controls this new behaviour. Settings this option to 'Off' will not derive a definition for a .stochastic suffix parameter.
- When using a display statement in which indices were not mentioned, to write the stochastic data in a composite table form to a file, a severe internal error could occur.
- AIMMS now allows you to display the units of stochastic data in page

... Stochastic Programming

objects.

- It is now ensured by the compiler that stochastic variables have their **Stage** attribute filled in. Furthermore, the **Stage** attribute can now be selected in the properties of a node in the **Identifier Selector** tool.
- The combination of stochastic programming and mathematical programs with violation penalties resulted in a severe internal error. For AIMMS 3.7 this combination is no longer allowed.
- AIMMS now issues an error message when the objective variable in a stochastic program is a stochastic variable (it should be an ordinary variable), and also when the .stochastic suffix on variables are used inside constraint formulations.
- The option `Warning.Stochastic.Programming.Scenario.Consistency` is added to the optimization warning category. The default is 'Warning'. When this option is not set to 'Off', a consistency check on the scenarios in a stochastic program is executed: If a group of rows differs only in their scenario index then the coefficient of each column should be equal to that in the row corresponding to the representative scenario.

- An expression like `2**3` could result in a severe internal error.
- When a procedure was called inside an expression and an output argument of that procedure is used inside the same expression, this resulted in incorrect behaviour.
- Changing the unit of a calendar could incorrectly result in an error message.
- AIMMS used to evaluate a definition directly after modification which could take a considerable amount of time in case of a dense definition. Now this behaviour is controlled by the new option `Evaluate.Definition.Directly_after_Modification`. The default is 'Off', which means that the definition is not evaluated after a modification in the definition, but when the data is requested.
- When a set and its domain set (specified in the **Subset of** attribute) were both declared within a module, the compiler gave an error message when the order of declaration was not the order in which the sets were used.

... *Compiler*

- The backup setting **Create an AIM File at Project Close** only created the .aim file when a change to the model was made. Now, AIMMS also checks whether the .aim file already exists: if it does not exist, or if it is older than the corresponding .amb file (and it is not read-only), the .aim file will be created. Furthermore, the .aim files of the library projects are now written as well.
- The model editor does not allow moving sections that are stored in read-only source files to another name space anymore, because the necessary name changes cannot be applied.
- When on a breakpoint in a procedure, the green arrow buttons to walk through the data pages did not work for local identifiers.

... *Model editor*

- When editing a project that was originally created with an AIMMS version earlier than 3.7, the corresponding .amb file could grow very large, because old pieces of model text were not always deleted. Furthermore, every save triggered an internal reorganization of the .amb file, which could lead to unnecessarily long save times.
 - The handling of index domain conditions and data modifiability of parameters that have a comment only definition attribute is improved.
- ...
- The content of the comment attribute could be obscured by its scrollbar. *... Syntax editor*
 - In some situations, the coloring of multi-line comments was ended before the actual 'end of multi comment line' token (*//) was encountered.
 - In certain scenarios the syntax editor could enter an internal infinite loop.
 - The index-domain name completion did not work for **arcs** and **nodes**.
- ...
- When during startup a Library could not be opened, pressing Cancel in the **Library Manager** dialog resulted in a fatal application error. *... Libraries*
 - The encryption of library modules has been improved.
 - The prefix of a Library could not be the same as the name of a local identifier or argument of a procedure.
 - Identifiers inside **Modules** inside Libraries were not handled correctly in (Data) Pages.
 - The command **Show Data** for an element parameter or index that was declared as an attribute of a Set in a Library, did not work correctly.
 - In the **Template Manager**, when the original template of a 'cross-library duplicate' was removed while this template was duplicated in multiple libraries, these libraries could incorrectly refer to each other. This caused a fatal application error when reopening the duplicated templates.
 - AIMMS incorrectly indicated that a project with new libraries was modified if you would try to close it directly after saving it.

AIMMS 3.7 build 103, release date 2007-02-15

In this AIMMS 3.7 bugfix release, the following issues have been fixed or improved. The issues are itemized in no particular order.

Build 3.7.103

2007-02-15

- Several suffices regarding the statistics for a mathematical program are added. These suffices contain more information on the number of variables and constraints of a certain type. For these statistics, also new GMP functions are added to retrieve the statistics. These functions can be found in the GMP library in the AIMMS Function Reference, at GMP::Instance Procedures and Functions.
- The intrinsic functions **Character** and **CharacterNumber** have been added to the library of AIMMS predeclared functions.

New features

- AIMMS now supports loading a .aim file into a section.
- The performance of a sequence of calls to `GMP::Row::Generate` is improved.
- A filter has been installed on the **Messages/Errors Window**, only allowing unique error messages to appear.

Improvements

- When the end date of a calendar is modified from within the model editor, elements that are no longer in the calendar are now removed from subsets of the changed calendar.
- The updatability of the .nonvar suffix is now determined by the value of the option `Variable_Bounds_Are_Updatable`.
- Name changes were not applied correctly to procedure bodies that contain quoted strings with quotes in it (by using escaped characters: a character preceded by `\` and thus taken literally). E.g. *"This is a \"quoted\" string"*.
- AIMMS now issues a warning when a unit overwrite is used, on an expression that already has a unit commensurate with the one used to overwrite. This warning can be turned off by setting the new option `Warning_Unnecessary_Unit_Overwrite` to 'Off'.
- The `PageSetCursor` function did not work correctly for **Pivot Tables** in which either the row or column tree did not contain any indices (e.g. default **Pivot Tables** displaying a one-dimensional identifier).
- The pasting of blocks in the **Pivot Table** has been improved.
- Name completion did not work correctly after a name change in a prefix.
- The name completion of the index domain for parameters will now add prefixes to the index domain when necessary.
- The message that appeared when an identifier move was not allowed, is now clearer.
- The functions `first` and `last` applied on the predeclared set `Integers` did not give the correct result.
- AIMMS now issues a warning message when indices or element parameters, declared as part of a set declaration, are present in the interface attribute of a library module. This warning can be turned off by setting the option `Warning_Identifier_Cannot_Be_Made_Public` to 'Off'.
- The **Index of All Examples** is adapted such that it also works properly when it is opened in x64 machines.
- The euro symbol is now accepted by **AIMMS Unicode** as it was already done by the **AIMMS ASCII**. In addition, both ASCII and Unicode version now accept the pound sign as currency symbol. Furthermore, **AIMMS Unicode** accepts all currency symbols as defined by the Unicode organization.
- The write statement, executed by the AIMMS Unicode executable, can now be redirected to an ASCII file using the device attribute of a file. Simply fill in: `DISK(ASCII)`.
- A compiler restriction has been relaxed. This can be illustrated by the

Bugs

following example: Let ep be an element parameter with range $S1$. Consider a parameter $A(i2)$ with range $i2$ is $S2$. AIMMS used to issue an error message on the expression $A(ep)$ when $S1$ was not a subset of $S2$. AIMMS now accepts it if $S1$ and $S2$ have the same root set.

- Empty calendars can now be generated by leaving the value of the string parameter referenced in the begin date or end date attributes empty.
- AIMMS now issues a warning when during execution, the range of a for loop index changes. This warning can be turned off by setting the option `Warning.Cardinality.Change.Range.For.Loop.Index` to 'Off'. See the Help file for more details.
- When the same tags were used by two significantly different compound sets and when these tags were removed in one of those sets, a severe internal error could occur.
- In the **Pivot Table**, numerical aggregators over cells that contain checkboxes are now evaluated as if they contain a value of 1 for a checked box and 0 for an unchecked box.
- An indexed set is now also displayed in a **Pivot Table** using checkboxes (instead of "x"-es), similar as how a single set and how a binary parameter or variable are displayed.
- When in a web service an AIMMS worker agent sends a message that contains output arguments to the AIMMS dispatcher agent, and both agents are logged on as 'xmlagent', the 'Nested XML' value of the called web service was ignored and assumed to be true. This could yield in an 'elem expected' error message in the worker agent.
- An "unexpected end of file" error could occur when reading .aim files containing duplicate section names declared in different namespaces.
- The AIMMS compiler now accepts macros to be used independent of the declaration order.
- Dragging objects and at the same time scrolling using the mouse scroll wheel, resulted in strange drawings. This combination is no longer possible.
- There was an inconsistency between wizards and text editor that caused special characters to disappear.
- The administration of public and protected symbols of a moved module was not always correct.
- After moving a template page between libraries, the prefixed names in the tree were not always displayed correctly.
- The automatic outlining of code-pieces (i.e. collapsing of block nodes) was triggered too often (e.g. after changing syntax-editor settings or re-compilation). Now block nodes only collapse automatically upon opening an attribute form.
- Importing sections into a newly created project could lead to inadvertent name changes whereby the original identifiers were prefixed with "NameAltered_".
- AIMMS only checked the feasibility of the rows for a single scenario during the generation of a stochastic mathematical program. It now checks

it for all scenarios.

- It wasn't possible to specify a one-dimensional string parameter for the property **legend entry** in a **2D-pie chart**, when showing a one-dimensional parameter.
- Stochastic data is now updatable in the GUI.
- In case `SubstituteStochasticVariables` was used as the generation mode in the function `GMP::Instance::Generate StochasticProgram`, the contribution to the objective of each scenario was calculated incorrectly.
- The **Delete** button in the argument wizard of a procedure, now asks whether you want to remove the argument from the local declaration section.
- For stochastic programming, when using the `first/last` operator in the index domain of a variable, a severe internal error could occur.
- After installing the Multi Agent module through the menu, the **Select community file**-menu item remained greyed out.
- In the pivot table the toggling of checkboxes related to scalar identifiers and sets did not always work as expected.
- When using `cleandependents` for local indexed parameters initialized with a scalar value, a severe internal error could occur.
- AIMMS did not accept a variable definition that only consists of `/* */` type comments.
- When a saved **Pivot Table** encountered an identifier for which the domain had changed, you did not get the option to **Browse** for the correct identifier.
- Adding **XPRESS 17** to the solver configuration could fail when an old version of **XPRESS** was installed on that machine.
- Deleting all existing agent roles in the **Agent Roles** tab of the **multi agent community setup** dialog, and then navigating to the **Webservice Properties** tab, resulted in a crash of AIMMS.
- The AIMMS 3.7 **License Manager** did not show the list of connected clients at all.
- The right hand side of stochastic rows was not scaled according to the unit of the constraint.
- Numbers with an explicit unit cast were converted to atomic units twice inside stochastic constraints, resulting in incorrect scaling of matrix coefficients.
- When displaying stochastic constraints in the **Math Program Inspector**, their constraint type was always shown as equality constraint.
- When determining the **Substructure Causing Unboundedness** for badly scaled models, the **Math Program Inspector** could sometimes return with a message saying the model itself was infeasible (instead of unbounded).
- Logging on external clients to an AIMMS Web Service was only allowed for stateful web services.
- If GMP routines were used to solve a MIP model, AIMMS could pass a basis to **CPLEX** even if the **CPLEX** option `MIP.Basis` was switched off. This could

result in a drastic decrease in performance because CPLEX would then skip the preprocessing step.

- Displaying the attributes of an element parameter with range compound set, whereby the compound set did not have tags, could cause the syntax highlighting module to fail.
- In some situations, data tooltips did not work correctly in combination with compound sets.
- Resolved an issue that caused incorrect syntax highlighting when comments that include brackets were placed inside bracket-scopes, e.g. `p(/* q(i) */ i, j)`
- In the **Pivot Table**, sorting string values did not always work correctly.
- The function `GMP::Instance::CreateDual` could generate an incorrect dual model if the primal model contained variables that were frozen to 0.
- The **Date/Timepicker** and **Calendar** object can now handle string parameters with a date-only format: `"%c%y-%m-%d"`.
- The intrinsic function `ListingFileCopy` has a new optional argument `overwrite`. When `overwrite` is non-zero, `ListingFileCopy` will overwrite an existing file. When `overwrite` is zero `ListingFileCopy` will not overwrite an existing file. The default of `overwrite` is 1.
- When sorting a column in a **Composite Table** the specified **Element Text** for an index was not taken into account. Now it uses a string comparison when the **element text** is a string parameter, or, if it is an element parameter, it uses the ordering of the corresponding range set.
- In the **Find** dialog, when you enter the name of a user function, say `MyFunc`, and then press the **Declarations** button, AIMMS presented a menu with two options: `MyFunc` and `MyFunc::MyFunc`. The second option was incorrect.
- Emptying the mapping attribute of a database table using the wizard did not work.
- AIMMS now copies the solution statistics of a solved generated mathematical program to the solution statistics of its parent mathematical program, even when the generated mathematical program has another name than its parent mathematical program.
- It was not possible to use an indexed set to define the set of **Implicit Identifiers** in the **Pivot Table**.
- Opening the **Math Program Inspector** from within the **Model Explorer** (using the right-mouse **Open With** command) resulted in a fatal error.
- The **Apply Changes** dialog box that appears when you close an attribute window in the model editor, now responds better to Alt-key strokes.
- Using the intrinsic function `CaseSave` and parameters without definition but with an index domain condition that is not a single identifier but a more complicated expression, did not always work correctly.
- A **Text** object on a print page with style **Spread over pages** always generated one extra empty page if at least two pages were needed.
- Changes of stochastic data identifiers in one page object did not always update that data in another page object.

- In the **Pivot Table** it was not always possible to change the value of binary identifiers with a default value of 1 (instead of 0).
- In AIMMS 3.7 the Round function did not work correctly when the first argument had a non-atomic unit (s) and the second argument was 0.
- Unit related information in the **Pivot Table** was not always updated correctly after a change in CurrentConvention.
- AIMMS now issues an error message when a unit parameter is used in the per unit attribute of a convention.
- In previous AIMMS versions it could occur that the displayed information of an encrypted model became corrupt (without the model file itself becoming corrupt). This situation could occur when the import of a non-encrypted model file was cancelled either by user or by aimms.
- Constraints containing the log10() function were passed incorrectly to **BARON**.
- The set CurrentAutoUpdatedDefinitions was stored erroneously in case files.
- The **Messages/Errors Window** now uses the same end-user menu and toolbar as are specified for the **Message Window**.
- There were some problems with the (wizard of the) Derivative call attribute of External Functions.
- The **Network License Manager** could show an incorrect status "Hardware key does not match with license file", but it continued to accept connections without problems.
- The contribution to the objective of each scenario could be calculated incorrectly in case the objective variable was also used in a constraint.
- When solving a model with **BARON**, the objective function could have an unexpected value. This was caused by AIMMS passing an incorrect objective function to **BARON**.
- In the Data Manager, a fatal error could occur if you tried to delete nodes just after a copy and paste operation.
- Changing a Case Type in a project that was originally created in an AIMMS version prior to 3.7, could cause the references to Data Categories to get lost.
- Trying to insert a Microsoft Access query with arguments resulted in error messages stating that the arguments shouldn't be enclosed in square brackets.
- When unit casting was used inside the then or else expression of an if-then-else statement, a severe internal error could occur.
- In the **Table** object, adding the following two identifiers did not produce the expected table:


```
A( <split> t )
B( i, j, k, <split> t )
```

 If the second identifier had more than 2 indices before the row/column split, the **Table** created a new set of columns for t.
- The function VarLicenseExpirationDate did not give the correct result if the var license did not have an expiration date. It returned "1969-09-11"

instead of "No expiration date".

- When using indexed unit parameters with a quantity, unit inconsistency warnings did not always refer to the correct index.
- In previous AIMMS versions the CPLEX LP + Barrier license did not allow all the functionalities that it should allow.

AIMMS 3.7 build 102, release date 2006-12-14

In this AIMMS 3.7 software update, the following issues have been fixed or improved. The issues are itemized in no particular order.

*Build 3.7.102
2006-12-14*

- There is a new version of **XPRESS**, i.e., **XPRESS 17**. It is no longer needed to turn off the presolver if GMP routines or AOA are used in combination with **XPRESS**.
- AIMMS web services has been enhanced with an automatic dispatch mechanism. It is now possible to assign more than 1 web service agent to a web service through the web service properties dialog box. The AIMMS HTTP Web Service Listener automatically assigns any incoming request to an available worker agent. If none is available, it automatically waits for one to become available (for as long as the timeout of the request allows it).

New features

- The database interface is now more robust in situations where a network connection has temporarily been lost. Furthermore, the functions **TestDatasource**, **TestDatabaseTable** and **TestDatabaseColumn** now first try to re-establish a database connection in case it has been lost. If they succeed, they return the value 1 as expected.
- The **2D Chart Object** now offers a property Color Scheme, with which you can specify a scheme to use for coloring the charts, analogue to the color scheme found in the **Curve** and **Bar Chart**.
- The complement attribute now also supports scalar element parameters with range **AllConstraints**. At runtime it is verified whether the constraint referenced via such an element parameter has indeed the same dimension and domain.
- Specifying contour values in a **3D Chart Object** involved the declaration of two identifiers in your model. This has been simplified by introducing the choice for an automatic contour values mode. The old mode is still available.
- A new option `Duplicate.Row.Update.Protection` replaces the option `Warning.Database.Non.Official.Key.In.Mapping` option to influence the compiler behavior when a write statement is encountered that might modify more rows in the database than expected.
- AIMMS now accepts macro argument names that are also declared as identifiers of another type.

Improvements

- When typing an identifier name in the **Find** dialog, the **Declaration** button will now also become active when the identifier is declared within a library, module or as the local identifier of a procedure or function. In case the identifier name occurs more than once a popup selection menu will occur when clicking the **Declaration** button.
- `GMP::Instance::SetMathematicalProgrammingType` can now be used to change the model type from MIP to LP.

- An incorrect inconsistency error could occur when loading a case. *Bugs*
- In AIMMS 3.7.101.2473, when running an existing project, making changes in a case type or data category could result in the case type and data category being removed when reopening the project.
- A corrupt case error could occur when reading a case containing compound set information.
- Reopening a project made in AIMMS 3.7.101.2473 could result in a recurring incorrect compile error: Error reading Case Type All Identifiers. All Identifiers Could not be found.
- In AIMMS 3.7 in some situations loading a case could result in a severe internal error.
- In the 64 bits version the dialogs for Case Load and Case Save As caused a fatal error.
- Working with data management in AIMMS 3.7 in a project made with a previous AIMMS version without making any changes in the Data Management Setup in AIMMS 3.7 could result in the loss of datasets.
- It is possible that opening the model explorer in AIMMS 3.7 fails due to insufficient .NET access and execution rights. In the previous version this resulted in a crash, in this version the old editor will be used if this situation occurs. Details about this problem can be found in our [Knowledge Base](#).
- AIMMS failed to open the Help file when internet explorer 7 has been installed. In addition when clicking the question mark button on the properties tab of the page objects, the help will be opened on the topic of the active tab.
- When dragging nodes from the **Model Explorer** on to the text in an attribute window, AIMMS will now insert the name of the identifier including the proper prefix.
- It could occur that after changing the range of a stochastic variable the range was not communicated correctly to the solver during a re-solve.
- The declaration sections in the **Predeclared Identifiers** library can now be viewed as a flat model, just like other model sections.
- AIMMS now allows modification of coefficients of a stochastic model.
- `GMP::Instance::Copy` did not work correctly for stochastic programs.
- In the **Pivot Table** default values other than 0.0, the empty element or empty string were not always displayed correctly when the setting Show Defaults had been switched on.

- When adding a library to a project AIMMS now ensures that the prefix is not an existing other identifier in order to avoid name clashes.
- Switching from **Resize Try Mode** directly to **Edit Mode** did not always redraw the page correctly.
- The function `ExitAimms` did not always end the AIMMS process correctly. Sometimes AIMMS seemed to be exited, but the Task Manager still showed a process for it.
- In the **Menu Builder** when moving duplicates between libraries, the duplicate link was not always retained.
- Incorrect compile errors could occur after adding an existing library to a model.
- AIMMS now also sorts root sets when the data of inactive cases is cached inside the memory of AIMMS.
- Before, when sorting rows or columns in a **Pivot Table**, both string and element parameters were sorted using a case-sensitive string comparison. Now, a case-insensitive comparison is used to sort string parameters and the set order is used to sort element parameters.
- Sometimes, the state of the **Expose as an AIMMS webservice** checkbox was not remembered correctly when changing the state of the **Generate WSDL for this role** checkbox in the web service properties dialog box. Furthermore, after switching between roles, this checkbox' state was not correctly remembered.
- When clicking on **Generate files** in the web service properties dialog box, `.AimmsWSConfig.xml` files were generated for all roles, i.o. only for the currently selected role.
- The **View - Font** menu in a Text Editor window did not work for the new editor.
- To better support Unicode characters, the font in the **Errors/Warnings** window has been changed.
- AIMMS does not evaluate the parameters anymore that represent the index domain conditions of variables and constraints just before a case save.
- When closing a page from within a procedure called by an object inside a **Tabbed** or **Indexed Page** object of that same page, a fatal error could occur.
- The menu command **Setup Localization** resulted in an incorrect warning message in the **Message Window**.
- In the **Project User Files** dialog, editing text files via the **Edit Text** button resulted in an error.
- In the **Pivot Table**, values for suffices of binary identifiers would sometimes be incorrectly displayed as check boxes.
- The location of the click labels in a **2D Stacking Bar Chart** was not always the most logical.
- Binary identifiers with a nonzero default value would not always be shown correctly when displayed in a **Pivot Table**.
- The saved index order was not always restored correctly in a **Pivot Table**

that contained multiple identifiers as well as artificial indices.

- Changing the column width in a **Pivot Table** for which the **Include entire header layout** setting has been specified did not set the page edited flag, causing it not always to be saved when it should have.
- The **Procedure Upon Doubleclick** was not called for the **Pivot Table**.
- A severe internal error could occur during the evaluation of an indexed set after its index domain was changed.
- Messages generated by a **GIS Network Object** did not correct the time stamp of the messages for daylight-saving-time.
- Localization was not fully supported by the **ActiveX Objects**.
- Localization was not implemented for the **Pivot Table**.
- The syntax editor is always operational in UTF16 (Unicode) character set; whenever the text content is passed to the AIMMS ASCII compiler a conversion takes place. The euro symbol is not officially part of the ASCII table and therefore this conversion failed (returning an empty string). The text content is now explicitly scanned for the euro symbol in case of conversion failure and adapted accordingly.
- When switching from an **Auto Hide** docking window (for example the **Errors/Warnings** window) to an attribute window, the menu bar was not always updated correctly.
- In a **Pivot Table**, if the identifier index is (outer and) fixed all remaining outer indices that are not used by the chosen identifier will not be visible.
- AIMMS incorrectly assumed that the user wanted to perform a name completion using unit identifiers inside the range attribute when using inclusive interval brackets.
- In a multi-agent application where all agent projects are already connected, adding new messages and sending them without first restarting the projects could cause a fatal error.
- Adding and removing var licenses in a library could result in corrupting the library model.
- In the previous AIMMS 3.7 versions the **Data Manager** file was not compatible with earlier AIMMS versions. In the current version the **Data Manager** file is compatible until you use libraries. For the use of libraries it is necessary to change the structure of the data management file causing an incompatibility with older versions.
- In the **Pivot Table** artificial indices in combination with multiple cases did not produce the correct result.
- Not all identifiers in a **Pivot Table** did show their associated unit initially. The unit was displayed only after the data was displayed at least once.
- In a **Pivot Table** with an outer index for the identifiers, selecting an identifier using the outer index did not work correctly if the identifiers had units.
- When using artificial indices in a **Pivot Table** all default values of an identifier were not updatable.
- When creating a **Composite Table** with a set or index as the domain, the set or index was also automatically added as a data column.

- When using Artificial Indices in a Pivot Table values could disappear.
- AIMMS now detects when an identifier is referenced in its own index domain, which will result in a compile error.
- Transparent **Picture Objects** were not stretched correctly.
- A selected **String Parameter** in a **Text Object** was not displayed in the properties dialog.
- When adding a new license to the license configuration, this new license was not automatically used. Also, when installing more than 1 license in the License Configuration dialog, you get an extra dialog message explaining how AIMMS will select its active license.
- Having a page containing a **3D Chart**, for which x, y, and z-identifiers have been specified that did not contain any data, resulted in a fatal error.
- When there was no data at all to display in a **3D Chart** of type Surface, a completely empty chart was drawn, even without any axes.
- Clicking in a **3D Chart** of type Scatter, when no data is displayed at all, resulted in AIMMS disappearing.
- In the **3D Chart**, the x and y-axes did not always scale properly when a numeric parameter instead of an index was used as the axis identifier.
- Expressions in the **Menu Builder** disappeared when the menu structure was read before opening any page.
- When copying (pages with) **Pivot Tables**, layout information that was stored in the developer state file was lost for the copied object. The same was true when moving pages between (library) projects.
- Using dense indices in a **Pivot Table**, now works better in combination with artificial indices. For example, when showing dense months in combination with an artificial index quarters, you now only get the 3 months that are valid for the corresponding quarter, and not all 12 months. A similar adjustment has been made to the dense calculation of aggregators in a **Pivot Table**.
- With **FileCopy** you can now copy a single **Project User File** to the normal file system.
- The new editor did not perform well on very large (> 8 MB) files. In these cases the old editor will be used.
- In the **Project User Files** dialog, the **Edit Text File** button did not work in combination with the new editor. Now the old editor will be used.
- In the **Properties** dialog of a **Bar Chart** or **Curve Object**, the multiple case property was not initialized correctly.
- When importing an .aim source file in a section and immediately afterwards running a procedure did not result in the procedure actually being run.
- The behavior of a **Pivot Table** in which the focus cell is overlapped by a scroll bar would sometimes not be as expected (e.g. during scrolling or cell editing).
- After an empty statement the data in a multiple case selection was not always correct.

- When using indexed unit parameters with a quantity, unit inconsistency warnings did not always refer to the correct index.
- **Search From Begin** from within the first attribute window that contains the search string, could result in an incorrect **Pattern not Found message**.
- An internal error could occur when removing the source file attribute from a module but keeping its sub tree.
- Changing the column width of numerical columns in a **Pivot Table** could in some situations lead to a fatal error.
- The AIMMS Unicode version did not handle true Unicode characters well in the **ActiveX 2D** and **ActiveX 3D Charts**.
- In AIMMS 3.7 the width of elements in the **Pivot Table** that were shown as outer indices was not stored in the state file.
- Incorrect option values could be passed to **XPRESS** when a model was run for a second time.
- Reading CSV files using the ODBC interface raised an error **Driver does not support this function**.
- A project with libraries could not be opened multiple times in end-user mode.
- Some error message that could occur when reading a case from one model into another model in which the set structure was changed are more clear now.
- An error message has been added when two complementarity variables use the same symbolic constraint for their complementarity definition.
- Loading a case with dataset references for which a referred dataset was deleted could result in a fatal application error.
- **Button objects** on old (AIMMS 2.20) pages were not read correctly in AIMMS 3.7. When pressed, a fatal application error occurred.
- Configuration error in MSXML component could lead to an error when loading the AIMMS start page.
- Moving **Data Categories** between library projects was not correctly implemented.
- The tab sequence was not correct on the **Color** tab in the page object properties dialog, so that you could not put the focus on the identifier selection field without using the mouse.
- The **Text Replace** dialog box sometimes obscured the text to be replaced.
- **Search and Replace From Begin** did not replace text that was found in the **Model Explorer** tree, when the search was initiated from within an attribute window.
- The **Debugger** and **Profiler** menu did not always appear correctly in the menu bar.
- **ActiveX 2D Bar Charts** with small numbers on the X-axis (< 0.5), did not scale properly.
- When a compile error occurred because of an incorrect order by attribute of an identifier it was necessary to first remove the order by attribute, even if the situation was fixed by for example changing the Subset Of

attribute.

- An error during the initialization of the data could lead to a situation where the **Data Management Setup** was not read correctly, which could cause case data not to be saved correctly.
- The export of a project could result in an internal error when separate source files were used.
- The menu command **File - Save All**, did also save library project files that were not changed.
- In a **Pivot Table**, the position of the identifier index was not read back correctly from the state file into the outer indices area.
- Number formatting was not done correctly when the 1000 and decimal separators were the opposite of the English default settings.
- All possible suffices for a given identifier type will now be recognized by the syntax highlighter as valid and will be shown in the auto completion dialog.
- Reading an AIMMS 3.6 project with the AIMMS 3.7 Unicode version resulted in an erroneous entry in the **Page Manager** and **Template Manager**.
- Suffices that are not valid for a certain identifier type will be shown as minus signs in the **Pivot Table**.
- The range attribute of a function was not always stored correctly.
- If the **Message Window** has a user defined menubar, closing the project while the **Message Window** was still open resulted in a fatal error.
- A user defined menubar that contains a rather large **Page Navigator** sub-menu structure did slow down the opening of pages, because it was re-build each time.
- User defined menus in combination with the 95 IDE style could result in a sudden exit from AIMMS or a fatal error.
- The predeclared sets **AllUpdatableIdentifiers** and **CurrentInputs** were stored in cases which could lead newly created identifier to become read only after loading an older case.
- Sending a web service request to AIMMS that does not use the m: prefix for the AIMMS generated namespace, resulted in an error message.
- The Linux versions of AIMMS now also support 84 time zones, like the Windows versions do. In addition, the United States rules for daylight saving time to be used for 2007 and onwards are incorporated.
- A **Composite Table** with multi-line entries did not listen correctly to the text color property and the alignment property.
- After changing the range or type of parameters, cleandependents did not always work correctly.
- The AIMMS compiler now accepts the property Level in a property statement.
- Advanced sensitivity analysis is now also available for RMIP models.
- When in a **Table** object, a suffix is used that is controlled by a property (for example: `SmallestRightHandSide`, which is controlled by the constraint property `RightHandSideRange`), then if the containing page is re-

opened without the property being set, the table is not displayed correctly. In fact, the suffix was then ignored.

- A fatal error appeared when displaying the data of a set after emptying its Order By attribute.
- In case **XPRESS** was used as MIP solver the progress window could display incorrect values for the Best Solution and the Gap.
- The daylight saving time for timezones south of the equator was not always calculated correctly.
- Units were not always casted correctly when using a unit parameter in a for loop.

What is new in AIMMS 3.7

This documentation reflects the state of AIMMS version 3.7. Compared to AIMMS 3.6, the following major new and extended functionalities have been added to the system:

What is new in AIMMS 3.7?

- integrated GIS support in the AIMMS network flow object,
- syntax editor,
- stochastic programming support,
- parallel solver sessions,
- improved multi-developer support,
- extensions to the pivot table object,
- extensions to the web services support,
- solver additions and updates, and
- additional supported platforms.

Integrated GIS support

The network flow object of AIMMS 3.7 has been extended with the capability to dynamically retrieve background images from one or more GIS servers, as illustrated in Figure 4.1. The GIS-generated background can consist of several layers, possibly obtained from multiple GIS servers. Alternatively, GIS files can be generated from within your AIMMS application. AIMMS supports retrieval of data layers from any GIS server that supports the WMS, WFS and GML standards.

Integrated GIS support

This GIS support of AIMMS will allow end-users to depict their geographical related solution data on a dynamic map which will enhance visualization, and ease the interpretation and modification of input/solution data for both end-users and developers within our integrated AIMMS environment.

Benefits



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

Syntax editor

AIMMS 3.7 has been extended with a state-of-the-art syntax highlighting control as illustrated in Figure 4.2, with support for such features as:

Syntax editor

- user-configurable formatting for keywords, elements, strings, and identifier types,
- collapsible code blocks,
- bracket matching capabilities,
- auto-indentation and indentation guidelines,
- member lists (e.g. to select an identifier from a namespace, or a suffix), and
- function and procedure prototype tooltips.

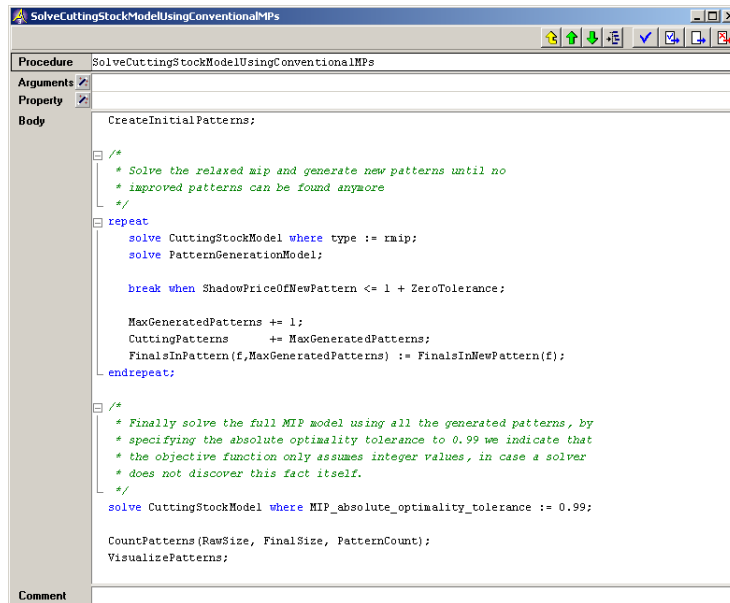
With the introduction of a syntax highlighting control in AIMMS, modifying AIMMS source code and visual code inspection will become much easier. Also, the user configurable formatting (personal profiles) and coloring makes it possible to clearly and visibly distinguish between variables and parameters in, for instance, constraint definitions.

Benefits

Stochastic programming support

AIMMS 3.7 will offer support for generating a stochastic LP/MIP recourse model from any given deterministic model, without the need to reformulate the deterministic model. By only supplying additional attributes for selected parameters, variables and constraints, AIMMS can generate both a deterministic and

Stochastic programming support



```

Procedure SolveCuttingStockModelUsingConventionalMIPs
Arguments
Property
Body
  CreateInitialPatterns:
  /*
  * Solve the relaxed mip and generate new patterns until no
  * improved patterns can be found anymore
  */
  repeat
    solve CuttingStockModel where type := rmip;
    solve PatternGenerationModel;

    break when ShadowPriceOfNewPattern <= 1 + ZeroTolerance;

    MaxGeneratedPatterns += 1;
    CuttingPatterns      += MaxGeneratedPatterns;
    FinalsInPattern(f,MaxGeneratedPatterns) := FinalsInNewPattern(f);
  endrepeat;

  /*
  * Finally solve the full MIP model using all the generated patterns, by
  * specifying the absolute optimality tolerance to 0.99 we indicate that
  * the objective function only assumes integer values, in case a solver
  * does not discover this fact itself.
  */
  solve CuttingStockModel where MIP_absolute_optimality_tolerance := 0.99;

  CountPatterns(RawSize, FinalSize, PatternCount);
  VisualizePatterns;
Comment

```

Figure 4.2: Example of AIMMS syntax editor

recourse model from the same formulation. Stochastic programming in AIMMS is discussed in full detail in Chapter 18 of the Language Reference.

To solve the recourse model, AIMMS 3.7 will generate and solve the corresponding deterministic equivalent, either with explicitly or implicitly added non-anticipativity constraints. Support for an open stochastic Benders solution scheme (implemented using the GMP library) is planned for AIMMS 3.8.

Solving stochastic models

Various user adaptable templates for generating a scenario tree and the corresponding stochastic input data for the recourse model are available in the form of a system module. This module can be imported into any AIMMS model implementing a stochastic model.

Scenario generation

The support of stochastic programming within AIMMS allows one to solve mathematical models with uncertainty to optimality and create robust solutions without structural changes to the underlying deterministic model.

Benefits

Parallel solver sessions

The GMP library of AIMMS 3.7 has been extended with support for parallel solver sessions, i.e. solver session running asynchronously in a separate process or thread of execution. This allows a modeling application to solve multi-

Parallel solver sessions

ple math programs in parallel on one or more processors. Support for parallel solver sessions is discussed in more detail in Chapter 19 of the Language Reference.

The controlling AIMMS model will be notified when an asynchronous solver session has completed, or when some user-definable event has occurred. This allows an AIMMS developer to create advanced solution algorithms for demanding applications that use the available computer resources to their full capacity.

Synchronization

For advanced optimization applications in which multiple independent mathematical programs can be solved simultaneously, this may dramatically increase the application performance on multi-processor computers or multi-core processors.

Benefits

Improved multi-developer support

AIMMS 3.7 offers substantially improved support for multiple developers to work on a single AIMMS application. AIMMS 3.7 applications can be divided into multiple library projects for relatively independent tasks that can be distinguished in the application, where the main project should glue together the parts of the model and GUI provided by the individual library projects. Library projects are discussed in more detail in Chapter 3 of the User's Guide.

Improved multi-developer support

Compared to the source code modules supported in AIMMS 3.2 and up, library projects can not only contain a part of the applications model source, but also the associated pages, templates and menus that must be included in the overall application end-user GUI on behalf of the library. Library modules provide an interface of public identifiers, while keeping all identifiers not in the interface strictly private. This allows library developers to work independently on their assigned sub-projects.

Comparison with modules

This feature will significantly increase AIMMS capabilities to allow multiple developers to work on all aspects (i.e. model and GUI) of a single application. It also allows for effective reuse of shareable parts of an AIMMS application.

Benefits

Pivot table extensions

AIMMS 3.7 offers various extensions to the graphical pivot table object introduced in AIMMS 3.6, including:

Pivot table extensions

- support for domain identifiers to influence the sparsity of a pivot table,
- advanced sorting capabilities of rows and/or columns,

- allow identifier suffices to count as an extra dimension,
- support for check boxes as cell values,
- support for grand totals, and
- ability to extend the table with several layers of additional indices that can be mapped to existing indices.

The latter extension enables application-specific aggregations. For instance, adding a set of quarters that are mapped to months and a set of years that are mapped to quarters, allows the display of both quarterly and yearly totals for data that is defined over months only.

Even more functionality and flexibility is available to provide more power to both the developer and end user to visualize and inspect data in the pivot table objects.

Benefits

Web services extensions

The support for web services introduced in AIMMS 3.6 has been significantly extended in AIMMS 3.7. Web services in AIMMS 3.7 support full asynchronous messaging between a community of AIMMS agents, and any external application that implements a web service according to an AIMMS-generated WSDL description of the service.

Web services extensions

The web service listener process of AIMMS has been implemented as a separate service in AIMMS 3.7. This has significantly eased the configuration of AIMMS web services compared to AIMMS 3.6, where the listener process was implemented as a rather hard to configure virtual directory under the IIS web server.

Easier configuration

The extended set-up allows for a much more effective use of AIMMS in Service Oriented Architectures (SOA). In addition, the new support for asynchronous messaging allows the creation of a distributed application consisting of one or more AIMMS instances and external applications.

Benefits

Solver additions and updates

AIMMS 3.7 has been extended with a link to the KNITRO 5.0 solver from Ziena Optimization, Inc. KNITRO is a world-class solver for NLP, complementarity and MPCC (mathematical programs with complementarity constraints, also commonly known as MPEC) models. The KNITRO solver can be purchased as an add-on to your AIMMS system.

KNITRO 5.0

AIMMS 3.7 supports the MPCC class by allowing you to formulate optimization problems with complementarity constraints in their constraint sets. To solve MPCC models, AIMMS requires a link to the KNITRO solver. Solving MPCC models is discussed in full detail in Section 17.4 of the Language Reference.

*Support in
AIMMS*

The CPLEX solver has been updated to version 10.1. Compared to CPLEX 9, CPLEX 10.1 provides

CPLEX 10.1

- improved performance,
- improved infeasibility analysis,
- solution polishing for finding the best solution for difficult MIP problems,
- advanced restart capabilities for MIP problems, and
- indicator constraints, a very efficient way of controlling whether or not a constraint takes effect, based on the value of a binary variable.

To support the indicator constraints feature of CPLEX 10.1, constraint declarations in AIMMS now support a new property `IndicatorConstraint`, which activates a new attribute `ACTIVATING CONDITION`. The value of this attribute is directly translated to the CPLEX 10.1 indicator constraints facility. The support for indicator constraints in AIMMS is discussed in full detail in Section 14.2.3 of the Language Reference.

*Support in
AIMMS*

AIMMS 3.7 supports these new constraint types for the BARON solver through the `.Convex` and `.RelaxationOnly` suffices for constraints in your model. The use of these suffices is explained in full detail in Section 14.2.5 of the Language Reference.

*Support in
AIMMS*

By adding new solvers, and keeping up-to-date with existing solvers and solver features, AIMMS allows you to

Benefits

- solve more model types, and
- solve existing models more efficiently.

Supported platforms

AIMMS 3.7 will be available for Linux x86_64 RHEL 4 platform. This port supports the same functionality as the Linux x86 version, while the 64-bit addressing allows larger models to be solved.

*Linux x86_64
port*

AIMMS 3.7 has been tested on the latest available beta versions of Windows Vista, and will be supported on Windows Vista as soon as it is officially released by Microsoft.

*Support for
Windows Vista*

By extending the support for AIMMS to more computing platforms and updating the support for existing platforms, you are able to benefit from improved hardware and operating system architectures.

Benefits

Improved functionality and bugs

In the AIMMS 3.7 official release, the following minor issues have been fixed or improved compared to the AIMMS 3.6 release.

*Improved
functionality*

- There is a new option category **Project - Page editing defaults**. This option category contains three options that specify the grid settings of pages. See the help file for more details.
- The **Pivot Table** is extended with a setting to specify whether inactive data should be shown. The **Show Inactive Data** setting can be found on the **General** tab of the **Pivot Table** properties.
- When pressing the **Del** key, or entering an empty string as new value in the **Table**, **Pivot Table**, **Composite Table**, **Scalar Object** or **Sparse List Object**, AIMMS will apply the default value of the identifier. In previous versions this was always translated to 0.
- If you have selected a single cell in a **Composite Table**, and that cell is not editable, then pressing a character on the keyboard will search for the next entry in the same column that starts with the typed character.
- In a **Navigation Object** with style **Tree** or **Extended Tree**, there is a new setting **Open Page upon Single Mouse Click**.
- The **Table** has two new settings: **Show All Horizontal Grid Lines** and **Show All Vertical Grid Lines**.
- For variables and constraints, data pages will now show all suffices for which non-default data is available, in a single **Pivot Table**.
- The **Sparse List Object** can now display an identifier with more than 32767 elements.
- In the **Page Editing** menus, the item **Used Identifiers** has been moved from the **View** menu to the **Edit** menu. This is more consistent with its location in the **Menu Builder** and **Page Manager** menus.
- When creating a **Composite Table** with in its domain a parameter instead of an index, then the parameter is used for both the domain and the first data column.
- In the **Text** properties of a **Table**, **Scalar Object**, **Sparse List**, **Bar Chart** and **Curve**, you can now choose between the display of the identifier name, the identifier text, and any other user specified text. In previous AIMMS versions it was not possible to have a dynamic reference to the identifier text.
- When renaming or moving a **Project User File**, all references to that file in the pages are updated accordingly.
- There is a new setting in **Case Types** and **Data Categories**, namely **Include All Referred (Domain) Sets**. If this setting is checked, then all sets

that are used in the domains of the selected symbols are automatically added to the list of symbols that are saved.

- When using the functions **Round**, **Ceil**, **Trunc**, and **Floor** the unit of the argument will now be used instead of the base unit to do the calculation. With respect to this new behavior two new options are introduced, `Rounding_Compatibility`, and `Warning_Rounding_Ambiguity`. See the help file for details.
- After generation the constraints are reordered in order to reflect the specification in constraints attribute of the Mathematical Program. When this attribute is empty, the order in `AllConstraints` is used. A new option is introduced, `Ordered_Constraint_Presentation`, when this option is switched Off the behavior of previous AIMMS versions will be used, i.e. the constraints will be presented in the order of generation.
- When adding book sections or declaration sections with an explicit name on the **Contents** tab of **Case Types** or **Data Categories** you will get the possibility to store these sections as a reference. When you do this, later modifications to these sections will be propagated in the **Case Type** or **Data Category**.
- The write statement has been extended to support writing of suffices of identifiers. See the language reference for details.
- The compiler issues a warning when a local identifier is made public. The option `Warning_Identifier_Cannot_Be_Made_Public` is introduced that influences the behavior of the compiler with respect to this situation. See the help file for details.
- There is a new option `Default_Datafile_Folder`. The value of this option will be used as the default folder when creating or opening a data file and when exporting or importing cases.
- The function `SessionHasVisibleGUI` has been introduced. If you have an application that can be run both as a stand alone application or as a component, this function can help you to distinguish between the two.
- The option `Combine_Quadratic_Terms` has a new default setting, namely **Automatic**. See the help file for details about this option.
- The function `SetAddRecursive` has been introduced. For details about this function see the function reference.
- The function `FileRead` has been introduced to read the contents of a text file directly into a string parameter. For details about this function see the function reference.
- **Execution Sections** are no longer supported. All execution sections available in a model will be automatically converted to the new **Block** statements when the model is opened.
- There is a new model status in the set `AllSolutionStates`, namely **InfeasibleOrUnbounded**. For now, CPLEX is the only solver that can return this model status.
- There is a new function `ReferencedIdentifiers` that returns all identifiers that are referenced within selected attributes of a subset of `AllIdentifiers`. See the function reference for details.

- The function `DialogGetColor` is introduced to select a color from a standard windows dialog. See the function reference for details.
- Working with AIMMS 2.20 and gams model files is no longer supported. When selecting such a file it will be upgraded to AIMMS 3 style. AIMMS 2.20 projects cannot be converted. If you still want to use such a project in this version, you will have to convert it first to the AIMMS 3 style in an older AIMMS 3 version.
- The function `Cube` has been introduced to calculate a value to the power of 3. See the function reference for details.
- In the breakpoint condition of the debugger, you can now specify a scalar reference using the bound indices of the statement.
- In developer mode a dialog page can now always be closed using the standard close button.
- There are four new information columns available in the profiler. See the help file for details.
- It is now necessary for .Net 2.0 to be installed before using AIMMS.
- The installation of AIMMS has been modified. The standard installation is now a .exe file that downloads any prerequisites that are not yet installed on the computer as part of the AIMMS installation. If you want to install AIMMS on a computer that does not have an internet connection you should make sure all the prerequisites are already installed on the computer. The prerequisites can be downloaded from the AIMMS 3.7 download page on our website.

In the official AIMMS 3.7 release a number of bugs present in the previous functional release have been fixed. *Bugs*