

---

## **AIMMS User's Guide - Calling AIMMS**

This file contains only one chapter of the book. For a free download of the complete book in pdf format, please visit [www.aimms.com](http://www.aimms.com) or order your hard-copy at [www.lulu.com/aimms](http://www.lulu.com/aimms).

Copyright © 1993–2011 by Paragon Decision Technology B.V. All rights reserved.

Paragon Decision Technology B.V.	Paragon Decision Technology Inc.	Paragon Decision Technology Pte.
Schipholweg 1	500 108th Avenue NE	Ltd.
2034 LS Haarlem	Ste. # 1085	80 Raffles Place
The Netherlands	Bellevue, WA 98004	UOB Plaza 1, Level 36-01
Tel.: +31 23 5511512	USA	Singapore 048624
Fax: +31 23 5511517	Tel.: +1 425 458 4024	Tel.: +65 9640 4182
	Fax: +1 425 458 4025	

Email: [info@aimms.com](mailto:info@aimms.com)  
WWW: [www.aimms.com](http://www.aimms.com)

AIMMS is a registered trademark of Paragon Decision Technology B.V. IBM ILOG CPLEX and sc CPLEX is a registered trademark of IBM Corporation. GUROBI is a registered trademark of Gurobi Optimization, Inc. KNITRO is a registered trademark of Ziena Optimization, Inc. XPRESS-MP is a registered trademark of FICO Fair Isaac Corporation. MOSEK is a registered trademark of Mosek ApS. WINDOWS and EXCEL are registered trademarks of Microsoft Corporation.  $\text{T}_{\text{E}}\text{X}$ ,  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ , and  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  are trademarks of the American Mathematical Society. LUCIDA is a registered trademark of Bigelow & Holmes Inc. ACROBAT is a registered trademark of Adobe Systems Inc. Other brands and their products are trademarks of their respective holders.

Information in this document is subject to change without notice and does not represent a commitment on the part of Paragon Decision Technology B.V. The software described in this document is furnished under a license agreement and may only be used and copied in accordance with the terms of the agreement. The documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Paragon Decision Technology B.V.

**Paragon Decision Technology B.V. makes no representation or warranty with respect to the adequacy of this documentation or the programs which it describes for any particular purpose or with respect to its adequacy to produce any particular result. In no event shall Paragon Decision Technology B.V., its employees, its contractors or the authors of this documentation be liable for special, direct, indirect or consequential damages, losses, costs, charges, claims, demands, or claims for lost profits, fees or expenses of any nature or kind.**

**In addition to the foregoing, users should recognize that all complex software systems and their documentation contain errors and omissions. The authors, Paragon Decision Technology B.V. and its employees, and its contractors shall not be responsible under any circumstances for providing information or corrections to errors and omissions discovered at any time in this book or the software it describes, whether or not they are aware of the errors or omissions. The authors, Paragon Decision Technology B.V. and its employees, and its contractors do not recommend the use of the software described in this book for applications in which errors or omissions could threaten life, injury or significant loss.**

This documentation was typeset by Paragon Decision Technology B.V. using  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  and the LUCIDA font family.

# Chapter 20

## Calling AIMMS

This chapter discusses the command line options of the AIMMS program, and explains the details for running AIMMS end-user applications. In addition, the chapter explains how you can link AIMMS to your own program as a DLL, and presents a short overview of the functionality available through the AIMMS-specific Application Programming Interface (API) provided by this DLL.

*This chapter*

---

### 20.1 AIMMS command line options

On the AIMMS command line, you can specify a number of options and arguments that will influence the manner in which AIMMS is started. The following line illustrates the general structure of a call to the AIMMS program.

*Calling AIMMS*

```
aimms.exe [command-line-options] [project-file [session-arguments]]
```

Table 20.1 provides an overview of the command line options that you can specify. AIMMS offers both long and short option names, and some options require a single argument. All short option names start with a single minus (-) sign, followed by a single character. By convention, short options that require an argument use capital characters. The long option names are always preceded by a double minus sign (--), followed by a descriptive text. In general, the long option names are easier to remember, while the short names permit a more compact command line. Short option names without an argument may be appended one after another with only a single minus sign at the beginning.

*Command line options*

When an AIMMS project is linked to an end-user database (see Chapter 21), you must log on to the project before being able to run it. Through the `--user` command line option, you can specify a user name and optionally a password with which you want to log on to the system. When you specify just a user name, a log on screen will appear with the provided user name already filled in. If you specify a password as well, AIMMS will verify its correctness and skip the log on screen altogether if the user name- password combination is acceptable. Providing both the user name and the password is not recommended for interactive use, but may be convenient when you want the model to run unattended.

*Specifying a user*

Long name	Short name	Argument
<code>--user</code>	<code>-U</code>	user[:password]
<code>--data</code>	<code>-D</code>	data manager file
<code>--backup-dir</code>	<code>-B</code>	backup directory
<code>--log-dir</code>	<code>-L</code>	log directory
<code>--config-dir</code>	<code>-C</code>	configuration directory
<code>--license</code>		license name
<code>--license-wait-seconds</code>		seconds to wait
<code>--run-only</code>	<code>-R</code>	procedure name
<code>--user-database</code>		user database file
<code>--minimized</code>	<code>-m</code>	—
<code>--maximized</code>	<code>-x</code>	—
<code>--hidden</code>		—
<code>--as-server</code>		—
<code>--developer</code>	<code>-d</code>	—
<code>--end-user</code>	<code>-e</code>	—
<code>--no-solve</code>		—
<code>--help</code>	<code>-h</code>	—

Table 20.1: AIMMS command line options

By default, AIMMS gets its cases and datasets from the last selected project-dependent data manager file. You can always select your own choice of data manager file using the **File-Open** menu. With the `--data` flag, you can specify an alternative data manager file with which you want to open the project. You can only use this option if you also specify a project file.

*Specifying a data manager file*

With the `--backup-dir` and `--log-dir` options you can override the default directories where AIMMS will store temporary information such as case and model backups, the AIMMS and solver listings, and the message log. You can modify the defaults for these directories using the project options dialog box (see Section 22.1).

*Backup and log directories*

By default, AIMMS stores a number of global configuration files, such as the AIMMS license file, VAR license files and the solver configuration file, in the common application area of your computer (see also Section 2.6.4). If you want to store configuration files in a different location, you can indicate this through the `--config-dir` option. You can use this option, for instance, to indicate where the configuration files for your particular machine can be found when the AIMMS system that you use is stored on a network disk, and when you do not use a license server.

*AIMMS configuration*

Through the `--license` option you can select any AIMMS license that you installed in the AIMMS **License Configuration** dialog box (see also Section 2.6). The value that you specify for the `--license` option should match an entry in the **License** column in the left pane of the **License Configuration** dialog box. In case you are using a network license with different profiles, you should make a different entry in the AIMMS **License Configuration** for each profile you want to use and you can use the `--license` option to open AIMMS with a license with a specific profile.

*License name*

When you are using a network license, the license server may not have a license available for you right away. Through the `--license-wait-seconds` option you can specify the number of seconds you want AIMMS to wait for a network license to become available. If you do not specify this option AIMMS will use a default timeout of 0 seconds. When reaching the given timeout, AIMMS will try the next license in your license configuration, or will return with a license error if no other licenses are available.

*Network logon timeout*

When your application has been set up for use by multiple users, all user and group information associated with the application is stored in a separate (encrypted) user database (see Section 21.3 for more details on this topic). Through the `--user-database` option you can move the location of this user database file (to for example a single location that is shared among all users on the network) even though you might not have developer rights to the application.

*User database location*

Through the `--minimized`, `--hidden` and `--maximized` options you can indicate whether you want AIMMS to start in a minimized or hidden state (i.e. just as a button on the task bar, or not visible at all), or to fill up the entire screen. Running AIMMS minimized or hidden may be convenient when AIMMS is called non-interactively from within another program through the AIMMS API (see Chapter 32 of the Language Reference). In this way, your program can use AIMMS to solve an optimization model after which it resumes its own execution. The `--as-server` option extends the `--hidden` option, and should be used when AIMMS is started with limited privileges by a system service (e.g. through the Internet Information Server). It suppresses all dialog boxes that may appear during startup of AIMMS, as well as during the execution of your model.

*Running minimized, hidden, or as server*

With the `--developer` and `--end-user` options you can request AIMMS to start up a project in developer mode or end-user mode, respectively, overriding the default start-up mode of the project (see also Section 16.1). When you are the developer of an AIMMS-based application, you are always allowed to run the application in end-user mode. If you are using a VAR-licensed AIMMS application, starting the application in developer mode may be prohibited, or subject to an additional developer password (as explained in Section 16.1).

*Developer versus end-user mode*

AIMMS strictly enforces that the number of AIMMS sessions with full solving capabilities running on your computer simultaneously is in accordance with your AIMMS license. Typically, for a single-user license, this means that you can only start up a single AIMMS session that is capable of solving optimization programs at a time. However, for every fully capable AIMMS session, AIMMS also allows you to start up an additional AIMMS session without solving capabilities. You can use such a session, for instance, to make modifications to your model, while a first session is executing an optimization run. In that case, AIMMS will present a dialog box during start up to indicate that the session has no solving capabilities. You can suppress this dialog box, by specifying the `--no-solve` command line option.

*Solverless  
AIMMS sessions*

When you want to run an AIMMS project unattended, you can call AIMMS with the `--run-only` option. This option requires the name of a procedure in the model, which will be executed after the project is opened. When you use the `--run-only` option, all other initial project settings, such as the initial case, procedure and page settings (see Section 16.1), will be ignored. AIMMS will, however, call the procedures `MainInitialization` and `MainTermination` as usual. Once the procedure has finished, the AIMMS session will be terminated. You can only specify the `--run-only` option if you also specify a project file on the command line.

*Executing a  
procedure and  
terminating  
AIMMS*

AIMMS will interpret the first non-option argument on the command line as the name of the project file with which you want to open AIMMS. If you specify a project file, the settings of the project may initiate model-related execution or automatically open a page within the project.

*Opening a  
project to run*

If you want to open a project for editing purposes only, you should hold down the **Shift** key when opening the project. The initial actions will also not be performed if the command line contains the `--run-only` option. In this case execution takes place from within the specified procedure only.

*Opening a  
project to edit*

Directly after the name of the project file, AIMMS allows you to specify an arbitrary number of string arguments which are not interpreted by AIMMS, but can be used to pass command line information to the project. In the model, you can obtain the values of these string arguments one at a time through the predefined function `SessionArgument`, which is explained in more detail in Section 19.4.6.

*Passing session  
arguments*

The following call to AIMMS, will cause AIMMS to start the project called `transport.prj` in a minimized state using the user name `batchuser` with password `batchpw`, run the procedure `ComputeTransport`, and subsequently end the session. A single argument "Transport Data" is provided as a session argument for the model itself.

*Example*

```
aimms --minimized --user batchuser:batchpw --run-only ComputeTransport \
transport.prj "Transport Data"
```

Note that the \ character at the end of the first line serves as the continuation character to form a single command line. Using the short option names, you can specify the same command line more compactly as

```
aimms -mUbatchuser:batchpw -RComputeTransport transport.prj "Transport Data"
```

In this command line, the -m and -U options are combined. No space is required between a short option name and its argument.

Given the above AIMMS call, you can use the function `SessionArgument` to fetch the first session argument and assign it to the string parameter `ODBCDataSource` as follows.

*Using session arguments*

```
if ( SessionArgument(1, ODBCDataSource) ) then
  /*
   * Execute a number of READ statements from ODBCDataSource
   */
endif;
```

Following this statement, the string parameter `ODBCDataSource` will hold the string "Transport Data". In this example, the string parameter `ODBCDataSource` is intended to serve as the data source name in one or more DATABASE TABLE identifiers, from which the input data of the model must be read.

---

## 20.2 Calling AIMMS from external applications

In addition to starting the AIMMS program itself, you can also link AIMMS, as a component, to your own application. Using AIMMS as a component has the advantage that, from within your program, you can easily access data with AIMMS and run procedures in the associated AIMMS project. Thus, for instance, when your program requires optimization, and you do not want to bother writing the interface to a linear or nonlinear solver yourself, you can

*Use AIMMS as a component*

- specify the optimization model algebraically in AIMMS,
- feed it with data from your application, and
- retrieve the solution after the model has been solved successfully.

When linking AIMMS as a component to your own application, you have several options:

*Several options*

- use the AIMMS COM object (see The COM Object [User's Guide](#) and Reference),

- call your AIMMS project as a webservice (see The Multi Agent and Web Services [User's Guide](#)),
- when linking from within an Excel spreadsheet, use the Excel Add-In (see The Excel Add-In [User's Guide](#)), or
- link directly against the AIMMS API (see Chapter 32 of the Language Reference).

Through the AIMMS component technologies described above you have varying degrees of control over the data inside your model. Use of these technologies requires, however, that you set up the interface to your model in a programming language such as C/C++, Java or .NET. While the control offered by these technologies may be relevant for advanced or real-time applications where efficiency in data communication is of the utmost importance, these technologies come with a certain learning curve, and if you only want to perform simple tasks such as communicating data in a blockwise manner and running procedures inside the model, you might consider setting up the communication using either ASCII data files or databases.

*Programming  
required*

Please note that using the AIMMS API to start up a new AIMMS session from within an external application that also performs other significant tasks than starting up that AIMMS session, is *not recommended*. Opening an AIMMS project from within another application may, especially under Windows, lead to unwanted interactions between the AIMMS and the original application. The AIMMS API is also not particularly suited to start up an AIMMS session from within the same process multiple times. In such cases we advise to use a technology that starts up an AIMMS session in a separate process.

*Using the  
AIMMS API*

---

### 20.3 The AIMMS command line tool

Next to accessing AIMMS from within your own programs through the AIMMS component technologies, AIMMS also supports a command line tool through which you can control an AIMMS project externally. You can start the AIMMS command line tool by running

*AIMMS com-  
mand line  
tool*

```
AimmsCmd project-path
```

The AimmsCmd program is located in the Bin directory of your AIMMS installation.

The AIMMS command line tool offers commands to

*Commands*

- assign values to sets, and to scalar and multidimensional identifier slices,
- display the contents of sets, and the values of scalar and multidimensional identifier slices,
- empty sets or multidimensional identifier slices,
- retrieve the cardinality of sets or multidimensional identifier slices,

- run procedures,
- execute system commands, and
- close the AIMMS project and quit the program.

Each command is terminated by a semicolon.

You can assign a value to sets and multidimensional identifiers and slices thereof through one of the commands *Assignments*

```
Let reference := data-expression ;
Let reference += data-expression ;
```

where the := operator refers to completely replacing the contents of *reference* and the += operator refers to a merge operation.

A *reference* in an assignment is either

*References*

- an identifier name such as “Transport”, or
- a reference to an identifier slice such as

```
Transport('Amsterdam', j)
```

where each sliced dimension must refer to a quoted set element.

The *data expressions* allowed in an assignment are

*Data expressions*

- a set expression preceded by the keyword Set as in

```
Set {'Amsterdam', 'Rotterdam'}
```

where all set elements must be quoted,

- a ranged integer set preceded by the keyword Set as in

```
Set {1 .. 10}
```

- a scalar numeric, element or string value as in

```
10
11.7
'an element'
"a string"
```

- a tuple list of numeric, element or string values preceded by the keyword List as in

```
List {'Amsterdam', 'Paris'} : 10, ('Paris', 'London') : 20}
```

- a dense multidimensional array of numeric, element or string values preceded by the keyword Array as in

```
Array [[1,2], [3,4], [5,6]]
```

You can request AIMMS to display the contents of sets and multidimensional identifier slices in your model through the command

```
Display reference [:precision] [as Array] ;
```

For multidimensional identifier data AIMMS will, by default, use the List format described above. Through the optional “as Array” clause you can instruct AIMMS to display the identifier data as a dense array.

To empty the data of sets and multidimensional identifier slices in your model you can use the command

```
Empty reference ;
```

You can request AIMMS to retrieve the cardinality of sets and multidimensional identifier slices in your model through the command

```
Card reference ;
```

With the command

```
Run procedure-name ;
```

you can request AIMMS to run a procedure (without arguments). When finished, AIMMS will display the return value of the procedure.

You can let AIMMS execute a system command through the command

```
System system-command ;
```

where *system-command* is a string to be executed by command shell.

Through the Help command, a list with a brief description all available commands will be displayed.

You can close the AIMMS project and quit the command line tool through the command

```
Quit ;
```