

# AIMMS 3.14

---

*Release Notes for Win32 Build 3.14 PR - SU2*

Visit our web site [www.aimms.com](http://www.aimms.com)  
for regular updates

AIMMS

---

November 19, 2014

# Contents

<b>Contents</b>	<b>2</b>
<b>1 System Requirements</b>	<b>3</b>
1.1 Hardware and operating system requirements	3
1.2 ODBC and OLE DB database connectivity issues	4
1.3 Viewing help files and documentation	5
<b>2 Installation Instructions</b>	<b>7</b>
2.1 Installation instructions	7
2.2 Solver availability per platform	8
2.3 AIMMS 3 licensing	9
2.3.1 Personal and machine nodelocks	11
2.3.2 Installing an AIMMS license	13
2.3.3 Managing AIMMS licenses	15
2.3.4 Location of license files	17
2.4 OpenSSL license	19
2.5 Conversion of projects developed in AIMMS 3.13 and before	21
<b>3 Getting Support</b>	<b>23</b>
3.1 Reporting a problem	23
3.2 Known and reported issues	24
<b>4 Release Notes</b>	<b>26</b>
What's new in AIMMS 3.14	26
AIMMS 3.14 PR - SU2, release date 2014-11-19	30
AIMMS 3.14 PR - SU1, release date 2014-07-18	31
AIMMS 3.14 PR, release date 2014-03-05	33
AIMMS 3.14 FR2, release date 2013-10-16	35
AIMMS 3.14 FR1, release date 2013-07-12	37

# Chapter 1

## System Requirements

This chapter discusses the system requirements necessary to run the various components of your Win32 AIMMS 3.14 system successfully. When a particular requirement involves the installation of additional system software components, or an update thereof, the (optional) installation of such components will be part of the AIMMS installation procedure.

*System requirements*

---

### 1.1 Hardware and operating system requirements

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

*Hardware requirements*

- 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.

The Win32 version of AIMMS 3.14 is designed to run under

*Supported Windows versions*

- Windows Server 2003,
- Windows Vista,
- Windows 7,
- Windows 8,
- Windows Server 2008, and
- Windows Server 2012.

Running the Win32 version of AIMMS 3.14 versions under the 32-bit Windows versions 95/98/ME/2000/NT 4.0/XP is not supported.

---

## 1.2 ODBC and OLE DB database connectivity issues

Before you can start using the database connectivity features built into the Win32 version of AIMMS 3.14, the following ODBC and OLE DB components need to be installed on your computer:

*Database connectivity*

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

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

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

*Microsoft Data Access Components*

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

*Data Access Components installation*

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

*Oracle ODBC drivers and OLE DB providers*

most recent version of the ODBC drivers and OLE DB providers for Oracle 8 and higher from the Oracle web site [www.oracle.com](http://www.oracle.com).

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

*Prefer native Oracle over MDAC drivers*

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

*OLE DB issues*

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

As AIMMS 3.14 and higher don't offer separate ASCII and Unicode versions anymore, it's important to work with the latest versions of the various ODBC drivers/OLE DB providers for your particular database. Specifically, in order to read and write multi-byte characters from and to MySQL databases, you have to make sure that you are using the Unicode version of MySQL's ODBC driver (i.e. the one without 'ansi' in its installer name, and with 'w' or 'unicode' in its actual driver name). For SQLite databases, next to using the latest ODBC driver, make sure that you are using the latest version of the database engine itself in order to use multi-byte characters.

*Unicode/UTF8 requirements*

---

### 1.3 Viewing help files and documentation

The AIMMS 3 User's Guide, the AIMMS 3 Language Reference, the AIMMS 3 Function Reference, and the AIMMS book on Optimization Modeling are available online as Adobe Portable Document Format (PDF) files. In order to view

*AIMMS documentation*

or print PDF documents, Adobe Acrobat Reader version 4.0 or higher needs to be installed on your computer.

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

*Acrobat Reader  
installation*

# Chapter 2

## Installation Instructions

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

*Installation instructions*

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

The latter is the standard procedure for obtaining and installing AIMMS 3 Software Updates. 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

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

*The AIMMS 3 CD-browser*

- start the AIMMS 3 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).

The download area of our web site [www.aimms.com](http://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.

*Obtaining AIMMS 3 from the web*

The AIMMS 3 setup program will guide you through the various steps that are necessary to successfully install AIMMS 3 on your computer. The AIMMS 3 setup program requires 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 installing the prerequisites.

*The AIMMS 3  
setup program*

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

*Administrative  
privileges  
required*

- the installation of the device drivers required to access the AIMMS hardware dongle (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.

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)

*Component  
selection*

---

## 2.2 Solver availability per platform

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

*Solvers in  
installation*

Please note that the following solvers require a computer with a CPU that supports the SSE2 instruction set: GUROBI, MOSEK 6 or higher, CPLEX 12.2 or higher, and CPOPT 12.2 or higher.

*SSE2*

CBC and IPOPT are open source solvers made available in the AIMMS distributable as solver DLL and through COIN-OR ([www.coin-or.org](http://www.coin-or.org)) as solver DLL and in source code format. Please check the 'OS Solvers' section in the License Agreement before using these open source solvers with AIMMS.

*Open Source  
Solvers*



Solver	Version	Win 32	Win 64	Linux 64
AOA		✓	✓	✓
BARON	7.5.3	✓	-	-
BARON	12	✓	✓	-
CBC	2.7.5	✓	✓	✓
CBC	2.8.0	✓	✓	✓
CONOPT	2.071 C	✓	-	-
	3.11 B	✓	-	-
	3.14 A	✓	-	-
	3.14 G	✓	-	-
	3.14 M	-	✓	-
	3.14 Q	-	-	✓
	3.14 V	✓	✓	✓
CPLEX	10.1	✓	✓	✓
	11.2	✓	✓	✓
	12.1	✓	✓	✓
	12.3	✓	✓	✓
	12.4	✓	✓	✓
	12.5	✓	✓	✓
	12.5.1	✓	✓	✓
	12.6	✓	✓	✓
CPOPT	12.2	✓	✓	✓
	12.4	✓	✓	✓
	12.5	✓	✓	✓
	12.6	✓	✓	✓
GUROBI	4.6	✓	✓	✓
	5.0	✓	✓	✓
	5.1	✓	✓	✓
	5.5	✓	✓	✓
	5.6	✓	✓	✓
IPOPT	3.11	✓	✓	✓
KNITRO	6.0	✓	✓	✓
	7.0	✓	✓	✓
	8.0	✓	✓	✓
	8.1	✓	✓	✓
	9.0	✓	✓	✓
LGO	1.0	✓	-	-
MINOS		✓	✓	✓
MOSEK	6	✓	✓	✓
PATH	4.4a	✓	-	-
	4.6	✓	-	-
SNOPT	6.1	✓	✓	✓
	7.2	✓	✓	✓
XA	13	✓	-	-
	14	✓	✓	✓
	15	✓	✓	✓

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

---

## 2.3 AIMMS 3 licensing

AIMMS offers the following two types of licenses:

AIMMS 3  
licensing

- single-user licenses, and
- network licenses.

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

Single-user licenses can be used by a single user on a single computer. To enforce the single-user character, AIMMS 3.14 requires that single-user licenses be protected by either

*Single-user  
license  
protection*

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

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

*You can choose*

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

*Pros and cons of  
dongles*

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

*Pros and cons of  
nodelocks*

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

*Physical  
shipments*

If you request your license to be protected by a dongle, an AIMMS dongle is sent to you along with your AIMMS 3 CD-ROM. If you upgrade from AIMMS 2.20, you can continue to use your existing green Activator dongles used by AIMMS 2.20, *but only with the Win32 version of AIMMS*. The green Activator as well as the grey Sentinel dongle must be connected to the parallel port of

*AIMMS dongles*

your computer. The purple Sentinel dongle must be connected to a USB port of your computer. The AIMMS 3.14 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>.

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

*Network  
licenses*

---

### 2.3.1 Personal and machine nodelocks

AIMMS offers two types of nodelocks:

- personal nodelocks, and
- machine nodelocks.

*Two types of  
nodelocks*

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

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

*Personal  
nodelock*

- Personal nodelocks can be transferred to another computer 3 times per 24 hours. This allows you to take your AIMMS license home in the evening and back to work the next morning without any problems.
- Personal nodelocks have a limited lifetime of 60 days, and should be renewed within that period to extend the lifetime to its full 60-day period. If the nodelock is not renewed within its 60-day lifetime, this does not invalidate your AIMMS license in any way—you only have to renew your nodelock prior to being able to use your AIMMS system again. Note that the renewal process does not require any manual intervention, as AIMMS will try to automatically connect to our internet license database to renew your nodelock once every day you are using AIMMS.
- Both activation and nodelock renewal of personal nodelocks require a working connection to the internet. As a consequence, in the absence of an internet connection you can continue to work uninterrupted for

a period of 60 days, before an internet connection is required to renew your nodelock.

- With every activation or nodelock renewal AIMMS will also update your license files if new license files are available (e.g. if your system is in maintenance), and will inform you of any messages that are available for you in our database.
- Because of their volatile nature, PDT will replace a personal nodelock without any questions asked in case of loss of or damage to your computer.
- You can switch your personal nodelock to a machine nodelock at any time.

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

*Machine  
nodelock*

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

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

*Privacy*

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

*Internet  
connection and  
proxy settings*

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

*Automatic configuration scripts*

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

*Manual proxy setting*

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

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

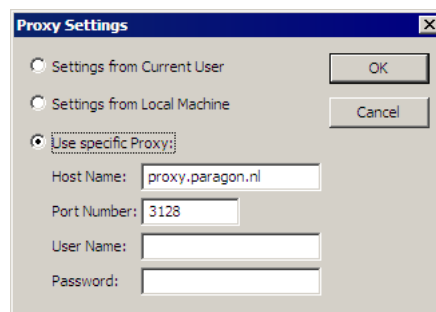


Figure 2.1: The AIMMS **Proxy Configuration** dialog box

---

### 2.3.2 Installing an AIMMS license

When you start up AIMMS 3.14 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.

*Managing your AIMMS licenses*

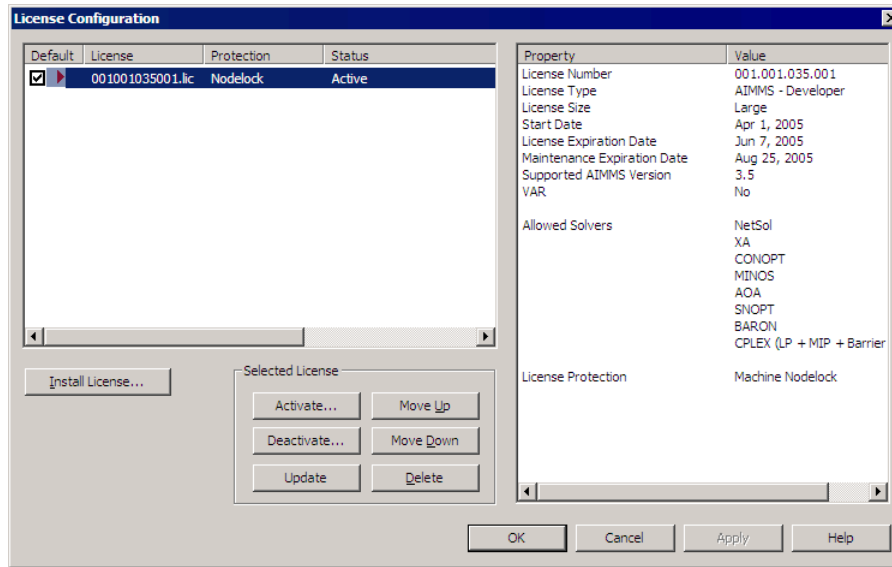


Figure 2.2: The **License Configuration** dialog box

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

*Installing a new license*

- existing AIMMS 3.13 licenses,
- nodelocked licenses,
- dongled licenses,
- network licenses,
- evaluation licenses, and
- student licenses.

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

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

*Single-user nodelocked licenses*

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

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.

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

*Single-user  
dongled licenses*

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

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

*Network  
licenses*

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

To install an AIMMS evaluation license you need the following information

*Evaluation  
licenses*

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

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

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

*Student licenses*

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

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

---

### 2.3.3 Managing AIMMS licenses

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

*Managing  
multiple AIMMS  
licenses*

- you have requested a trial license for a new AIMMS version which you want to run next to your existing license,
- you have temporarily borrowed or hired an AIMMS license with more capabilities than your regular license,

- your system administrator has created multiple network license profiles, each of which you may want to use to run AIMMS.

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

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

*Default licenses*

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

*Transferring licenses*

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

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

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

*Emergency nodelocks*



---

### 2.3.4 Location of license files

AIMMS keeps its license and configuration files in the folder

AIMMS

*Location of  
license files*

of the common application area of your computer. On 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 AIMMS folder contains three subfolders

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

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

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

*Do not move  
nodelock files*

You can specify whether the license and solver configuration that AIMMS uses is the same for any user of the machine, or different for each individual user. To prevent problems when running AIMMS as part of a computer service, AIMMS 3.14 will by default use the same configuration for any user. To modify this behavior, you should edit the file UserDistinction.cfg in the common Config folder. In the file UserDistinction.cfg.default, straightforward directions are given on how to edit it. If no UserDistinction.cfg file exists, AIMMS will use the UserDistinction.cfg.default file instead. You can use this file as a base for setting up your own configuration.

*User specific  
configuration*

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

*Any User*

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

*Current User*

```
C:\Document and Settings\\Application Data\AIMMS
```

On Windows Vista and Windows 7 it is:

```
C:\Users\\AppData\Roaming\AIMMS
```

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

*Accessing  
configuration  
files*

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

*Version  
dependent  
configurations*

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

*Network License  
Client Files*

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

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

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

*Project  
dependent  
configuration*

---

## 2.4 OpenSSL license

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 included - in separate RPC installation only*

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

*OpenSSL license*

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:  
"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact [openssl-core@openssl.org](mailto: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 ([eyay@](mailto:eyay@))

cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

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

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

This library is free for commercial and non-commercial use as long as the following conditions are adhered 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.]

---

## 2.5 Conversion of projects developed in AIMMS 3.13 and before

AIMMS 3.13 and previous releases came in two flavors: a single- and a double-byte flavor for handling strings and file IO. These flavors were also called the ASCII and Unicode flavors of AIMMS. The ASCII flavor was targeted at locales where an ASCII based code page was customary, such as the Americas and Europe. The Unicode flavor was targeted at locales where Unicode was customary to handle a large number of characters, such as Asia. The important distinction between the two flavors was the character encoding used for files. Character encodings are briefly explained on Page 18 in the Language Reference. Making this distinction between the flavors, led to the following problems:

*Unifying AIMMS flavors*

- A first problem was that converting a project from one flavor to the other was cumbersome.
- A second problem was that on the Linux platform, only the ASCII flavor was available.
- A third problem was that exchanging of files between projects of different flavors was cumbersome.
- A fourth problem was that the input of two files with different (but ASCII-based) encodings was very cumbersome.

AIMMS 3.14 is designed to overcome these problems by no longer making a distinction between the ASCII and Unicode flavor of AIMMS, by allowing Unicode input on Linux, and by selecting character encodings on a per file basis.

In order to convert AIMMS projects developed by either of the flavors of AIMMS 3.13 and older, such that the converted project will operate similar to the original one, AIMMS 3.14 will set a variety of options in the converted project. These option settings are conservative; they are geared towards compatibility and not towards the use of UTF8. UTF8 is the encoding preferred for the exchange of files between different locales. In order to take advantage of the encoding capabilities of AIMMS 3.14 and the ease offered by the UTF8 encoding for the exchange of files between different locales, these option settings warrant some consideration:

*Option settings created during conversion*

- **aim\_character\_encoding** Input (.aim) files are both read in and written out using this encoding. It will be set based on your current locale. We

suggest to convert your .aim files, if any, to UTF8 and then reset this option to UTF8.

- **ascii\_case\_character\_encoding** The encoding used to decode strings in case files created by the ASCII flavor of AIMMS 3.13 and older. It will be set based on your current locale. We suggest to keep this option as set, as new case files are marked as using UTF8 encoding and thus immune against any change of locale.
- **default\_input\_character\_encoding** The encoding used by the statement `read from file` and other statements that input text files, except for .aim files. It will be set based on your current locale. The best setting of this option depends on the developments in the deployment environment. We suggest to switch to using UTF8 encoding for the exchange of files. Note that a FILE has an encoding attribute that permits the selection of encoding to be used on a per file basis.
- **default\_output\_character\_encoding** The encoding used during the statements `PUT` and `write to file` and other statements that output text files, except for .aim files. It will be set based on your current locale for converted ASCII projects and to UTF-16LE for converted Unicode projects. The best setting of this option depends on the developments in the deployment environment. We suggest to switch to using UTF8 encoding for the exchange of files. Note that a FILE has an encoding attribute that permits the selection of encoding to be used on a per file basis.
- **external\_string\_character\_encoding** The encoding used to pass string parameters in external function calls. It will be set to the current locale for converted ASCII projects and to UTF-16LE for converted Unicode projects.
- **external\_dll\_aimms\_reference** This option is an indicator whether string communication of existing external libraries linked against `libaimms.dll` of AIMMS 3.13 is based on `char` or `wchar_t` characters.

Note that when a file has a byte order mark at the beginning, this byte order mark overrides the encoding used.

AIMMS 3.14 supports both the multi-byte API, whereby the API functions are suffixed with an `A`, and the wide char API, whereby the API functions are suffixed with a `W`. See also the Language Reference, page 572. For external functions and procedures calling back to AIMMS, which C++ character type is used, can be indicated by the option `external_dll_aimms_reference`. Otherwise, this needs to be indicated by rebuilding your executable. In other words, we have the following situation:

- Existing executables (.exe files), linked against `libaimms.dll`, will need to be rebuilt, with linking against `libaimms3.dll`.
- Existing dynamic link libraries (.dll files), linked against `libaimms.dll` can be reused by AIMMS 3.14; you do not need to adapt your external libraries. However, if you do adapt the source code of your library and want to rebuild it, you will now need to link it against `libaimms3.dll`.

*External  
executable  
building not  
converted*

# Chapter 3

## Getting Support

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

*Getting support*

---

### 3.1 Reporting a problem

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

*Problem fixing*

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

*Reporting a problem*

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

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

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

*Problem processing*

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

*Modeling support*

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

*Automatic Error Reporting*

---

## 3.2 Known and reported issues

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

*Known issues*

- The AIMMS Excel Add-In does not yet function well enough in the AIMMS 3.14 Feature Release 1. Therefore, we advise you not to use it in combination with this release. We will do our best to improve this situation.

*Excel Add-In*



- 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.

*Windows*

*Vista/Windows 7*

# Chapter 4

## Release Notes

This chapter contains a description of the new features developed for AIMMS 3.14. In addition, it contains the release notes of all AIMMS 3.14 releases. For each release, the following items are listed:

*Release notes*

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

Up to AIMMS 3.13, the installation procedure installed either the single-byte or the double-byte AIMMS executable. Starting with AIMMS 3.14, only a single version is available, which supports all possible character encodings.

*AIMMS 3.14 versions*

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

---

### What's new in AIMMS 3.14

This documentation reflects the state of the second AIMMS 3.14 Feature Release. Compared to the AIMMS 3.13 Production Release, the following major new and extended functionalities have been added to the system:

For **AIMMS 3.14 FR2**, the new features/extended functionalities are:

*What's new in AIMMS 3.14 FR2?*

- Indexed sets allowed in element parameter's/variable's range
- Multi-variants of several GMP update procedures
- Solver updates

A small change in the AIMMS language may make significant impact throughout the system as illustrated by the `range` attribute of element parameters and element variables. This attribute may now be an indexed set. The impact is best explained by means of an example.

*Indexed sets allowed in element parameter's range*

Consider the states of the USA, and finding a route through these states. From state `s`, in a step you'll go to an adjacent state `nextState(s)`. The set of adjacent states is the indexed set `adjacentStates(s)`.

The difference in declaration is rather small; only contents in the range attribute change.

Previous AIMMS versions:

```
ELEMENT PARAMETER:
  identifier : oldNextState
  index domain : s
  range      : USAStates
```

AIMMS 3.14 FR2:

```
ELEMENT PARAMETER:
  identifier : nextState
  index domain : cs
  range      : AdjacentStates(cs)
```

The modeling language will ensure that any assignment to `nextState(s)` is not only a state in the set `USAStates`, but also in the set `AdjacentStates(s)`. However, consider a data page for these two identifiers, and selecting an element for the next state of Connecticut. With the declaration of `oldNextState`, you'll be offered all states, whereas with the declaration of `nextState`, you'll only be offered the relevant choices:

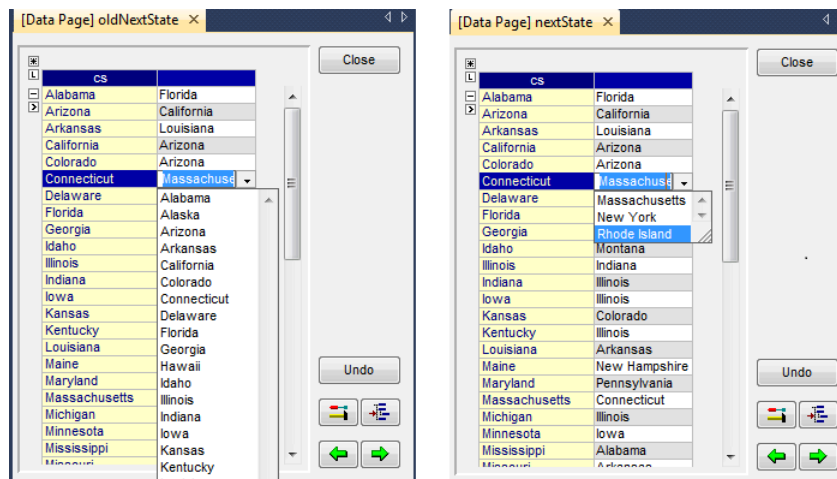


Figure 4.1: Selection in previous AIMMS versions vs. AIMMS 3.14 FR2

Moreover, once a selection is made, with the declaration of `oldNextState`, you should add code to check whether a proper selection has been made by the user. With the declaration of `nextState`, this is ensured from within AIMMS and you do not have to add code to check this yourself. Finally, changing `nextState` into an element variable and optimizing, AIMMS will automatically add the constraint `nextState_range(s)` with definition `nextState(s) in AdjacentState(s)`.

Multi-variants of several GMP update procedures have been added, which will be more efficient if much data for columns and/or rows has to be updated.

These new procedures are:

GMP::Coefficient::SetMulti

GMP::Column::FreezeMulti

GMP::Column::UnfreezeMulti

GMP::Column::SetLowerBoundMulti

GMP::Column::SetUpperBoundMulti

GMP::Row::SetRightHandSideMulti

*Multi-variants  
of several GMP  
update  
procedures*

In this second Feature Release of AIMMS 3.14, a number of solver updates are included.

*Solver updates*

- There is a new version of **CBC**, i.e. **CBC 2.8.0**. This new version fixes several issues including an issue where **CBC 2.7.5** returned 'infeasible' for a feasible MIP model.
- There is a new version of **CPLEX**, i.e. **CPLEX 12.5.1**.
- There is a new version of **BARON**, namely version **12**. The new version does better memory management and reduces memory requirements for large or difficult problems. **BARON 12** automatically sets many options based on problem characteristics and learning algorithms and, as a consequence, many options have been eliminated.
- There is a new version of **KNITRO**, i.e., **KNITRO 8.1**.
- There is a new version of **IPOPT**, i.e. **IPOPT 3.11**.

---

For **AIMMS 3.14 FR1**, the new features/extended functionalities are:

- UTF8 and other encodings
- Modern Styling
- Multiple Text Lines on Buttons
- Automatic Benders
- Gurobi Solver

*What's new in  
AIMMS 3.14  
FR1?*

In order to further the internationalization of AIMMS projects, AIMMS 3.14 projects can now be exchanged between locales without change; the standard text encoding UTF8 is used for all files. The distinction between the ASCII and Unicode flavor of AIMMS is lost; AIMMS 3.14 is only provided in the UTF8 flavor. With this change, AIMMS follows the trend of adopting UTF8 by many software vendors, which eases the communication between AIMMS and other programs. To enable data exchange with legacy programs, AIMMS is able to read in/write to files using various other encodings, allowing the encoding to be selected on a per file basis. The conversion of both ASCII AIMMS 3.13 and Unicode AIMMS 3.13 projects to UTF8 AIMMS 3.14 projects is automated; for details see Section [2.5](#).

*UTF8 and other  
encodings*

As a follow-up on the completely new IDE, introduced in AIMMS 3.13, we now completed the new look by applying the Visual Styles on many of the common controls that are used throughout AIMMS, such as buttons, check boxes, tab strips, dialog boxes, etc. These controls are now displayed according to the style that is dictated by the Windows version on which AIMMS is running. Slight drawback is that it is much harder and sometimes impossible, to style a page object to your specific preferences.

*Modern Styling*

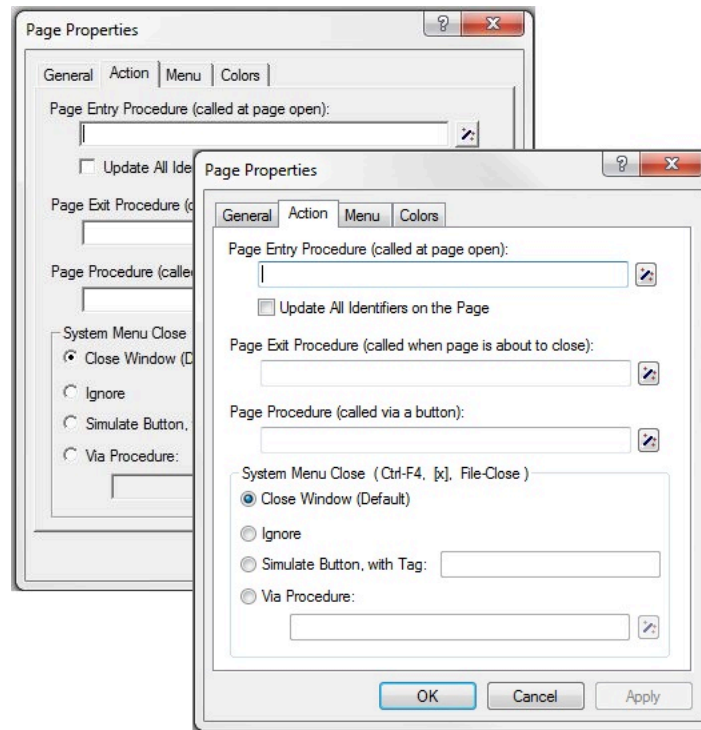


Figure 4.2: The old and new styling of AIMMS

The button object has been changed in the way it displays text that is too long to fit within its width. In previous versions, the text was simply cut-off, but in this new version the text will automatically be wrapped to the following line.

*Multiple Text Lines on Buttons*

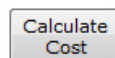


Figure 4.3: Multiple text lines on a button

Benders decomposition is a solution method for solving certain difficult optimization problems. Instead of considering all decision variables and constraints of a problem simultaneously, Benders decomposition partitions the problem into multiple smaller problems and solves them iteratively, which can sometimes be more efficient. The new Automatic Benders module is easy to use and implements several variants of the Benders decomposition algorithm. The implementation is an open algorithm which can be customized in order to obtain better performance and/or a better solution for a particular problem.

*Automatic  
Benders*

There is a new version of **GUROBI**, i.e. **GUROBI 5.5**. **GUROBI 5.5** includes a new automatic option tuning tool. **GUROBI 5.5** introduces the concurrent MIP solver which performs several independent MIP solves in parallel, reporting the best results obtained by any of them. **GUROBI 5.5** can use one or more Gurobi Compute Servers for its computations. A Gurobi Compute Server license is required. This functionality is only available for the 'link-only' Gurobi license.

*Gurobi Solver*

---

### **AIMMS 3.14 PR - SU2, release date 2014-11-19**

In this second Software Update of the AIMMS 3.14 Production Release, the following issues have been fixed or improved.

*3.14 PR - SU2  
2014-11-19*

- **CPLEX 12.6** and **CP Optimizer 12.6** have both been upgraded to version 12.6.0.1.
- Unfortunately, **Yahoo** has stopped offering their Yahoo Maps service. For the **GIS Network** object in AIMMS, this means that you can not select maps from Yahoo Maps anymore.
- If two different pages had the same user defined menubar, but a different toolbar, the keyboard shortcuts in the menubar could disappear.
- Conversion of the old data management style file (the .dat file) skipped cases, datasets and/or folders that had invalid characters in the name (like '/').
- The text in the dialog to convert a Data Manager file (from the menu command **File-Open-Data File...**) has been simplified.
- In a multiple case view of a page object, the names of the displayed cases are now equal to only the base name of the underlying .data file. In other words: the folder path and the extension .data are removed. This only applies to the Data Management style 'Using disk files and folders'.
- If the **Network License Server** exited unexpectedly, AIMMS only tried to reconnect to the same License Server and not to any other License Servers in the license configuration.
- In some specific situations, the pasted text in a **Scalar** object was not applied to the underlying identifier.

*Improvements*

*Issues ...*

- Spreadsheet::RetrieveTable sometimes did not work properly when its arguments involved local sets.
- The procedure WriteXML could write superfluous data when using a write filter.
- If the CPLEX option Deterministic\_time\_limit was set and CPLEX actually hit this time limit, the program and solver status passed to AIMMS were incorrect.
- A **Pivot Table** object could not set a reverse link to an identifier in a library because of its prefix.
- Committing a local numeric parameter with a non-atomic unit after initial compilation could result in an erroneous scaling of the initial value.
- AIMMS could produce a severe internal error after the error "The limit of 32700 Identifiers is exceeded." was issued.
- When two or more defined parameters and sets were referencing each other, AIMMS could go into an infinite loop determining whether or not these definitions were constant.
- Semi-continuous variables were handled incorrectly when doing a GMP solve on a model without integer variables.
- There was a performance problem with inline variables with a complicated index domain expanded in high-dimensional variables. This could lead to an unnecessary re-evaluation of the index domain while generating the high-dimensional constraint.
- The **Progress Window** was not updated during a solve if **BARON** was used.
- A small improvement has been done on the display of disabled buttons that have a non-default color.
- An expression like FormatString("5.0n", 1.5) resulted in a rounding warning.
- A statement like Write anIndexedSet to file filtering i in someSubSet; would ignore the filtering altogether.
- Empty index sets could lead the function DisAggregate with a locus set into an infinite amount of memory allocated.

---

### AIMMS 3.14 PR - SU1, release date 2014-07-18

In this first Software Update of the AIMMS 3.14 Production Release, the following issues have been fixed or improved.

*3.14 PR - SU1  
2014-07-18*

- **Gurobi 5.6** has been upgraded to version 5.6.3.
- The option read\_parameter\_file has been added for **CPLEX 12.6**. This option can be used to read in a file with **CPLEX** option (parameter) settings.

*Improvements*

- Expressions of the form `i IN someIndexedSet( someElemPar( i ) )` could return erroneous results. *Issues ...*
- **CPLEX 12.5** and higher can now resume a GMP solve that was interrupted or had hit a time, node or iteration limit.
- Written XML data could be incorrect for identifiers depending on an ordered set.
- The option `database_translate_0_to_zero` is made more symmetric by using it to write to a database as well; 0.0 numeric values are now translated in the database to NULL instead of 0. Please check the Help as well for options `database_translate_NULL_to_0`, and `database_translate_0_to_zero`.
- `CaseCompareIdentifier` did not handle the comparison of special values correctly.
- AIMMS could issue a superfluous warning on rounding ambiguity when the units of the arguments involve parametric units.
- The Indexed Element parameter range was not always updated at the right time.
- `StringToMoment` now searches for the unit in the unit symbol space from the perspective of the procedure it is called.
- Consider the following declarations:

```

SET:
  identifier : S
  index      : i, j ;
SET:
  identifier : IS
  index domain : i
  subset of   : S ;
PARAMETER:
  identifier : P
  index domain : (i, j in IS(i) )
  range       : S
  definition  : j-1

```

In the definition attribute of P, j was limited to IS(i) but was not bound to it, resulting in the lag operation -1 being applied with respect to the set S instead of the set IS(i).

- Consider  $A(i, j)$ ,  $i$  in S,  $i_s$  in subS subset of S, and the statement `EMPTY A(i_s, j_s)`. AIMMS could forget to update the definition of subS before executing `EMPTY A(i_s, j_s)`.
- Deleting at runtime, identifiers of type indexed set could lead to severe internal errors.
- The upper bound for the option `warning_round_tolerance` is lowered from 0.5 to 0.1 in order to avoid considering the value 0.5 to be the integer 1 during the writing of this option value to the option file.
- When AIMMS raises an error, the message is traced to the **log/aimms.err** file. This file is opened using the default output character encoding. When determining this default output character encoding results in an error, this could lead to an indefinite recursion and subsequent stack overflow.



- The option `progress_awareness` has been added for **CPLEX 12.6** to make AIMMS more aware of progress updates. AIMMS could become unresponsive while solving a difficult problem or during solution polishing.
- After `maximal_number_of_execution_errors_reported` errors reported, AIMMS did not get into an On Error block. Now this option only applies to the global error handler.
- **Scalar** objects with a multi-line edit field, no longer shows the default popup menu if a user popup menu has been defined.
- Scrolling the **Gantt Chart** calendar could end up in an endless loop.

---

### AIMMS 3.14 PR, release date 2014-03-05

In this AIMMS 3.14 Production Release, the following issues have been fixed or improved.

3.14 PR  
2014-03-05

- **CPLEX 12.6** has been added. **CPLEX 12.6** can find global solutions for non-convex quadratic programs (QP and MIQP).
- The function `TestDate` now has a new optional argument `requireUnique`.
- The option `auxiliary_root_threads` has been added for **CPLEX 12.5** and higher.
- There is a new version of **GUROBI**, i.e. **GUROBI 5.6**.
- **CP Optimizer 12.6** has been added. **CP Optimizer 12.6** can use failure-directed search.
- **KNITRO 9.0** has been added. **KNITRO 9.0** introduces the Tuner which can be used to test a variety of different option settings and automatically suggest non-default user option settings for a user model for improved performance.
- The AIMMS presolver can now linearize more quadratic constraints. This is controlled by the option `reformulate_constraints` which has a new default setting of `'multi'`.
- User authentication properties (username and password) were added to the GIS background of the **Network** Object for Yahoo! Maps, Bing Maps and OpenStreetMap.
- The functions `DirectoryGetFiles` and `DirectoryGetSubDirectories` filtered out files and directories with the `system` bit set. They now both have an additional argument with which additional filtering options can be specified.

*Improvements*

- The AIMMS model editor incorrectly pasted Unicode characters as question marks.
- Reading the comment of a case did not close the `.dat` file afterwards.
- Name completion in dialog boxes did not work correctly when the name contained underscores.

*Issues ...*

- When implicitly opening an OpenOffice workbook using the function `SpreadSheet::SetActiveSheet`, a crash could occur when a connection with the workbook could not be established.
  - Using the Microsoft Text ODBC driver for reading `.csv` files, raised an error with newer versions of this driver.
  - UTF-8 encoding did not work correctly when writing an XML file from an Ascii version of AIMMS.
- 
- It was possible to open/close empty branches in a **Pivot Table** object, resulting in a misalignment between the header and the grid area. *... GUI*
  - In the **Gantt Chart** object, you now get a warning if you enter a reference date before 1970.
  - When changing the content of a **Gantt Chart** object, any mismatches in the indices of a color identifier are now still reported as an error, but the content change is applied anyway.
  - The **FileSelect** dialog did not initialize to the correct folder if the given file name contained only a partial path.
  - When multiple sources and multiple layers were defined in a **Network** object with a GIS background and not all layers were visible, the **Network** object showed some unnecessary flickering.
  - Repeatedly resizing a page with a button placed on it, could eventually lead to a crash.
- 
- Several math program suffices were not updated after a solve. *... Solving*
  - The procedure `GMP::Column::SetAsObjective` could fail if the model type was LS (i.e., if there was no objective).
  - The procedure `GMP::Tuning::TuneMultipleMPS` can now also be used to tune SAV files.
  - Incorrect column or row numbers were sometimes passed by the following GMP routines:  
`GMP::Column::GetName`,  
`GMP::Row::GetName` and  
`GMP::Solution::SetColumnValue`.
  - The AIMMS presolver could generate an incorrect constraint definition if it removed variables from a constraint containing the `/$` operator.
  - Coefficient values passed to `GMP::Coefficient::Set` were multiplied by -1 if the row contained the objective column. This has been changed. The new option `negate_objective_coefficient_modifications` can be switched on to get the old behavior. For upgraded projects the default will be automatically switched to 'on'; for new projects the default is set to 'off'.
  - AIMMS could produce a severe internal error executing a (non-GMP) solve statement inside a callback procedure of another (non-GMP) solve statement.
  - Setting 'aggressive node probing' was missing for option `MIP_node_presolve`

of CPLEX 12.5.1 (and higher).

- When there are multiple compound sets with the same root set domain but different contents, a search could visit too few elements resulting in incorrect results.
- The scalar iterative Min operator, with no condition and iterating over an empty set, would return 0 instead of inf.
- Defining an indexed set through an expression such as Union( k, PE(k) ) could result in a superfluous error about a mismatch of domain sets.
- Removing the definition from a variable would not remove that variable from the set AllConstraints.
- Set definitions were evaluated during a CleanDependents statement, instead of before it, resulting in inconsistent data.
- Consider a condition such as EP(i) <> e;. AIMMS will translate such a condition into REL\_EP(i, e) to improve the performance. After modifying the data of EP, AIMMS did not detect that the corresponding constraint is out of date when regenerating a mathematical program.
- Calling the function ExitAIMMS from within the startup procedure did only result in closing the project, and not in exiting AIMMS entirely.
- Deleting a unit parameter and subsequently creating a unit parameter with the same name could lead to superfluous error messages.
- Identifiers that depend on a database-related AIMMS identifier, are now properly marked as such.
- Identifier types no longer publicly supported, but present in upgraded models, will now be added to the predeclared set AllIdentifierTypes during compilation.
- At the beginning of a recompile step, index domain information is now rescanned to check whether deleted identifiers have been replaced by new ones.

... Compiler and Execution

---

### **AIMMS 3.14 FR2, release date 2013-10-16**

In this second AIMMS 3.14 Feature Release, the following issues have been fixed or improved.

3.14 FR2  
2013-10-16

**PLEASE NOTE:** The AIMMS Excel Add-In does not yet function well enough in this Feature Release. Therefore, we advise you not to use it in combination with this release. We will do our best to improve this situation.

**PLEASE NOTE**

- AIMMS could become unresponsive while writing a case, when all subsets of the set Integers have the property NoSave (there should be at least one subset of Integers in the model). If only the subset of Integers with the smallest integer has a NoSave property, loading cases could result in erroneous data.

Issues ...

- A resolve in the **Math Program Inspector** would result in a severe internal error after the solver was deleted.
  - Using the argument wizard to add an argument that used an index in its domain that was already used as a local index for some other procedure, did not work.
  - In some situations, the database mapping wizard did not allow you to select element parameters.
- 
- Docked pages, which are open during startup, could interfere with the startup sequence and sometimes even lead to a crash. *... GUI*
  - An early call to PageClose in the startup procedure could lead to a crash.
  - When determining the maximum column width in a **Pivot Table** object, any unit string present on an intermediate level in the column tree is now also taken into account.
  - Using an indexed color parameter to specify the text color of an explicit index in the **Pivot Table** object, did not work.
  - The **Gantt Chart** object's navigation keys did not work when the chart was scrolled down.
  - Adding new content to an existing **Gantt chart** object did not work anymore in AIMMS 3.14 FR1.
  - In the **Document Viewer** object you can now hide the scrollbar.
  - The PageGetNextInTreeWalk function now gives a correct error when a template page is passed as argument.
  - In a **Pivot Table** object on a data page for a parameter with a binary range, the values were not represented via check boxes, which was the case in earlier AIMMS versions.
- 
- The **Take Column Header Area Into Account** option (when 'Resetting the Column Width' for a **Pivot Table** object), now takes all rows in the column tree into account when determining the maximum size. Previously, it only considered the leaf cells in the column tree, but in some situations the text displayed at a parent level could be wider than the text displayed in a leaf cell. *... Pivot Table*
- 
- The GMP::instance::GetRowNumber function didn't take defined variables into account. *... Solving*
- 
- Giving an individual variable a .nonvar value of -1 after the model was generated for the first time, may result in ignoring the nonvar status of that column when regenerating the model. *... Compiler and Execution*
  - Consider the declaration:
 

```
SET:
  identifier : S
  index      : i
  order by   : user
  initial data: data { b, a } ;
```

and the assignment

```
S := sort( i, i );
```

In the above statement, the ordering of S would be used erroneously by the Sort operator, instead of the appropriate lexicographic order.

- Interrupting an AIMMS session executing a statement stmt within

```
Block
  stmt
OnError Err Do
  HndlErrStmt
EndBlock
```

would position the error message on the HndlErrStmt instead of on the actual (long running) stmt.

- Consider the following declarations:

```
SET:
  identifier : S
  index      : i ;
SET:
  identifier : T
  parameter  : t0
  index      : j ;
ELEMENT PARAMETER:
  identifier : eP
  index domain : i
  range      : T ;
ELEMENT VARIABLE:
  identifier : eV
  range      : S
  property   : emptyElementAllowed ;
CONSTRAINT:
  identifier : selectedTshouldBe
  definition : eP(eV) = t0 ;
```

With the above declarations, AIMMS would base its element numbering for the set S on a possible presence of the empty element, but did not do so for the set T. It should have done so, however, because an empty element for eV results in eP(eV) also being the empty element.

- The Prod operator over an empty condition could erroneously return 0 instead of 1.
- A runtime compiler check on the StringToElement function was added, which issues an error message when StringToElement extends a defined set from within a statement.

---

### **AIMMS 3.14 FR1, release date 2013-07-12**

In this first AIMMS 3.14 Feature Release, the following issues have been fixed or improved.

*Build 3.14 FR1  
2013-07-12*

**PLEASE NOTE:** The AIMMS Excel Add-In does not yet function well enough in this Feature Release. Therefore, we advise you not to use it in combination with this release. We will do our best to improve this situation.

**PLEASE NOTE**

- A new function has been introduced to obtain all spreadsheet names in a workbook: `Spreadsheet::GetAllSheets`.
- `Fit field` or `e format` or `0` was added as a new option value to the option `put_number_style`. Small values will be printed as `0`.
- The function `GMP::Instance::GetBestBound` replaces the function `GMP::Instance::GetLinearObjective`, which has become hidden and deprecated. Similarly, the function `GMP::SolverSession::GetBestBound` replaces the function `GMP::SolverSession::GetLinearObjective` and the function `GMP::Solution::GetBestBound` replaces `GMP::Solution::GetLinearObjective`.
- The concurrent optimizers of **CPLEX 12.5** and **Gurobi 5.5** are now supported by AIMMS. The concurrent optimizer can apply multiple algorithms to a single LP problem. The concurrent optimizer can be enabled by setting the **CPLEX** option `LP_method` or the **Gurobi** option `method`.
- The nonlinear presolver has been renamed to "AIMMS Presolver" because it can now also be used for linear problems, by switching on the new option `linear_presolve`. The AIMMS Presolver tries to reformulate more constraints as controlled by the new option `reformulate_constraints`.

*Improvements*

- AIMMS RPC is not included anymore from this release on. Depending on your specific needs, you may consider to use AIMMS PRO or our SDK technology for calling AIMMS remotely.
- Web Services are not included anymore from this release on. Depending on your specific needs, you may consider to use AIMMS PRO or our SDK technology for calling AIMMS remotely.
- The **Multi-Agent** feature has been removed from the AIMMS system.

*Deprecated functionality*

- The **XML Schema Mapping** tools stopped working because of the upgrade to MSXML 6 in earlier versions of AIMMS.
- Setting the option `data_management_style` to `Disk Files and Folders` could result in an incorrect message during AIMMS startup, especially if you got an early message about the solver configuration.
- Long error messages from `TestDataSource` could corrupt the internal memory leading to severe internal errors or fatal application errors later on.
- Combining multiple cases and "prior to case load procedure" could lead to a severe internal error when switching between active cases.

*Issues ...*

- When writing to a SQLite database, the database tracing didn't show the data that was transferred (with option `Database_trace_level` set to 2).
  - Various examples have been removed that make use of no longer supported features of AIMMS.
  - In the **CaseLoad** or **CaseSave** dialog box, when using data management style **Disk Files and Folders**, filtering on a specific Case Type did not work correctly when the case type itself was defined in a module or library. In other words, when the full name of the case type contains a module or library prefix.
  - Removing the **Schedule domain** attribute from an **Activity** could lead to a severe internal error when restarting the project.
  - AIMMS could become unresponsive while writing a case, when all subsets of the set **Integers** have the property **NoSave** (there should be at least one subset of **Integers** in the model). If only the subset of **Integers** with the smallest integer has a **NoSave** property, loading cases could result in erroneous data.
- 
- The use of indexed unit parameters as the **Unit** attribute of other parameters and variables, could prevent data display of the inactive cases in a multiple case object on a page. *... GUI*
  - The **Pivot Table** object will not display a unit anymore when the cell is empty (e.g. because the cell value is at its default).
  - Sometimes the GIS background image of a network object appeared as background of other areas of the page.
  - A **Curve** object with the property **Remove Defaults**, did not render correctly if the indices in the **Y identifier** and **X identifier** were not in the same order.
  - Property dialogs that display properties in a tree format, like the ones used in the **ActiveX** objects or the **Pivot Table** object, would not always correctly display multiplicity properties for which the only entry was a double-click to create new entry.
  - When running under AIMMS PRO, tooltips in a **Pivot Table** object could cause a crash.
- 
- In a **Pivot Table** object, switching to another outer index for cases in which the element description (shown in the dropdown) was identical for distinct elements, would not update the grid area correctly. *... Pivot Table*
  - The multi-element filter in a **Pivot Table** object did not allow you to select the first identifier in your model.
- 
- Generating more than 20.000 non-linear terms in a constraint programming model could result in a severe internal error. *... Solving*
  - Producing a solution listing could, in rare circumstances, result in a severe internal error.
  - Solution values outside bounds are now marked as infeasible in the con-

- straint listing with columns.
  - see issue 25127
  - Solution information is now available for **Gurobi** if the model is infeasible, the presolver is turned off, and the model type is LP or RMIP.
  - Solving an LP and regenerating and solving it two times more, whereby a constraint exists with the following properties:
    - there are multiple terms referencing the same variable and
    - some term would generate non-zero coefficients, but these coefficients cancel out resulting in no row generated for that constraint could lead to a severe internal error.
  - The postsolve step used by **CBC** and **Gurobi** could return incorrect variable values.
  - Procedure `GMP::Coefficient::Set` could result in a severe internal error if the row was not generated before.
  - There is a new version of **CP Optimizer**, i.e. **CP Optimizer 12.5**.
  - The option `MIP_kappa` has been added for **Cplex** 12.5.
  - The **XPRESS** link has been removed from the AIMMS installation.
  - **GMP-AOA** could crash if **MOSEK** was used as NLP solver.
- Data entry for indexed sets, whereby a domain set or the range of that indexed set is a compound set, is now supported. *Model Editor*
  - Editing values in the **Identifier View** window could result in a crash.
  - A comment-only **Body call** attribute of an external function could lead to a severe internal error.
- The `FileCopy` function did not check whether the destination file was already opened by some other process on Linux. *... Compiler and Execution*
  - Executing a function with a three or higher dimensional result multiple times, could result in memory corruption and severe internal errors later on.
  - A `Raise Error` statement, guarded by a `block / onerror / endblock` statement, but not marked as handled, might not stop AIMMS.
  - AIMMS allowed identifiers that start with a number to be used as a prefix. Now this is flagged as an error.
  - The use of `UNDF` in data of an identifier and used as operand in the `AND` operator, could result in unjust use as a truth value in indexed operations.
  - Quantity based scaling, introduced as part of Constraint Programming in AIMMS 3.13, could cause a severe internal error when a parameter has a unit, say PSI, that is commensurate with another unit, say PA, and PA is defined over three other units, say kg, m, and s, and a **Convention** is used to convert kg, m, or s.
  - Specifying a unit for an optional argument of a procedure or function and subsequently calling that procedure or function without the optional argument, would lead to a superfluous warning.



- The intrinsic function `OptionGetDefaultString`, which was missing from AIMMS, is now implemented.
- When using a construct like `Q(c, i, j, k) | c = TheCase`, where `c` is an index in `AllCases` and `TheCase` is an element parameter in `AllCases`, with a definition of `c.P(i, j)`, AIMMS might use the incorrect case for retrieving the value of the first element of `P(i, j)`.
- `P(n-i+m)`, where `n,m` are integer valued expressions and `i` is an index over a subset of the integers, would be handled as if parsed as `P( n-(i+m) )` instead of `P( (n-i)+m )`.
- AIMMS now supports units declared in the `SI_Time_Duration` quantity such as "week". Such a time unit needs to be based on one of the granularities { seconds, minute, hour, day, month, year }. In addition, there is a warning added against huge calendars.
- Iterating over the elements in a set stored in a case as in the following statement:
 

```
for i in epSelectedCase.someSet do
! do something with i
endfor ;
```

 might lead to superfluous error messages when the elements in the set `epSelectedCase.someSet` are not in the root set of `someSet`.
- Consider the expression: `mean( i, otherExpr )` In case `otherExpr` uses an iterative operator this could lead to a superfluous error message.
- AIMMS could produce erroneous results for expressions like:
 

```
sum((i,j,k) | k in anIndexedSet(i,j-1))
```

 Here, `k` is the last running index and a lag/lead is used in the referencing of the indexed set `anIndexedSet`.
- Expressions involving atomic/compound index translation might erroneously be translated into identifier references, resulting in incorrect data to be displayed in the GUI and display statements.