



MOSEK Release notes
Release 10.0.29

MOSEK ApS

21 November 2022

Contents

1	Supported platforms	1
2	Major changes	3
3	Known issues	6
4	Bug fixes	7

Chapter 1

Supported platforms

Below are the **minimal requirements** for various **MOSEK** interfaces and operating systems. In some cases using **MOSEK** with older versions of the software will be possible, but is neither actively supported nor tested.

Operating systems

Table 1.1: Operating systems

Platform	Minimal OS version	Specific library dependencies
linux64x86	RHEL 7, Ubuntu 18.04 or compatible	GLIBC 2.17, GLIBCXX 3.4.11
osx64x86	macOS 10.15	
win64x86	Windows 10, Server 2016	
win32x86	Windows 10, Server 2016	
linuxaarch64	Ubuntu 20.04 or compatible	GLIBC 2.29, GLIBCXX 3.4.21
osxaarch64	macOS 11	

Optimizer API and Fusion API

Table 1.2: Optimizer API and Fusion API.

Platform	C	C++(Fusion)	Java	.NET	.NET Core	Python
linux64x86	Yes	C++11	1.8	–	netstandard2.0	3.7-3.10
osx64x86	Yes	C++11	1.8	–	netstandard2.0	3.7-3.10
win64x86	Yes	C++11	1.8	4.5	netstandard2.0	3.7-3.10
win32x86	Yes	–	–	4.5	–	–
linuxaarch64	Yes	C++11	1.8	–	netstandard2.0	3.7-3.10
osxaarch64	Yes	C++11	17	–	netstandard2.0	3.8-3.10

Optimization Toolbox for MATLAB, Rmosek and other MOSEK tools

Table 1.3: Other APIs and tools.

Platform	MATLAB	R	AMPL Shell	MOSEK to AMPL link	OptServer	Imgrd
linux64x86	R2017a	3.6	—	Yes	Yes	Yes
osx64x86	R2017a	3.6	—	Yes	–	Yes
win64x86	R2017a	3.6	—	Yes	–	Yes
win32x86	–	–	–	Yes	–	Yes
linuxaarch64	–	–	–	–	–	Yes
osxaarch64	–	4.1	–	–	–	Yes

Other distribution channels (stable releases only)

- A Python Anaconda package. <https://anaconda.org/MOSEK/mosek>
- A Python wheels package. <https://pypi.org/project/Mosek/>
- A NuGet package. <https://www.nuget.org/packages/Mosek/>
- A Julia package. <https://github.com/MOSEK/Mosek.jl>
- A Rust package for the Optimizer API (unofficial). <https://lib.rs/crates/mosek>

Other remarks

- If you are using a floating license, the license server on Linux requires *Linux Standard Base 3* or newer is installed (package **lsb**).
- Numpy is required in Python Fusion.

Chapter 2

Major changes

Specific information regarding particular APIs, parameters and portability of code from version 9 can be found in the section *Interface changes* towards the end of the respective manual. This section lists general changes throughout **MOSEK**.

2.1 Release notes for 10.0

2.1.1 New features

Platform support

- Introduced native support for Apple Silicon M1 `osxaarch64`.
- Improved support and introduced multithreading for Linux ARM 64bit `linuxaarch64`.

Optimizer

- Introduced new cone types: generalized power cone (primal and dual), geometric mean cone (primal and dual) and the cone of vectorized lower-triangular parts of semidefinite matrices.

Presolve

- Presolve has been improved significantly for conic problems.

Interior-point Optimizer

- Improved performance for large-scale linear problems.
- Significantly improved performance when running on recent AMD CPUs.
- Significantly improved multithreaded performance for a class of SDPs.

Mixed-integer Optimizer

- (Optimizer API and Fusion). Introduced disjunctive constraints (DJC), a language for writing constraints of the form $A_1 \text{ OR } A_2 \text{ OR } \dots \text{ OR } A_k$ where A_i are ordinary linear equations or inequalities.
- Improved cutting plane separation; in particular, implied bound cuts, controllable with the parameter `MSK_IPAR_MIO_CUT_IMPLIED_BOUND`, are used by default.
- Improved presolve, in particular probing and the aggregator. The latter can be controlled with the new parameter `MSK_IPAR_MIO_PRESOLVE_AGGREGATOR_USE`.
- Introduced symmetry detection and exploitation, controllable with the parameter `MSK_IPAR_MIO_SYMMETRY_LEVEL`.

- Introduced reformulation methods for MIQCQO problems, controllable with the parameter `MSK_IPAR_MIO_QCQO_REFORMULATION_METHOD`. This allows the solution of non-convex MIQCQO problems that are amenable to such a reformulation.
- Introduced parameters `MSK_IPAR_MIO_NUMERICAL_EMPHASIS_LEVEL` and `MSK_IPAR_MIO_MEMORY_EMPHASIS_LEVEL` that may be used to reduce numerical and memory issues, respectively.
- Distinction between complete and partial initial user solutions, the exploitation of the latter being controllable with the parameter `MSK_IPAR_MIO_CONSTRUCT_SOL`.

Multithreading

- Switch from Cilk to the [oneTBB](#) library for multithreading in the interior-point optimizer. The reason is Cilk has been deprecated in the Intel C compiler tool chain. Moreover, compared to Cilk oneTBB allows fine grained control of the number of threads employed in each optimization.
- Introduce parallel optimization of a number of tasks in the Optimizer API (`optimizebatch`) and Fusion (`solveBatch`).
- Previously missing support for multithreading has been enabled on the `linuxaarch64` (Linux ARM64) platform.

Licensing

- Updated FLEXlm to version 11.18.3. Upgrade of floating license servers is required to use **MOSEK** 10 clients.
- License server available for all platforms.

Interface

- (Optimizer API). Introduced affine conic constraints (ACC) as a preferred method of expressing conic problems. An affine conic constraints allows the user to directly write $Fx + g \in \mathcal{K}$ without introducing slack variables.
- (Optimizer API). Introduced the possibility to create tasks without an environment.
- (Optimizer API, Python, Java, .NET). Methods can return arrays and tuples instead of taking them as arguments.
- (Fusion). Allow more flexible naming in *Fusion*.

I/O

- Faster I/O across all file formats.
- Fixed interpretation of integer keyword in LP files.
- The PTF format replaces the OPF format as a read/write human-readable file format for conic problems, supporting also semidefinite variables.
- (Optimizer API and Fusion) Allow writing data files to streams instead of files.

2.1.2 Changes compared to version 9

- The function `computesparsedcholesky` has changed API: the argument previously indicating whether to use multiple threads or not is now an integer denoting the number of threads to use, with 0 meaning that **MOSEK** makes the choice.

2.1.3 Removed features

- Support for Python 2.7 on all platforms.
- Support for Java on Windows 32 bit.
- Support for all versions of Python on Windows 32 bit.

2.1.4 Deprecated features

- Conic constraints restricted to $x \in \mathcal{K}$ for a variable x are deprecated as a method of representing conic problems in favor of affine conic constraints (ACC) of the form $Fx + g \in \mathcal{K}$. This affects mainly the Optimizer API, in a minor degree the Optimization Toolbox for MATLAB and Rmosek, and does not affect *Fusion* where this language was already used.
- The OPF file format for conic problems is deprecated in favor of PTF.
- It is recommended to move away from Windows 32 bit platform.

Chapter 3

Known issues

Chapter 4

Bug fixes

10.0.29

- Fixed a bug in the presolve occurring for certain conic problems. The bug would trigger an assert.
- Added work limits for variable linear dependency check in the mixed-integer solver.

10.0.28

- Fixed bugs in the presolve occurring for certain conic problems.
- Small updates in documentation and FAQ.
- Some improvements in symmetry detection in the mixed-integer optimizer, especially regarding memory consumption.

10.0.27

- Fixed an issue with cloning a *Fusion* model which contains solutions.
- Fixed an issue with reading task files containing certain partial solution configurations.
- Fixed an exception handling error which could occur when reading invalid task files.

10.0.26

- Fix fetching values for partially defined user solutions.

10.0.25

- Fixed possible inconsistencies when one domain object enters many constraints in *Fusion*.
- Fixed a minor issue regarding reading special parameter values from a task file.

10.0.24

- Fixed a memory leak appearing when disposing a C++ *Fusion* model with conic or disjunctive constraints.

10.0.23

- Fixed a rare memory reallocation issue in *Fusion*.
- Silenced some deprecation warnings related to `numpy` in *Fusion*.

10.0.22

- Fixed a bug occurring for certain conic problems.
- Fixed a status code check syntax error affecting `intlinprog`.
- Fixed an issue with cleaning the task when reading data from a string.

10.0.21

- Downgrade .NET Core package target to .NETStandard2.0.
- Fixed a bug occurring for certain conic problems.

10.0.20

- Fixed a bug occurring for certain ways to call BLAS/LAPACK functions on AMD.
- Fixed a bug in mixed-integer optimizer setup for quadratic problems (QO).
- Fixed some minor bugs in the presolve occurring on conic problems.

10.0.18

- First stable release.
- Fixed an issue with propagating initial solutions for MIQO and MIQCQO problems.
- Improved efficiency of file I/O.

10.0.17(BETA)

- Fixed a bug that could causing invalid infeasibility certificates to be reported for quadratic optimization problems.
- Improved presolve slightly for conic problems.
- Improved performance for some semi-definite problems, particularly when using multiple threads.

10.0.16(BETA)

- Improved performance for some semi-definite problems, particularly when using multiple threads.
- Fixed some issues related to file formats.

10.0.15(BETA)

- Improved documentation.
- Environment-free task creation in optimizer API.
- Fixed a multithreading issue on Linux ARM64.

10.0.14(BETA)

- The (plain) solution file now contains the solution for the affine conic constraints when present.
- Fixed a bug occurring when deleting one or more variables.
- Documented new naming possibilities in Fusion.
- Removed some debug output.

10.0.13(BETA)

- Downgrade Linux 64x86 platform requirements to RHEL 7, Ubuntu 18.04.
- Downgrade MATLAB version requirement to R2017a.
- Fixes in the remote optimization server and improved communication with the server.
- Writing files to streams.
- Batch optimization also available in Fusion.
- Fixed a bug that in rare cases caused an assert in the interior-point optimizer.

10.0.12(BETA)

- First beta release.
- See [Sec. 2](#) and the *Interface changes* section towards the end of your API manual.