AIMMS 3.9

Release Notes for Win32 Build 3.9.9

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

Paragon Decision Technology

March 21, 2011

Contents

Chapter 1

System Requirements

System requirements

This chapter discusses the system requirements necessary to run the various components of your Win32 AIMMS 3.9 system successfully. When a particular requirement involves the installation of additional system software components, or an update thereof, the (optional) installation of such components will be part of the AIMMS installation procedure.

1.1 Hardware and operating system requirements

Hardware requirements

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

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

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

Supported Windows versions

The Win32 version of AIMMS 3.9 is designed to run under

- Windows 2000,
- Windows XP,
- Windows Server 2003,
- Windows Vista,
- Windows 7, and
- Windows Server 2008.

The Win32 version of AIMMS 3.9 Unicode is supported under

- Windows 2000,
- Windows XP,
- Windows Server 2003,
- Windows Vista,
- Windows 7, and
- Windows Server 2008.

Running the Win32 version of AIMMS 3.9 versions under the 32-bit Windows versions 95/98/ME/NT 4.0 is not supported.

1.2 ODBC and OLE DB database connectivity issues

Database connectivity

Before you can start using the database connectivity features built into the Win32 version of AIMMS 3.9, the following ODBC and OLE DB components need to be installed on your computer:

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

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

Microsoft Data Access Components

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

Data Access Components installation

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

Oracle ODBC drivers and OLE DB providers

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

Prefer native Oracle over MDAC drivers

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

OLE DB issues

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

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

1.3 Viewing help files and documentation

AIMMS *documentation*

The AIMMS 3 User's Guide, the AIMMS 3 Language Reference, the AIMMS 3 Function Reference, and the AIMMS book on Optimization Modeling are available online as Adobe Portable Document Format (PDF) files. In order to view or print PDF documents, Adobe Acrobat Reader version 8.1 or higher needs to be installed on your computer. Other PDF readers will not support the integrated documentation reference features such as Help on and Search all documents.

Acrobat Reader installation

The Aimms 3 CD-ROM contains the Acrobat Reader 8.1 setup executable, and you can install it using the CD-browser which automatically starts up when you insert the Aimms 3 CD-ROM into your computer. Alternatively, if you have downloaded the Aimms 3 installation executable from the Aimms 3 web site www.aimms.com, you can download the Acrobat Reader 6.0 installation executable from the Adobe web site www.adobe.com.

Chapter 2

Installation Instructions

Installation instructions

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

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

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

2.1 Installation instructions

The Aimms 3 CD-browser

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

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

Obtaining Aimms 3 from the web

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

The AIMMS 3 setup program

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

Administrative privileges required

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

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

Component selection

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

2.2 Solver availability per platform

Solvers in installation

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

2.3 AIMMS 3 licensing

AIMMS 3 licensing

AIMMS offers the following two types of licenses:

- single-user licenses, and
- network licenses.

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

Solver	**Version**	**Win 32** **Typical**	 **Complete**	**Win 64** *	**Linux 32** *	**Linux 64** *
AOA		√	√	√	√	√
BARON	7.5.1	-	√	-	-	-
	7.5.2	-	√	-	-	-
	7.5.3	√	√	-	-	-
CONOPT	2.071 C	-	√	-	-	-
	3.11 B	√	√	-	-	-
	3.14 A	√	√	-	√	-
	3.14 G	√	√	-	-	-
	3.14 M	-	-	√	-	-
	3.14 Q	-	-	-	-	√
CPLEX	8.1	-	√	-	-	-
	9.0	-	√	-	-	-
	9.1	-	√	√	√	-
	10.0	-	√	√	√	√
	10.1	√	√	√	√	√
	11.0	-	√	√	√	√
	11.1	-	√	√	√	√
	11.2	√	√	√	√	√
	12.1	√	√	√	√	√
GUROBI	1.0	-	√	√	√	√
	1.1	-	√	√	√	√
	2.0	√	√	√	√	√
KNITRO	5.1	-	√	√	√	√
	5.2	√	√	√	√	√
	6.0	√	√	√	√	√
LGO	1.0	√	√	-	-	-
MINOS		√	√	√	√	-
MOSEK	5	√	√	√	√	√
PATH	4.4a	-	√	-	-	-
	4.6	√	√	-	-	-
SNOPT	6.1	√	√	√	√	√
	7.2	-	√	√	√	√
XA	13	-	√	-	-	-
	14	-	√	√	√	√
	15	√	√	√	√	√
XPRESS	15	-	√	-	-	-
	16	-	√	-	-	-
	17	-	√	-	-	-
	18	-	√	-	-	-
	19	-	√	-	-	-

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

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

Single-user license protection

Single-user licenses can be used by a single user on a single computer. To enforce the single-user character, AIMMS 3.9 requires that single-user licenses be protected by either

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

teristic of your computer.

You can choose

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

Pros and cons of dongles

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

Pros and cons of nodelocks

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

Physical shipments

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

AIMMS *dongles*

If you request your license to be protected by a dongle, an AIMMS dongle is sent to you along with your AIMMS 3 CD-ROM. If you upgrade from AIMMS 2.20, you can continue to use your existing green Activator dongles used by AIMMS 2.20, *but only with the Win32 version of* AIMMS. The green Activator as well as the grey Sentinel dongle must be connected to the parallel port of your computer. The purple Sentinel dongle must be connected to a USB port of your computer. The AIMMS 3.9 setup program will only install the required device drivers for accessing the grey and purple Sentinel dongles. If you still use the green Activator dongles supplied with AIMMS 2.20, you can obtain the required drivers separately from our FTP site ftp.aimms.com.

Network licenses

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

2.3.1 Personal and machine nodelocks

Two types of nodelocks

Aimms offers two types of nodelocks:

- personal nodelocks, and
- machine nodelocks.

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

Personal nodelock

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

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

Machine nodelock

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

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

Privacy

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

Internet connection and proxy settings

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

Automatic configuration scripts

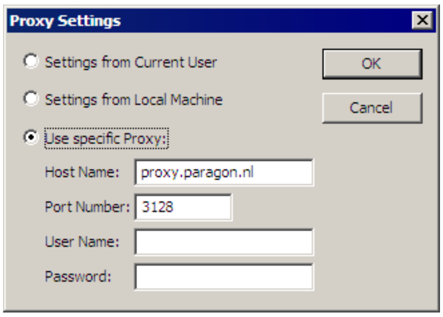
It should be noted that the use of auto-configuration scripts in determining the proxy server will fail if these use any other scripting language than Javascript. This is due to the libraries underlying the SOAP library used by AIMMS to connect to our license server. If you are in this situation, you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

Manual proxy setting

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

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

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



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

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

2.3.2 Installing an AIMMS license

Managing your AIMMS licenses

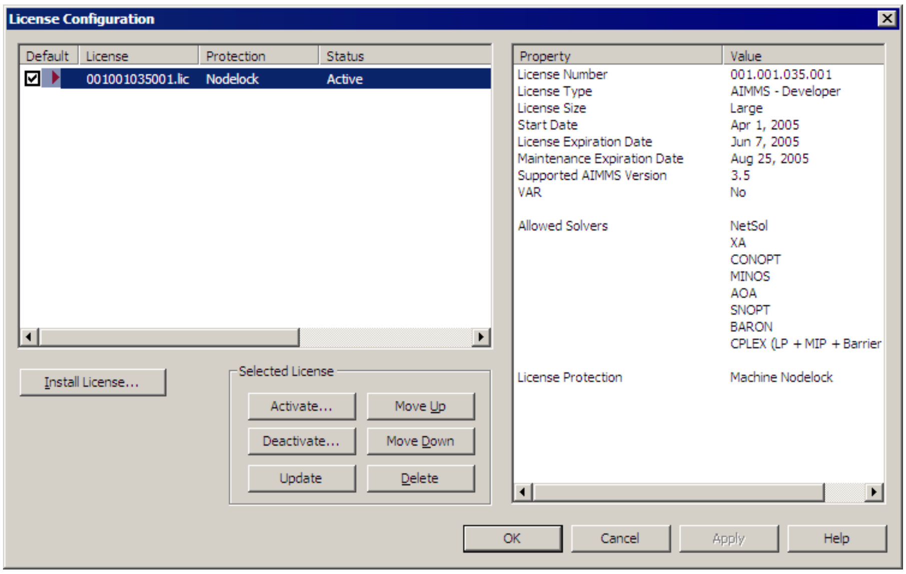
When you start up AIMMS 3.9 for the first time after installation, AIMMS will open the **License Configuration** dialog box illustrated in Figure 2.2. Through this dialog box you can install new AIMMS licenses and manage all AIMMS licenses that already have been installed on your computer.

Installing a new license

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

- existing AIMMS 3.8 licenses,
- nodelocked licenses,
- dongled licenses,
- network licenses,
- evaluation licenses, and
- student licenses.

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



Default	License	Protection	Status
☑	001001035001.lic	Nodelock	Active

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

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

Single-user nodelocked licenses

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

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

You have the choice to request a personal nodelock or a machine nodelock. A personal nodelock must be requested online, a machine nodelock can be requested online or offline. Refer to Section 2.3.1 for a more detailed introduction to personal and machine nodelocks.

Single-user dongled licenses

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

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

Network licenses

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

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

Evaluation licenses

To install an AIMMS evaluation license you need the following information

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

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

Student licenses

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

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

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

2.3.3 Managing AIMMS licenses

Managing multiple AIMMS licenses

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

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

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

Default licenses

In the **License Configuration** dialog box displayed in Figure 2.2, all AIMMS licenses installed on your machine will be displayed in the left pane of the dialog box. The license details of each license are displayed in the right pane of the dialog box. During startup AIMMS will consider all licenses in the left pane of the **License Configuration** dialog box which have the **Default** column checked, and will use the first valid license it finds starting from top to bottom. Using the **Move Up** and **Move Down** buttons you can change the order in which AIMMS will search the list.

Transferring licenses

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

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

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

Emergency nodelocks

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

2.3.4 Location of license files

Location of license files

AIMMS keeps its license and configuration files in the folder

```
Paragon Decision Technology
```

of the common application area of your computer. On Windows 2000, Windows XP, and Windows Server 2003, the common application area is located, by default, at

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

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

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

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

Do not move nodelock files

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

User specific configuration

You can specify whether the license and solver configuration that AIMMS uses is the same for any user of the machine, or different for each individual user. To prevent problems when running AIMMS as part of a computer service, AIMMS 3.9 will by default use the same configuration for any user. To modify this behavior, you should edit the file `UserDistinction.cfg` in the common `Config` folder. In the file `UserDistinction.cfg.default`, straightforward directions are given on how to edit it. If no `UserDistinction.cfg` file exists, AIMMS will use the `UserDistinction.cfg.default` file instead. You can use this file as a base for setting up your own configuration.

Any User

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

Current User

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

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

On Windows Vista and Windows 7 it is:

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

Accessing configuration files

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

Version dependent configurations

Inside the `Config` folder of the `(any)user` folder, each major AIMMS version (3.6, 3.7, 3.8, 3.9, etc.) will create its own specific subfolder when it needs to write a configuration file. During an attempt to read, AIMMS will first look for the specific file in the folder that matches its own major version number, and otherwise it will subsequently search through the folders of previous versions. In other words, when you upgrade to a new AIMMS version, initially your configuration will be the same as the one you were using for the previous AIMMS version, but if you change something in the configuration, this will only affect the configuration of the AIMMS version you are working with.

Network License Client Files

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

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

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

Project dependent configuration

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

2.4 OpenSSL license

OpenSSL included - in separate RPC installation only

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

OpenSSL license

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).

SSLeay license

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

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 acknowledgement: ”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

Getting support

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

3.1 Reporting a problem

Problem fixing

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

Reporting a problem

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

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

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

Problem processing

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

Modeling support

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

Automatic Error Reporting

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

3.2 Known and reported issues

Known issues

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

RPC

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

Windows Vista/Windows 7

- Deploying a web service with attachments in Windows Vista or Windows 7, with a manually started AIMMS agent project, may result in temporary attachment files not being removed from the system's temporary directory. This happens because the `AIMMS HTTP/RPC Handler Service` runs under the `SYSTEM` account, while a manually started AIMMS agent project usually does not. If the non-`SYSTEM` account under which the AIMMS agent project runs does not have deletion rights on the `SYSTEM` temporary directory (which usually is the case), the temporary attachment files cannot be deleted. There are two ways to work around this problem:
  1) By using an automatically started AIMMS agent (which, in a deployment scenario, is the preferred way of starting up AIMMS agents).
  2) By stopping the `AIMMS HTTP/RPC Handler Service` and instead starting the `AimmsServiceHandler.exe` executable. This file is located in the AIMMS `Common\Bin`-folder.

Chapter 4

Release Notes

Release notes

This chapter contains a description of the new features developed for AIMMS 3.9. In addition, it contains the release notes of all AIMMS 3.9 releases. For each release, the following items are listed:

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

AIMMS *3.9 versions*

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

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

AIMMS 3.9 build 9, release date 2011-03-21

Build 3.9.9 2011-03-21

In this last AIMMS 3.9 Software Update, the following issues have been fixed or improved.

Issues . . .

- AIMMS now also supports Acrobat X.
- Two new **Math Program Inspector** tolerance options have been introduced. These allow more fine-grained control over the behavior of the feasibility and unboundedness analysis in the **Math Program Inspector** during special cases. The new options are `Feasibility_tolerance` and `Unboundedness_analysis_bound`.
- The AIMMS profiler now shows the running times of assertions.

. . . GUI

- The **Page Entry** procedure on a subpage of either the **Tabbed Page** object or the **Indexed Page** object, could not refer to any of the page object tags that existed on that same page.
- The background color of a multiline **Scalar** object was not always updated correctly.
- The **Page Navigation** object with text buttons did not respond to changes in the titles of the displayed pages.
- Removing entries from the list of explicit identifiers, while the contents of a **Pivot Table** were specified through the set of 'Implicit Identifiers', could lead to a crash.
- Header heights in the **Pivot Table** object are now also recalculated after each header move.

. . . Compiler and Execution

- A superfluous error message has been removed, which occurred when reading [-] for a unit from a text file.
- In the expression

  ```
  q(i) := if (p(i)) then
  ...
  elseif (q(i)) then
  ..
  endif ;
  ```

  a superfluous check on the data type of q(i) has been removed.
- There was a superfluous error message on indexed variables X(i) with an indexed unit UP(i) used with an expression like X(j).unit, whereby indices i and j were different (but compatible).
- There was a performance problem in the matrix generator for symbolically quadratic constraints.

Aimms 3.9 build 8, release date 2010-12-08

Build 3.9.8 2010-12-08

In this Aimms 3.9 Software Update, the following issues have been fixed or improved.

Improvements

- The evaluation of inline variables at the end of a solve statement has been improved, by taking the index domain conditions into account.
- During mathematical program generation, the execution of the index domain condition of a variable can be dense due to lags/leads on an index ranging over an ordered set. Now the index domain condition of the constraint and/or other conditions in the expression are considered as well.

Issues . . .

- Sometimes, when pasting data into the **Pivot Table** in which tooltip text was specified, the data did not appear in the correct order.
- Read-only name change files of libraries could lead to spurious entries in the name change file of the main model.
- When the name change file corresponding to the main model was read-only, name changes might not be stored in the name change files of the libraries.
- The error positioning of an execution error, which occured during the evaluation of an inline variable, might have pointed to the evaluation of a defined scalar parameter.
- When executing a stored database procedure in SQL Server, using an output argument of type `varchar`, AIMMS raised a severe internal error.
- Units declared in modules or libraries could lose their prefix when used inside the model editor.

. . . Solving

- In rare cases, solving with **CONOPT 3.14G** could result in a severe internal error.
- The Hessian was not passed in case an NLP solver (like **CONOPT**) was used to solve a QP or QCP model. This could result in reduced performance.
- A mathematical program in which all symbolic constraints contain non-linear terms, but which after generation contains a linear system (LS) (i.e. no non-linear elements), could cause a severe internal error when executing selected actions from within the **Math Program Inspector**.
- Infeasibilities in linear systems (i.e. LP models without an objective) were not always detected correctly by the **Math Program Inspector**.

AIMMS 3.9 build 7, release date 2010-09-22

Build 3.9.7
2010-09-22

In this AIMMS 3.9 Software Update, the following issues have been fixed or improved.

Issues . . .

- Repeatedly creating and deleting a section that is the last but not the only node of another node might lead to a severe internal error.
- An action in the **Math Program Inspector**, involving a solve, could result in a crash if the model contained indicator constraints.

. . . GUI

- The marginal data (`.ShadowPrice` and `.ReducedCost`) is obtained from a solver when the corresponding properties are set for the identifiers or when the option `always_store_marginals` is switched on. When only the option `always_store_marginals` is switched on, this information was presented erroneously on the page objects of the GUI.
- GUI expressions containing superfluous references with prefixes to identifiers in the same library could confuse the name change module, lead-

ing to pages that could no longer be opened after a name change.

- The drop-down list in the outer index tree of a **Pivot Table** did not always show all elements.
- A fatal application error could occur when copying an entire **Table** with copy settings 'All' and 'Include Headers'.

. . . Compiler and Execution

- The element numbering of root sets with a dynamic element ordering could become corrupted in some cases (at least involving deleting an element and executing `CleanDependents`), resulting in element names like `'Element xx not found for domain set xxx'`.
- A symbolic MINLP that, when being generated, results in a MIP, was still solved as MINLP instead of MIP.
- Multiplying two scalar unit parameters, whereby one of them has value `[-]`, could lead to the erroneous result of `[-]`.
- Part of the reading of data from a text file involves lexical analysis (splitting strings into tokens). Lexical analysis is also needed for the function `StringToUnit`. In AIMMS 3, these tasks are performed by the same code, and only one instance of that code can run at any time. Therefore, an error message is now issued when the updating of a definition needed during the reading of a text file, involves the function `StringToUnit`.

AIMMS 3.9 build 6, release date 2010-06-16

Build 3.9.6 2010-06-16

In this AIMMS 3.9 Software Update, the following issues have been fixed or improved.

Issues . . .

- Storing exceptionally large cases could fail. This resulted in cases which were still present in the **Data Manager** (albeit without date indication), but could not be opened.
- AIMMS issued a severe internal error when a definition contained an error and was still evaluated.

. . . GUI

- If a **Table** object had a procedure upon change in which functions like `PageSetCursor` or `PageSetFocus` were called, then pasting a value to the table could lead to a crash.
- The **Gantt Chart** object did not scroll through the week(number)s correctly.
- When calling the `PrintPage` function for a non-print page, a dialog pops up in which you can specify the size of the bitmap-based print. The `PrintPage` function now has an optional argument that allows you to skip this dialog, and to use the default settings instead.

. . . Pivot Table

- The identifier that was shown when the indentifier index was positioned in the outer index area of a **Pivot Table**, did not always correspond to the data slice that was shown in the grid area.
- In the **Pivot Table**, the header width did not always respond correctly when it was specifed by a parameter.
- In the **Pivot Table**, resizing column/header widths that were specified through an (updatable) identifier did no longer work correctly.
- AIMMS now tries to set the outer indices of a **Pivot Table** without an associated state file, such that the initial grid area is not empty.
- In the **Pivot Table**, the identifier that is specified in the 'Store Selection in' property, was not updated when the selection in the **Pivot Table** changed.

. . . Solving

- The solving time reported in the **Progress Window** was not correct for models using a time unit.
- In the **Math Program Inspector**, an action that involves a solve, could result in a crash if the model contained indicator constraints.

AIMMS 3.9 build 5, release date 2010-03-10

Build 3.9.5 2010-03-10

In this AIMMS 3.9 Software Update, the following issues have been fixed or improved.

Issues . . .

- The index wizard now checks whether the names suggested for new indices already exist.
- **Slider** objects were not cleaned up correctly, which sometimes caused strange errors when switching projects (without closing AIMMS).
- Ensuring subset consistency at the end of of a case load is now controlled by the option `case_ensure_subset_consistency_method`. If and how inconsistencies are reported, is now controlled by the newly introduced option `warning_ensuring_subset_consistency`. Please refer to the AIMMS Help for detailed information.
- Data stored in cases for identifiers that were declared in a library or module, but are now deleted, could overwrite the data of identifiers in a global scope and cause superfluous case consistency warnings.
- When running a batch of cases, the log file `_batch_.log` did not contain sufficient newline characters.
- **Excel** version 2007 was not always recognized properly by AIMMS, leading to unjust error messages when using the functions `ExcelColumnName` and `ExcelColumnNumber`.

. . . GUI

- An unnecessary recompilation was required for subsets of compound sets, when they were used in a **Multiple Selection** object.
- When making changes in the **Contents** tab of the **Properties** dialog box of the **Gantt chart** object, entries in the **Element Text** tab were lost.
- Checkmarks in user menus were not always updated correctly after a data change in the underlying identifier.
- The function `GeoFindCoordinates`, which internally uses a third party web server to obtain the coordinate data, did not work anymore, since the third party does not provide the data anymore. Therefore, the function has been removed from the AIMMS language.
- **2D and 3D Chart** ActiveX objects could not handle the values `inf` and `-inf` well, which could lead to crashes when trying to display them in combination with regular values. Now, `inf` and `-inf` values are not displayed anymore.
- The function `PrintPage` did not work correctly for non-printing pages. Now, it makes a bitmap print of the page, similar to the functionality of the **File - Print** menu command.
- Disabling submenus (built with the **Menu Builder** tool) that were preceded by hidden submenus, did not work correctly.
- The **Selection** object with radio buttons did not work correctly when the underlying data was a 1-dimensional parameter.

. . . Pivot table

- The **Contents** properties of a **Pivot Table** object were not always updated correctly when identifiers were added through drag-and-drop from the model tree.
- Trying to display the set `Integers` in a **Pivot Table**, **Table** or **Composite Table** caused a fatal application error. In the **Pivot Table**, this set is now filtered out from the identifiers to be displayed. In general, it is not recommended to try to display the contents of the set of all integers in the AIMMS GUI.
- When changing the cardinality of the multiple case selection from zero to a nonzero value (or vice versa), it could happen that a 'multiple case' **Pivot Table** object was not displayed correctly.
- Data changes in **Pivot Table** objects that contain an aggregator in the first row or column, sometimes resulted in a fatal application error.
- When data in a **Pivot Table** object changes, the sort criterion was only reapplied when the option `store_entire_tree_layout` was set. Now, the last sort is always reapplied.
- If not all paths in a tree in the **Pivot Table** were of equal length (e.g. both a one and a two-dimensional identifier are displayed in the row or column tree), it could happen that a unit was not displayed when it should.
- When switching tabs in a **Tabbed Page** object, the `PivotTableSaveState` function in the **Page exit** procedure of the page that was being deactivated, could not find the tags of the objects on that page.

- In **Pivot Table** objects, the use of checkboxes that use indices that are linked to element parameters, could lead to a fatal application error.

. . . Solving

- After the following sequence of statements

  ```
  solve MP;
  GMP::Instance::CalculateSubGradient( 'MP' );
  solve MP;
  ```

  AIMMS no longer regenerates the entire coefficient matrix during the second solve.
- An incorrect option value for `time_limit` was passed to **CONOPT** after creating a new solver session.
- The **Math Program Inspector** can no longer be opened for network models (solved with **NETSOL**).
- Calculating sensitivity information for the **Mathematical Program Inspector** (during the postsolve stage), was not executed with **GUROBI** if the model contained SOS constraints.
- **GUROBI** did not handle cutoffs correctly for maximization problems.
- Resolving a model with cut pool constraints in the **Math Program Inspector**, resulted in a crash.
- AIMMS failed to write the node file to disk for **GUROBI** 2.0.

. . . Compiler and Execution

- An unnecessary recompilation was required for subsets of compound sets, when they were used in a **Multiple Selection** object.
- A superfluous compilation error was given for the test

  ```
  idIndex in SectionName
  ```

  where `idIndex` is an index in a subset of `AllIdentifiers` and `SectionName` is a section in an AIMMS model.
- The "write- to-window" feature of the **put file** could slow down execution, when the output window was not the topmost window.
- Inline variables, which are not referenced inside other constraints, but which are referencing defined parameters, did not use the correct values of these defined parameters when evaluated.
- Consider the following parameter declarations:

  ```
  PARAMETER:
  identifier :  x_lo ;
  PARAMETER:
  identifier :  x_up ;
  ```

  and variable declaration:

```
VARIABLE:
identifier:  x
range :  [x_lo, x_up]
```

Then a reference in an expression to the bound of such a variable may give an unexpected value, for instance in a statement like (with the above declarations):

```
x_mid := ( x.lower + x.upper ) / 2;
```

This only happens when the parameters are scalar.
This effect can be suppressed via the new option

```
scalar_suffix_reference_substitution
```

with range {off, on}, which can be found in the compatibility option section. For AIMMS 3.9, the default value of the option is set to off.

AIMMS 3.9 build 4, release date 2009-12-16

Build 3.9.4 2009-12-16

In this AIMMS 3.9 Software Update, the following issues have been fixed or improved.

Improvements

- There is a new version of the **GUROBI** solver: **GUROBI** 2.0.

Issues . . .

- The function GeoFindCoordinates, which internally uses a third party web server to obtain the coordinate data, does not work anymore, since the third party doesn't provide the data anymore. Unfortunately, at the moment of writing, it is not known whether this is only temporarily. In the meantime, we advise you not to use this function. If it turns out that the third party web server discontinues its support and that no alternative web server is found, we might remove the function in future AIMMS releases.
- Name change (.nch) files associated with .aeb files in one way encrypted project libraries, were not properly read in.
- Multi-line **Scalar** objects did not always print the correct number of lines on print pages.
- Importing .dat files in the **Data Manager** did not always overwrite existing entries.
- When reading or writing web service requests or responses, the index attributes of <Value> tags for multi-dimensional message arguments did not always honor the WSDL specification.
- Vertical bands in the **Gantt chart** object, that are less than 1 pixel wide, are now printed as a single line.

- Executing the **Export Viewer Project** menu action in an end-user project did not work correctly if the model files were encrypted. It only worked when the project was started from a `.aimmspack` file.
- After changing a value with a non-base unit on a page, applying an undo (CTRL + Z) did not result in the correct previous value.
- When you try to do an export to an existing and read-only `.aimmspack` file, you now get a proper error message.
- When copying a variable, AIMMS now also copies attributes such as `priority`, `relax status` and `stage`, whose availability depends on the contents of the attributes range and property.
- When using any of the page functions, you can now use the `::` prefix to explicitly refer to a page that is contained in the main project.

. . . GUI

- In rare cases, when using a **2D chart** object of the High/Low/Open/Close type, switching between properties could cause a fatal application error.
- When closing a page from within the double-click procedure of a **Selection** object, the page was not closed correctly.
- When browsing for an identifier to be used in an **ActiveX** object, the buttons for permutating the indices were not available.
- If `PageGetActive` and `PageGetFocus` returns a page whose name is not unique over all libraries, this page name will be prefixed with the prefix of the library (or with `::`, in case of the main project).
- Bitmap buttons for which the file name was specified by a string parameter, were not correctly saved and restored upon re-opening the AIMMS project.
- A reverse link element parameter on an index of the **Pivot Table**, could result in an error message when the corresponding set did not contain any elements.

. . . Solving

- Lazy equality constraints were passed incorrectly to **CPLEX**.
- A memory leak in the nonlinear presolver has been sealed.
- The MIP start was not passed correctly for models with semi-continuous variables.
- Tabs in the solver configuration file were not handled correctly.
- Stochastic LHS's of ranged constraints could be generated incorrectly.

. . . Compiler and Execution

- The running index of the iterative operator `First` may, in rare circumstances, refer to a non-existing set, causing a severe internal error.
- The suffix `.solutionTime` of a mathematical program is now internally stored in the base time unit, avoiding unit conversion problems when the model base time unit is not equal to seconds.
- In rare circumstances, a severe internal could occur when adding elements one-by-one to a set in an indexed set.
- Compound sets affected by a `CleanDependents` statement (because they are declared over a set that is cleaned), are now cleaned at the start of

that CleanDependents statement, such that identifiers declared over this compound set are also cleaned.

- The procedure OptionSetString did not properly handle options with spaces in their names.
- Using external functions with more than 16 arguments, could lead to corrupting internal data structures. This could lead to severe internal errors when closing the project.

Aimms 3.9 build 3, release date 2009-09-28

Build 3.9.3 2009-09-28

In this Aimms 3.9 Software Update, the following issues have been fixed or improved.

Improvements

- The **Excel Add-In** of the Aimms 32-bit versions can now also recognize and use the 64-bit versions of Aimms, when installed on a 64-bit platform.
- There is a new version of the **CPLEX** solver: **CPLEX** 12.1. This version includes improvements in the performance of solving MIP problems and the *Barrier* algorithm. Furthermore, parallel algorithms in **CPLEX** 12.1 are now included as a standard (i.e. at no additional charge).
- There is a new version of the **GUROBI** solver: **GUROBI** 1.1.

Issues . . .

- In some situations, the method Identifier.RetrieveTable of the **Aimms COM** object gave an incorrect error on the format of the passed row array and/or column array.
- The project option hide_status_bar did not work in the XP IDE style.
- If multiple Aimms sessions on the same pc connected to the License Server simultaneously, an error could occur when a session tried to read a license file that another session was writing to.
- Due to an outdated .xsd file in the installation, XML schema mappings containing **Virtual Attribute** bindings would not validate when loaded in the XML Schema Mapping wizard.
- ExcelRetrieveTable stopped (and reported an error) when the spreadsheet column range contained empty cells. Now it simply skips the corresponding columns.
- Switching between projects within a single Aimms session could lead to retaining a non-default value of the option severe_internal_error_contact in case the new project had a default value for this option.
- The License Installation wizard could give fatal errors when handling nodelock licenses, while no internet connection was available.

. . . GUI

- In the Unicode version of AIMMS, the tab order of page objects was not consistent.
- A page could not be read, if a button had a title defined by a string parameter, with a value containing newline characters.
- When the controlling set of the **Indexed Page** object was changed from within one of the displayed inner pages, a fatal application error could occur.
- When trying to switch a **Gantt Chart** from a continuous range to a discrete range, a fatal application error occurred.
- Reopening a PUT Window with a name length longer than 127 characters, resulted in opening a new window instead of the same window. Opening a PUT Window with a name longer than 127 characters now results in a proper execution error.
- If, during the opening of a page, AIMMS allowed you to browse for an identifier that was not recognized, a crash could occur if the newly chosen identifier had a smaller dimension than the original one.
- The **Identifier Selector** now also lets you select suffices related to extended variables and constraints (if applicable).
- If the string parameter representing the title of a page changed, a tabbed page object did not update the text on the corresponding tab.
- Disabling submenus in an end-user menu did not always work correctly.
- When displaying only a small portion of data in a **2D Chart** object by specifying a minimum or a maximum value in the X-axis scale properties, while the set which defined the X-axis data was very large, resulted in a very slow performance when showing the chart.
- If an **Indexed Page** object has a single inner page, then its Page Exit Procedure is now also called when the containing page is closed.
- The functions `PageCopyTableToClipboard` and `PageCopyTableToExcel` could not find a table that was located on a single inner page of an **Indexed Page** object.

. . . Pivot Table

- When printing a **Pivot Table**, the correct number of lines was not always printed.
- **Pivot Tables** for which sort order information was stored in the state file, were not always displayed correctly.
- By accident, the Value Bar Minimum and Value Bar Maximum properties of the **Pivot Table** were allowed to be specified through indexed identifiers. This has now been restricted: in the explicit identifier category, they should be scalar and in the default category, they can be indexed over the set `AllIndentifiers`.
- For inactive cases, the multiple case view of the **Pivot Table** now also takes into account the restriction imposed for indices referencing subsets of the domain sets of an identifier.
- The height of **Pivot Table** elements that ended with a `'\n'` character was not computed correctly.

- When restoring state info from a **Pivot Table** that had the option Store Entire Tree Layout set, the ordering of elements in ordered sets was not always restored correctly.
- In addition to using the specified Selection - Background Color in the **Pivot Table**, a bar in a selected cell is now colored slightly lighter than a bar in a cell which is not selected, to even better distinguish between cells that are selected and cells that are not.
- **Pivot Tables** that displayed an aggregator in the first row or column, were not always updated correctly after a data change.
- In some situations the sort order that was initially present was not remembered correctly after the data of the **Pivot Table** was changed.
- Specifying **Pivot Table** properties through an identifier declared over the set `AllIdentifiers`, did not work for identifiers that were declared in a module.
- In some situations, initializing or refreshing a **Pivot Table**, for which sort order information had been stored in the state file, could cause a fatal application error.
- In some situations, after drag-and-drop of headers in the **Pivot Table**, the **Pivot Table** would not display correctly.
- In the **Pivot Table** properties, an unnecessary Unit property could sometimes appear near the Value Bar Minimum property.

. . . Libraries

- Renaming a prefix of a nested module could lead to a spurious `::` in the **Prefix** attribute.
- The functions `CaseTypeContents` and `DataCategoryContents` did not return identifiers declared in libraries.

. . . Solving

- The bound analysis in the **Math Program Inspector** now uses presolved bounds (computed by the NLP presolver) if available, and only reports improvements with respect to the presolved bounds.
- The MIP Search Tree tab in the **Math Program Inspector** is not displayed anymore when it is not applicable.
- The default behavior of the **CPLEX** option `pool_intensity` has changed to moderate intensity such that the incumbent callback (if any) only returns distinct incumbent solutions. You can set the option value to `Mild` to get the old default behavior. Please note that both the old and the new default *value* of the option is `Automatic`, only the interpretation of the value has changed.
- AIMMS now generates an error if the **CPLEX** tuning tool runs out of memory.
- The initialization of the **GUROBI** solver could result in a crash on old computers (with CPU's that do not support SSE2). AIMMS now checks whether the computer supports SSE2, and, if not, it will generate an error.
- The function `GMP::Solution::GetObjective` returned an incorrect value for a nonlinear presolved model, which could result in incorrect results when

the **MultiStart** module was used in combination with the nonlinear presolver.

- AIMMS did not accept the `elpar` part of the syntax: `solve mp where direction := elpar;` (where `elpar` was used instead of `maximizing/minimizing` and `elpar` was an element parameter with the set `MaximizingMinimizing` as its range).
- The GMP functions now accept an element parameter with range `AllMathematicalPrograms` at the place where they used to expect only a mathematical program.
- The **MultiStart** module has been changed: solutions are now also stored if the parameter `UsePresolver` is switched on.
- Hessian information was not completely cleaned when regenerating part of a mathematical program. When the structural changes involved variables involved in the nonlinear expressions in an NLP, this could result in an incorrect Hessian communicated to the NLP solver.
- Resolving a non-linear mathematical program (NLP), in which a bound (lower or upper) of a scalar variable with explicit numeric bounds is modified to freeze the corresponding column, could result in incorrect level values being used as frozen values during the solve.
- Printing the constraint listing for a linear model in which a complementarity variable was fixed, could result in a severe internal error.
- The **MultiStart** module no longer restores initial solutions present in the solution repository.

. . . Compiler and Execution

- Consider procedure `Proc(P(i,j))`, where `P` is an output argument and `i` and `j` are indices over locally declared sets (and thus initialized with the actual sets when procedure `Proc` is called). Calling this procedure with `Proc(Q)`, where `Q` is a filled parameter, could result in old elements being retained in `Q`.
- Like the `read from file` statement, the `read from table` statement now expands subsets of `AllIdentifiers` and does not check sets anymore, which are not related to any identifier read in.
- A severe internal error could arise when sparse execution searching of an indexed set would explicitly reference an empty set.
- AIMMS raised a severe internal error when an `option` statement would be executed after a solve statement with a `where` clause. Now a proper error message is created, for example:
  ```
  solve where solver := 'CPLEX 11.2';
  option solver := 'CPLEX 11.2';
  ```
- The procedures `GenerateXML` and `WriteXML` did not omit the XML header if the `Merge` argument was non-zero.
- Reading back data that was indexed over a calendar with a string parameter defining the timeslot format, and its value containing one of the modifiers `%e`, `%E` or `%I`, could lead to superfluous error messages. In addition, when the time slot format could lead to superfluous trailing spaces

in the elements, reading back data could also lead to superfluous error messages.

- The ordering of set elements was not taken into account when writing XML data.
- AIMMS can now read back the data of identifiers of type `MODEL` or `FILE`.
- The definition of a violation penalty parameter was not always updated in time to guard a mathematical program against infeasibility.
- The statement after a `skip` statement in a `then` or `else` clause of an `if-then-else` statement, was not always skipped.
- The expression `element-expression = running-index` could take unnecessarily long to evaluate.
- The performance has been improved in tests such as `index = eltpar`, where the range of `index` and `eltpar` is a set with a specified **Order by** attribute.
- If the option `enhanced_long_run_responsiveness` was set to `automatic`, it was not turned on when running on a Windows Vista machine.

. . . Units of Measurement

- In the **Unit** tab of the Properties dialog box, you could sometimes get an incorrect error message: ”Not a correct unit”, which did not allow you to apply and leave the dialog box.
- The `EvaluateUnit` function returned 0 for a unitless unit parameter. Now it returns 1.

. . . Linux

- After a **Compact Files** command, the project file was no longer readable on a Linux system.
- On Linux, the `FileMove` function did not always work well if the destination location was on a different file system than the original file location.
- Also on Linux, the `Execute` function did not honor all of its arguments.

AIMMS 3.9 build 2, release date 2009-06-23

Build 3.9.2 2009-06-23

In this AIMMS 3.9 Software Update, the following issues have been fixed or improved.

Improvements

- The option `Populate_time_limit` has been added to the set of options of **CPLEX 11.0** and higher.
- The option `Ord_file` has been added for **CPLEX 10.1** and higher. It can be used to write a file containing the variable priority orders.
- The maximum integer value `MaxInt` can now be used in the AIMMS language.
- A **Common Shortcuts** list and an **Available Documents** list have been added to both tutorials.
- An expired emergency nodelock license can now be deleted more easily.

- In the **Pivot Table** you can now specify a **Standard Format** in the default section of the **Contents** tab. For the format of explicit identifiers you can indicate whether or not you want to use the **Standard Format** that is specified as default.
- In a **Pivot Table**, **Value bar minimum** and **Value bar maximum** can now also be specified by a numerical identifier indexed over `AllIdentifiers`.

Issues . . .

- If in both the main project and a library a similar procedure existed with similar local identifiers, then **Show Data** did not always show the data of the selected local identifier.

. . . GUI

- Closing a page from within the **Procedure upon change** of a **Selection** object did not close the page correctly.
- In a **Composite Table**, a column with checkboxes for a sliced identifier did not show the checkbox on the last row(s).
- `PageGetFocus` now also returns the tag of the focus object if that object is placed inside an **Indexed** or **Tabbed Page** object.
- When a **Button** object on an **Indexed Page** or **Tabbed Page** object was removed during one of its own actions (e.g. when the set of the **Indexed Page** object changes), the action list was removed as well, causing a fatal error when AIMMS tried to run or continue the remaining actions.
- Applying a **Foreground** or **Background colour** of a **Button** object was not immediately visible in the button.
- In dialogs, buttons with text coming from a string parameter were sometimes not drawn correctly when an execution error had occurred.
- In an **ActiveX** object, the notation `P( i in MySet )` did not work.
- The `.Stochastic` suffix can now be used in the **2D/3D Chart ActiveX** objects.
- Data in a **Pivot Table** that had been specified by means of a subset of `AllIdentifiers` was not always updated correctly after the current case changed.
- In the **Pivot Table**, displaying an identifier with a unit that was specified via an (indexed) expression, could lead to a very slow performance.

. . . Syntax Editor

- The **Property** attribute is no longer colored by syntax-highlighting.
- The suffix `.Stochastic` for a mathematical program no longer colors as ’Unknown’ in the text editor.

. . . XML

- An XML data file was not always read in correctly if it contained an `xmlns` attribute.
- In the XML schema mapping dialog you can now create mappings to identifiers that are defined in the interface of a library.

. . . Excel Add-In

- The **Aimms Excel Add-In** sometimes displayed the sequence list in the **Execution Sequences** tab on the **Aimms Interface Setup** dialog box too high.
- The Unicode version of the **Excel Add-In** listed Aimms Unicode versions as being installed (in the dropdown list of the **Aimms Interface Setup** dialog box), if only the corresponding Aimms Ascii versions were installed.

. . . Solving

- Aimms could create a superfluous error message when deleting a GMP that had just been emptied (a GMP can be emptied by loading another case).
- Aimms could become non-responsive when the irreducible infeasible set (IIS) was calculated by **CPLEX**.
- Coefficients for diversification filters were sometimes passed incorrectly to **CPLEX**.
- Diversification filters were not generated during a resolve.
- The populate algorithm for **CPLEX** now starts with a normal solve to find the optimal solution (and other incumbent solutions).
- The optional argument `timeout` has been added to the procedure `GMP::SolverSession::Interrupt`.
- The procedure `GMP::Solution::SetMIPStartFlag` did not set the MIP start flag for the first solution in the solution repository of the GMP.
- `GMP::Instance::FixColumns` could fail if the objective variable was part of the set of variables to be fixed.
- Using `GMP::Coefficient::Set` on a stochastic GMP (generated by `GMP::Instance::GenerateStochasticProgram`) could give incorrect results.

. . . Compiler and Execution

- In **AcitiveX** objects and also in **AimmsCOM**, default values with units were converted twice (e.g. a default of 100[%] was converted to 10000[%]).
- `CleanDepedents` on an element parameter with a defined set as range, could use an out-of-date version of the data of that defined set.
- When starting the debugger implicitly by means of `Ctrl + Shift + S`, a newly added breakpoint was not hit.
- The option `Warning_variable_range_violation` was not saved correctly.
- Sometimes the Model Explorer was not updated correctly after an **Import** command.
- The conversion from GAMS to Aimms 3 did not properly handle the '[' in the statement `if [ cond, stmts else stmts ]`.
- Modifying the option `Singleton_unit_namespace` requires a project restart in order to take full effect. The user is now notified of this requirement.
- Errors during the import of a `.aim` file into a section were ignored and only visible in the **Message** window. Now the import is aborted and the **Error/Message** window allows you to position the cursor on the error in the `.aim` file.
- The **Index** wizard accepted Aimms keywords as new indices/parameters.
- Sometimes, an error during a procedure run led to a situation where

AIMMS took a very long time to try and display that error in the **Error** window.
- A division inside a summation operator of a non-indexed constraint, might generate multiple errors which were subsequently erroneously assigned to the next constraint.
- Errors during the generation of a mathematical program now count as ordinary execution errors and will be used to stop execution of the invoked procedure.
- Parameter definitions depending on the results of `GMP::Instance::CalculateSubGradient` were not timely updated.

AIMMS 3.9 build 1, release date 2009-04-17

Build 3.9.1 2009-04-17

In this AIMMS 3.9 Software Update, the following issues have been fixed or improved.

Improvement

- There is a new version of the solver KNITRO, i.e., **KNITRO 6.0**. It can be used to solve mixed integer nonlinear programming (MINLP) problems.

Issues . . .

- After the **Math Program Inspector** had been opened (and closed) the menu commands **Edit - Find** and **Edit - Repeat Find** were disabled.

. . . Libraries

- The **Interface** wizard for a library does not show or add local identifiers anymore.
- The option `Warning_local_in_identifier_collection` was added to specify whether an error or warning should be given when local identifiers are included in an identifier collection attribute, such as the **Interface** attribute of a library.

. . . Solving

- The file `xav14.dll` has been added to the typical installation because it is used by **BARON**.
- A call to `GMP::Column::Add` on a generated mathematical program, created via `GMP::Instance::Copy`, could lead to a severe internal error.
- AIMMS maintains a mapping between the symbolic variables and the actual columns in a generated mathematical program, say `g`. When `g` is regenerated, this mapping is cleared by visiting all variables mentioned in the constraints of the Mathematical Program from which `g` is generated. In addition, AIMMS allows `GMP::Column::Add` to add columns from any variable, including variables not yet mentioned in the constraints of the mathematical program. Regenerating `g` and then again adding columns from variables not mentioned in the constraints of the mathematical program, could lead to inconsistencies in the mapping and consequently to severe internal errors.

. . . Compiler and Execution

- The option Warning_range_violation was already available and now the option Warning_variable_range_violation has been added to specify a different behaviour for the range checking of variables.
  The default of Warning_variable_range_violation is warning.
  The default of Warning_range_violation is error.

AIMMS 3.9 build 0, release date 2009-03-06

Build 3.9.0 2009-03-06

In this AIMMS 3.9 release, the following issues have been fixed or improved. Please note that almost all issues that have been fixed in the upcoming AIMMS 3.8 Software Update, have also been fixed in this AIMMS 3.9 Release.

Improvements . . .

- The function GeoFindCoordinates has been enabled again (it was present in the AIMMS 3.8 Release Candidate, but not in later versions, due to the blocking of the third-party server that delivered the data for this function).
- A wizard is now available for the **Timeslot Format** attribute of a **Convention**.
- When executing a long procedure, Windows Vista sometimes marked AIMMS as **(non-responding)** and also prevented that the **Progress Window** could update its contents. To deal with this, AIMMS has an alternative way to handle various messages from the operation system during a run. This new handling is turned on by default under Windows Vista, but it can be controlled by the new option Enhanced_Long_Run_Responsiveness as well.
- A new method to deploy AIMMS end-user projects via AIMMS RPC has been added.
- Upon an execution error, AIMMS now breaks into the debugger immediately, except in situations where the debugger is unavailable, like when using the end-user version or when using the profiler. The new option Debugger_break_on_execution_error, which has a default value of Off, can be used to control this behavior.
- The **Identifier Info** dialog box has been added to AIMMS. It can be used to easily navigate to the declaration, the attributes and the data of each identifier in the model. It is accessible from the **Tools** menu and the toolbar.
- The **Debugger** can now be accessed using the shortcut key combination Alt + F6.
- AIMMS options in the non-default category can now be found easily in the tree by pressing the new **Locate in tree** button.
- The language has been extended with the intrinsic functions FindReplaceNthString and FindReplaceStrings. See the Function Reference for details.
- The function TestDate has been added. This function determines whether

a date is specified according to a given format. Please see the Function Reference for details.

- There are new options in the **Project Options** dialog to specify which object to use in a data page when inspecting multi-dimensional identifiers. You can find these options in the Option Tree under **Project - Standard data pages**.
- In the **model editor**, collapsed `if/for/while` statements are now shown as collapsed lines with their condition still visible.
- The **Cross Library References** dialog box has been extended with:
  - a **Print** button, and
  - the concept of **preferred references**

. . . Math Program Inspector

- The **Math Program Inspector** has been extended with a **Find** command.
- The **Math Program Inspector** has been extended with features to assist you in analyzing NLP models.

. . . Excel

- Two new functions are available to copy data from AIMMS tables (and pasting them to e.g. **Excel**): `PageCopyTableToClipboard` and `PageCopyTableToExcel`. For details, see the Function Reference.
- The AIMMS **Excel Add-In** has been changed, such that you can now specify the AIMMS version with which you want to open the linked AIMMS project. AIMMS versions from 3.6 and higher can be selected. For details, see the AIMMS Excel documentation.

. . . GUI

- The **Date Time Picker** ActiveX object now supports custom date/time formats similar to the AIMMS date/time format.
- **Pie Charts** have been extended with the possibility to automatically calculate and show the percentages of the individual pie slices.
- The **GIS** background of a **Network** object can now also be obtained from MS Virtual Earth.
- If a **Scalar** object is displaying an element parameter, you can now switch off the auto-complete dropdown list that pops up during editing. This property can be found on the **Format** tab of the properties dialog.
- The **Gantt Chart** has been extended with two new features for coloring the background area: vertical bands (e.g. for specifying a different background color for weekend days, etc.) and alternating row colors.
- The **Selection object** with radio buttons or check boxes now has the new property **Number of Lines per Item**. If this value is greater than 1, each individual radio button or check box occupies more space vertically, and its text is wrapped over multiple lines.

. . . Pivot Table

- Reverse links from the **Pivot Table** are now also updated when the user selects an element in the row or column area.
- The **Pivot Table** has been extended with the option to display numerical values as so-called 'value-bars' (small colored bars).
- The currently sorted column in the **Pivot Table** now shows an indication mark.
- The mouse pointer now changes when hovering over a column side line in a **Pivot Table**, to make it clear that the column can be resized by dragging the line.
- In the **Pivot Table**, you can specify whether or not aggregators should be present for each identifier separately.
- A **Color** property has been added for second level lines in the grid of a **Pivot Table**.
- The **Pivot Table** now supports standard formats for the display of values.
- The **Pivot Table** has a new button that can be used to reload the layout that was stored in the state file.
- The **Pivot Table** has been extended with a reverse link property for the identifier, multiple case and suffix indices.
- In the **Pivot Table**, element text that contains `\n` characters is now displayed over multiple lines when displayed in a non-leaf level in the row tree, or in a leaf level in the column tree.
- The option **Maximum Number of Tree Entries** of the **Pivot Table** can now be controlled through a parameter as well.
- Some properties in the **Default** section of the **Pivot Table** properties now accept indexed identifiers, whereas previously you could only specify scalar references (or they were not available at all in the **Default** section).
- The selection that the end-user makes in a **Pivot Table** can now be stored in an identifier, enabling you to take action in your model based on which cells have been selected.

. . . Solving

- The `GMPOuterApproximation` module has been extended with a procedure that implements an outer approximation algorithm using only a single branch-and-bound tree. It is based on an article by Quesada and Grossmann (1992) and can only be used for convex MINLP problems with binary variables. Currently only **CPLEX** can be used as MIP solver.
- AIMMS has been extended with a link to **Gurobi 1.0**, a new state-of-the-art **LP** and **MIP** solver.
- The **MOSEK** link has been upgraded to version 5.0. **MOSEK 5.0** supports nonlinear convex optimization.
- The `Solver` option settings can now easily be copied from one version of a solver to another version.
- The function `GMP::Solution::GetFirstOrderDerivative` has been added to the language.
- A branch callback procedure can be installed by using the new procedure `GMP::Instance::SetCallbackBranch`. See the Function Reference for more

information.

- There is a new constraint property `IncludeInLazyPool`. If this property is set, **CPLEX** will add the constraint to the pool of lazy constraints.
- There is a new constraint property `IncludeInCutPool`. If this property is set then **CPLEX** will add the constraint to the cut pool.
- Under Linux, if for some **CONOPT** version no options are set in the project, then the option settings from the closest matching **CONOPT** version are used.
- AIMMS now supports filters for filtering the solution pool of a MIP problem (only when using **CPLEX** for solving).
- The function `GMP::Solution::SetMIPStartFlag` has been added to support the **CPLEX 11.2** feature **Multiple MIP Starts**.
- The procedure `GMP::Tuning::SolveSingleMPS` can now be used for any linear solver.
- The performance of first order derivative evaluations by **KNITRO** has been improved. Furthermore, the performance of Hessian evaluations used by **CONOPT** and **KNITRO** has been improved.
- A link to **XPRESS 19** has been added to the **Complete** AIMMS installation.

. . . Databases

- The function `GetDatasourceProperty` and the predeclared set `AllDatasourceProperties` have been added. These give you access to a number of datasource-specific properties, like the name or the version. For details, please refer to the Function Reference.
- Six other new database functions have been added to the AIMMS language:
  - `SQLNumberOfColumns`
  - `SQLColumnData`
  - `SQLNumberOfTables`
  - `SQLTableName`
  - `SQLNumberOfViews`
  - `SQLViewName`

  These functions give (low level) information about database objects.

. . . IT-related

- AIMMS now contains a commandline tool `AimmsCmd.exe`, in the AIMMS `Bin` directory) for running an AIMMS project in end-user mode. Please see the User’s Guide for details.

Issues . . .

- **IMPORTANT:** AIMMS regarded some procedures as `UndoSafe` while in fact they were not (because they were changing global data). Now, such a procedure is not regarded as `UndoSafe`, unless explicitly specified in the **Properties** attribute of the procedure. This improved implementation may have an effect on your application because an **Undo** action in a page object might ’suddenly’ no longer work. The solution is to mark the procedure (upon change or upon select) as `UndoSafe` explicitly.

- **IMPORTANT:** Units now follow the same namespace rules as ordinary identifiers. As a positive consequence: two procedures can now declare both local unit parameters with the same name. As a negative consequence: Some models may not compile anymore because units / unit parameters are declared within modules and referenced without prefix outside the scope of that module. Because of this change, the option `Singleton_unit_namespace` in the category backward compatibility has been added with range { `off`, `on` } and default `off`. When `on`, AIMMS effectively uses only a single unit namespace.
- The suffices `.Violation` and `.DefinitionViolation` now have the same unit as the identifier with which they are associated.
- Using the `MemoryStatistics` function on 64 bits Linux systems, sometimes displayed a large negative value for the number of allocations in the resulting memory log file.
- After de-installing a different AIMMS version than the one you used, the start page links might have stopped working correctly.
- The developer password dialog did not respond correctly to the `--hidden` command line flag. Now it responds by opening the project in end-user mode.
- When using uniquely named ranges in the functions `ExcelCopyRange` and `ExcelClearRange`, and no active sheet has been set for the workbook, no error message about missing an active sheet is yielded anymore.

. . . Cases

- In the **Case Type** and **Data Category** properties, referencing to a model section in another library did not work correctly.
- If, during a **Save Case As**, you tried to overwrite an existing case using a different **Case Type**, then the case was not written using that **Case Type**. Now, you must confirm that you try to change the **Case Type** of the existing case, after which this **Case Type** is used to save the case.

. . . GUI

- The `PageOpen` function now listens to the scope of the procedure in which it is called when searching for the page. So, when `PageOpen` is called from a procedure declared in a library, AIMMS starts searching in that library first.
- Adding set elements in a **Selection** object did not work for adding elements to a subset if this subset was used in the definition of its super set. In order to use this, you need to change the contents to Subset(Subset) instead of Subset(MasterSet).
- If a button was placed inside an **Indexed Page** object, and the actions of that button caused the contents of the **Indexed Page** to change, a fatal application error could occur.
- In the **Gantt Chart** you can now choose to ignore daylight saving time in the calendar.
- The **Network** object now displays nodes of size 0 correctly (i.e. they are not displayed at all). In addition, a node of size 1 is indeed only 1 pixel

wide.

- In many properties of the **Pivot Table** and **ActiveX** objects, it was possible to specify ’reverse index links’ while they were not used by the object. Now, you can only specify it in the relevant locations.
- `PageGetNextInTreeWalk` now has an extra optional argument: `IncludeHiddenPages`. When this argument is set to 1, then the resulting page may be a hidden page. Otherwise, these hidden pages are skipped.

. . . Printing

- A button in a **Navigation** object on a print page was showing a font that was too large.
- An **Indexed Page Object** on a print page sometimes caused a fatal application error.
- The title of the row headers in a **Pivot Table** were not printed correctly if the **Pivot Table** was positioned more to the right side of a print page.
- If a **Pivot Table** on a print page was printed, the outer indices were not always printed at the expected location.

. . . Pivot table

- In the **Pivot Table**, the last selected outer index was not stored in the saved layout, and hence was not restored when reopening the page.
- In the **Pivot Table**, aggregated values could show wrong values after the value of a fixed element in the row or column tree had been changed (by selecting another element from the drop-down list).
- The format option `Scientific notation if too small` also affected large numbers in the **Pivot Table**.

. . . Libraries

- The function `ReferencedIdentifiers` sometimes didn’t take the identifiers into account on the first line of an **Interface**, **Public** or **Protected** attribute.

. . . Solving

- When re-committing an empty complement attribute of a complementarity variable, the old value could re-emerge.
- A math program without a specified type, no integer variables but with an `Sos` constraint could be solved erroneously as an LP (instead of a MIP).
- The two GMP related functions `GMP::Instance::SetCallbackNewIncumbent` and `GMP::Instance::SetCallbackIterations` are now also supported when **CPLEX** uses dynamic search.

. . . Databases

- The database `Read` and `Write` statements did not allow a set name to be used in their column list.

. . . Compiler and Execution

- When the option `Put_number_style` is set to `Fit_field_width`, very large values will no longer be printed in exponential format but as a sequence of asterisks.

- Local identifiers were erroneously accepted in the **Interface**, **public** and **Protected** attributes of **libraries** and **modules**.
- Root sets are now refreshed when loading an active case.
- The option `Warning_read_defined_identifier` did not recognize scalar constant defined parameters and constant defined sets. A constant parameter or set is a parameter or set that does not depend on other parameters or sets that do not have a definition, directly or indirectly.
- When a parameter is emptied and the default of that parameter is outside the bounds, a range check violation is issued.
- When reading or writing an XML file, AIMMS now gives a warning if the string value of a string parameter is longer than the value of the new option `XML_Maximal_String_Length`. By increasing the value of this option it is now possible to read/write longer strings.
- The case of elements generated in an expression like `data { label1 .. label10 }` was influenced if there were other identifiers or elements with the name `LABEL1`. Now, the case of the elements is always as expected.
- The AIMMS compiler erroneously accepted unused keywords such as `from` to be declared as indices.
- When the **Order by** attribute of an indexed set contains a reference to an index that is contained in the index domain of that indexed set, AIMMS now gives an error message.
- Consider a parameter `P` with range `[P_Lo, P_Up]`. When the value of `P` is changed from within a page, the definitions (if any) of `P_Lo` and `P_Up` were not first updated, causing the new value of `P` to be checked against the out-of-date range.
- Some error messages regarding format errors in the function `CurrentToString` have been improved.
- Superfluous error messages could be generated if a model was loaded without compilation, followed by adding/changing identifiers and attributes and invoking the compiler after that.

. . . Model Editor

- When a `.aim` file was created via the source file wizard, comments were not written to the `.aim` file.
- Importing a section could keep a lock on the `.amb` file.
- When changing the type of a parameter into another type of parameter, e.g. from element parameter to string parameter, the data is now converted if possible, or discarded otherwise.
- Exporting a project to a new folder in the root of the disk did not work.
- The wizard dialog that appeared when entering an invalid name for a new identifier in the model tree was rather useless and has been removed.
- Tooltips were not displayed when hovering above a **File** identifier.
- Tooltips were not displayed correctly when XML escape characters were used in the **Comment** or **Text** attributes of identifiers.
- Moving a declaration section from a procedure into a global scope no longer requires a restart of the project in order to make the declarations

globally visible.

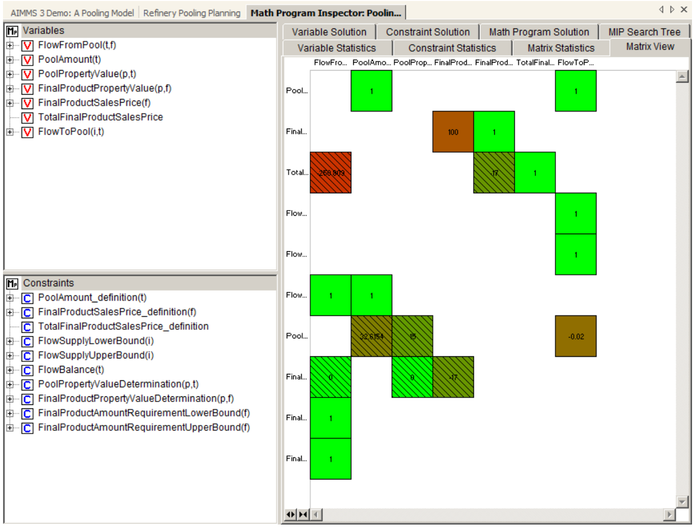
What is new in AIMMS 3.9

What is new in AIMMS 3.9?

This documentation reflects the state of AIMMS version 3.9. Compared to AIMMS 3.8, the following major new and extended functionalities have been added to the system:

Nonlinear Math Program Inspector

The Math Program Inspector (MPI) in AIMMS is now also available for nonlinear models (NLP). The MPI can list the generated nonlinear constraints and several new nonlinear-related statistics. The MPI matrix view can visually indicate matrix blocks that contain nonlinearities (see Figure 4.1), and the MPI also reports the reductions that have been realized by the nonlinear presolver.



Variables
- FlowFromPool(t,f)
- PoolAmount(t)
- PoolPropertyValue(p,t)
- FinalProductPropertyValue(p,f)
- FinalProductSalesPrice(f)
- TotalFinalProductSalesPrice
- FlowToPool(i,t)

Constraints
- PoolAmount_definition(t)
- FinalProductSalesPrice_definition(f)
- TotalFinalProductSalesPrice_definition
- FlowSupplyLowerBound(i)
- FlowSupplyUpperBound(i)
- FlowBalance(t)
- PoolPropertyValueDetermination(p,t)
- FinalProductPropertyValueDetermination(p,f)
- FinalProductAmountRequirementLowerBound(f)
- FinalProductAmountRequirementUpperBound(f)

	FlowFro...	PoolAmo...	PoolProp...	FinalProd...	FinalProd...	TotalFinal...	FlowToP...
Pool...		1					1
Final...				100	1		
Total...	-268.909				17	1	
Flow...							1
Flow...							1
Flow...	1	1					
Pool...		-32.6154	15				-0.02
Final...	0		0	-17			
Final...	1						
Final...	1						

Figure 4.1: Example of Matrix View in Nonlinear MPI

The MPI enables you to drill down into your nonlinear programs to analyze the causes of infeasibilities, unrealistic results and other problems associated with solving your math program.

New CPLEX and AOA options

The following new features of the CPLEX solver are accessible from AIMMS:

- Storing multiple MIP solutions with solution pool filters
- Lazy constraints
- User-cut pool
- Branching callbacks
- Multiple MIP Starts

The power of branching callbacks will be illustrated with a new, advanced variant of the AIMMS Outer Approximation (AOA) algorithm, based on the approach of Quesada & Grossman (1992). This approach uses branch-and-cut in combination with outer approximation, and often solves convex models with binary variables much faster than other algorithms.

AIMMS enables you to use advanced CPLEX 11 options to develop specialized applications without the need to write complicated code or solver files.

New Solver: Gurobi

AIMMS 3.9 supports the new Gurobi solver. Gurobi is a brand new high-performance optimization solver for linear programming (LP) and mixed integer programming (MIP). The Gurobi MIP solver allows for shared memory parallelism to take full advantage of modern multiple cores and multiple sockets architectures.

New MOSEK version

AIMMS 3.9 supports the latest version of the MOSEK solver, version 5. The MOSEK 5 solver can be used with AIMMS to solve LP, QP, QCP, MIP, MIQP and MIQCP models, and now also convex NLP models.

Additional database functionality

A number of database functions have been introduced, which give the AIMMS user access to detailed database information, like the number and names of all tables in a database and the details of all columns in a table.

New features in the IDE

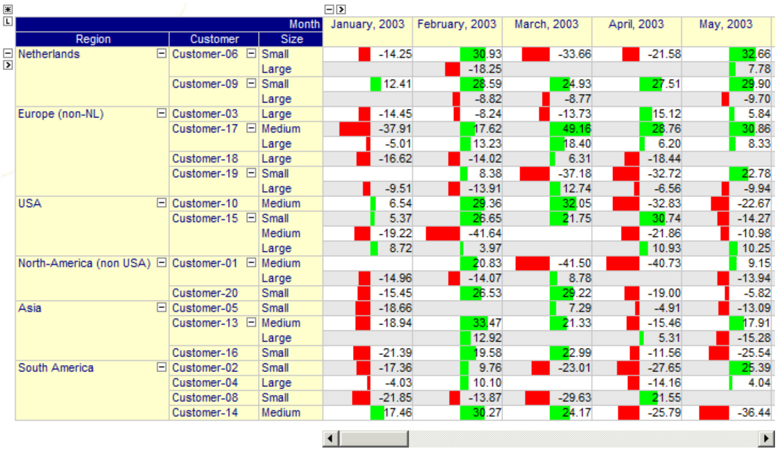
The following new features have been added to the Integrated Development Environment (IDE).

- The AIMMS debugger can be activated automatically on the line where a run-time error occurs, helping you to locate and fix problems quickly during application development.
- The Cross Library Reference dialog box offers a feature to manage preferred and non-preferred references between projects, and it allows for the printing of all cross references.
- The new Identifier Info dialog box offers an easy way to quickly jump to the declaration, attributes or data of an identifier based on its name.

Enhanced Pivot Table functionality

The Pivot Table functionality is enhanced extensively, making it an even more versatile object to use in your GUI:

- You can display cell values as small horizontal bars in the Pivot Table (see Figure 4.2).
- You can show multiple line element text in the Pivot Table.
- You can specify whether or not aggregators should be present for each identifier separately.
- You can use indexed references and identifiers that are indexed over the set AllIdentifiers to specify the default properties for identifiers in a Pivot Table.



		Month	January, 2003	February, 2003	March, 2003	April, 2003	May, 2003
Region	Customer	Size					
Netherlands	Customer-06	Small	-14.25	30.93	-33.66	-21.58	32.66
		Large		-18.25			7.78
	Customer-09	Small	12.41	28.59	24.93	27.51	29.90
		Large		-8.82	-8.77		-9.70
Europe (non-NL)	Customer-03	Large	-14.45	-8.24	-13.73	15.12	5.84
	Customer-17	Medium	-37.91	17.62	49.16	28.76	30.86
		Large	-5.01	13.23	18.40	6.20	8.33
	Customer-18	Large	-16.62	-14.02	6.31	-18.44	
	Customer-19	Small		8.38	-37.18	-32.72	22.78
		Large	-9.51	-13.91	12.74	-6.56	-9.94
USA	Customer-10	Medium	6.54	29.36	32.05	-32.83	-22.67
	Customer-15	Small	5.37	26.65	21.75	30.74	-14.27
		Medium	-19.22	-41.64		-21.86	-10.98
		Large	8.72	3.97		10.93	10.25
North-America (non USA)	Customer-01	Medium		20.83	-41.50	-40.73	9.15
		Large	-14.96	-14.07	8.78		-13.94
	Customer-20	Small	-15.45	26.53	29.22	-19.00	-5.82
Asia	Customer-05	Small	-18.66		7.29	-4.91	-13.09
	Customer-13	Medium	-18.94	33.47	21.33	-15.46	17.91
		Large		12.92		5.31	-15.28
	Customer-16	Small	-21.39	19.58	22.99	-11.56	-25.54
South America	Customer-02	Small	-17.36	9.76	-23.01	-27.65	25.39
	Customer-04	Large	-4.03	10.10		-14.16	4.04
	Customer-08	Small	-21.85	-13.87	-29.63	21.55	
	Customer-14	Medium	17.46	30.27	24.17	-25.79	-36.44

Figure 4.2: Example of Horizontal Bars in the Pivot Table

- End-Users can restore or reset the Pivot Table state.

Other enhancements in the GUI

The following features in the Graphical User Interface (GUI) have been enhanced:

- GIS Link: you can obtain the background for network objects also from Microsoft Virtual Earth server. The Geocoding functionality will be reintroduced, allowing you to retrieve latitude/longitude coordinates automatically.
- Gantt Chart: you can give rows alternating background colors, and the background of specific periods along the x-axis can be colored individually (see Figure 4.3).

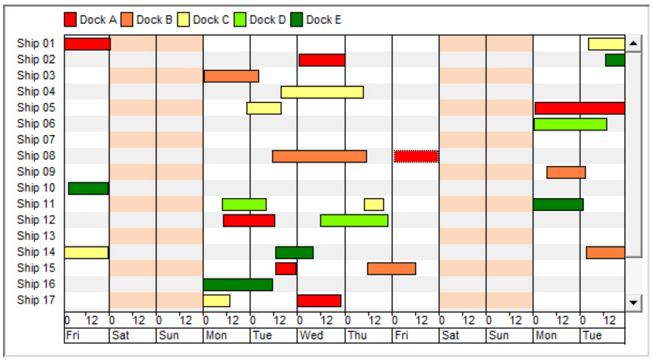


Figure 4.3: Example of alternating colors and vertical bands in the Gantt Chart

- Pie Chart: you can automatically calculate and display the percentages of all pie slices.

- Excel: End-Users can copy data more easily from an Aimms GUI to Excel, using a new function to copy data from a table, composite table or pivot table to a specific range in an Excel sheet.