

Ansys Granta MI 2022 R2

MI Scripting Toolkit 3.0 for Python

Overview and Installation

Copyright and Trademark Information

© 2022 ANSYS, Inc. or its affiliated companies. All rights reserved.

Ansys, Ansys Workbench, AUTODYN, CFX, FLUENT and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries located in the United States or other countries. ICEM CFD is a trademark used by ANSYS, Inc. under license. CFX is a trademark of Sony Corporation in Japan. All other brand, product, service and feature names or trademarks are the property of their respective owners. FLEXlm and FLEXnet are trademarks of Flexera Software LLC.

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS.

The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. and ANSYS Europe, Ltd. are UL registered ISO 9001: 2015 companies.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-Party Software

See the legal information in the product help files for the complete Legal Notice for Ansys proprietary software and third-party software. If you are unable to access the Legal Notice, contact ANSYS, Inc.

Published in the U.S.A.

We welcome your feedback on this document. Please let us know if anything is unclear, if you spot an error, or have an idea for new content, by emailing granta-docs@ansys.com.

Document version: DOCMISTKP3.0/01

Last Revised: May 2022

Table of Contents

1	<i>MI Scripting Toolkit Overview</i>	4
1.1	Architecture.....	4
2	<i>Installation instructions for Windows</i>	5
2.1	Download package	5
2.2	System requirements	5
2.3	Installation instructions.....	5
3	<i>Installation instructions for Linux</i>	6
3.1	Download package	6
3.2	System requirements	6
3.3	Installation instructions.....	7
3.4	Required shared libraries	7

1 MI Scripting Toolkit Overview

The MI Scripting Toolkit enables you to analyze and transform the materials data in your Granta MI system using Python. Data can be extracted based on known record lists or on user-supplied criteria, and the results of the analysis can be written back to the Granta MI database with full traceability.

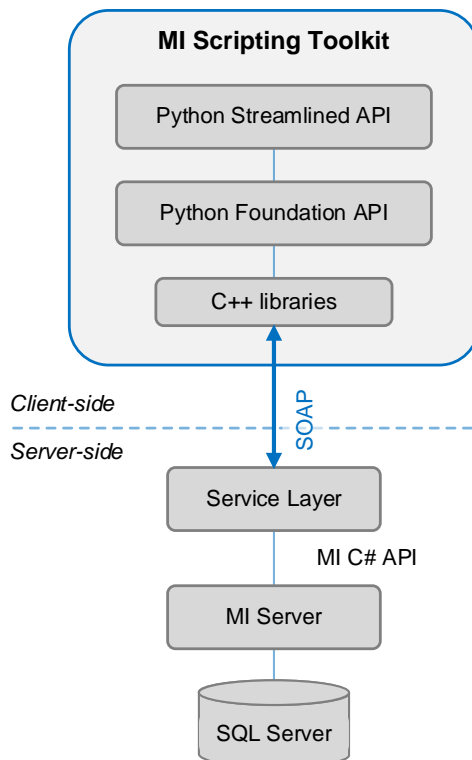
The toolkit supports the manipulation of the following Granta MI data types:

- Numerical (Point, Range, Integer)
- Text (Short Text, Long Text, Discrete)
- Functional (Float Functional)
- Tabular
- Record links
- Picture
- Other (Logical, Date, File, Hyperlink)

1.1 Architecture

The MI Scripting Toolkit comprises the Streamlined API, the Foundation API, and C++ libraries.

It is installed locally and connects to your Granta MI application server (MI Server) via the Service Layer.



2 Installation instructions for Windows

2.1 Download package

The MI Scripting Toolkit for Python download package includes:

- This document
- *MI Scripting Toolkit 3.0 for Python Release Notes*
- Help (module reference and example Python Notebooks)
- The installer file for Windows: *MIScriptingToolkit-3.0.xxx-py3-none-win_amd64.whl*

2.2 System requirements

- Windows® 10 64-bit
- Granta MI 2021 R1 or later, with Service Layer Interface 20/05
- Python 3.7 64-bit or later

2.3 Installation instructions

1. Before installing MI Scripting Toolkit, please uninstall any earlier versions.
2. To install the MI Scripting Toolkit, enter the following into the windows command line:

```
$ pip install MIScriptingToolkit-3.0.xxx-py3-none-win_amd64.whl
```

Or, if you have multiple versions of Python installed:

```
$ python -m pip install MIScriptingToolkit-3.0.xxx-py3-none-win_amd64.whl
```

When installation is complete, the MI Scripting Toolkit for Python files are installed into the Python site-packages location.

3 Installation instructions for Linux

3.1 Download package

The MI Scripting Toolkit for Python download package includes:

- This document
- *MI Scripting Toolkit 3.0 for Python Release Notes*
- Help (module reference and example Python Notebooks)
- An installer file for your version of Linux:

Linux version	Installer file name
CentOS 7.9	CentOS7x\MIScriptingToolkit-3.0.xxx-py3-none-any.whl
Debian 9.11	Debian9\MIScriptingToolkit-3.0.xxx-py3-none-any.whl
Red Hat Enterprise 7.9	RHEL7x\MIScriptingToolkit-3.0.xxx-py3-none-any.whl
SUSE Enterprise 15	SLES15\MIScriptingToolkit-3.0.xxx-py3-none-any.whl
Ubuntu 18	Ubuntu18\MIScriptingToolkit-3.0.xxx-py3-none-any.whl

For information on required shared libraries, see Section 3.4 on page 7.

3.2 System requirements

3.2.1 Supported Linux distributions

The following Linux distributions and versions are supported. The MI Scripting Toolkit 3.0 has been tested and certified on the specific OS versions indicated in the right-hand column:

Distribution	Version	Certified on
CentOS	CentOS 7	7.9.2009
Debian	Debian 9	9.11
Red Hat Enterprise Linux	RHEL 7	RHEL Client 7.9
Ubuntu	Ubuntu 18	18.04.6
SUSE Enterprise	SUSE Enterprise 15	15 SP3

For information on required shared libraries, see Section 3.4.

3.2.2 Granta MI version

Granta MI 2021 R1 or later, with Service Layer Interface 20/05.

3.2.3 Python

Python 3.7 64-bit or later.

3.3 Installation instructions

1. Before installing MI Scripting Toolkit, first uninstall any earlier versions:

```
$ sudo python3 setup.py uninstall
```

2. To install MI Scripting Toolkit, enter the following into the Linux command line (using the appropriate installer file for your Linux version):

```
$ pip install MIScriptingToolkit-3.0.xxx-py3-none-any.whl
```

Or, if you have multiple versions of Python installed:

```
$ python -m pip install MIScriptingToolkit-3.0.xxx-py3-none-any.whl
```

When installation is complete, the MI Scripting Toolkit for Python files are installed into the Python site-packages location.

3.4 Required shared libraries

In the tables in this section, the required shared library names are listed in the left-hand column; these usually represent symbolic links to shared libraries. The shared library may have a different name, containing a more specific version number, and so, to aid the diagnosis of potential problems, the full names of the shared libraries on our test systems are listed in the right-hand column.

3.4.1 Debian 9 shared libraries

Required shared library name	Full library name on our Debian 9.11 test system
/lib64/ld-linux-x86-64.so.2	/lib64/ld-linux-x86-64.so.2
libc.so.6	/lib/x86_64-linux-gnu/libc.so.6
libdl.so.2	/lib/x86_64-linux-gnu/libdl.so.2
libgcc_s.so.1	/lib/x86_64-linux-gnu/libgcc_s.so.1
libm.so.6	/lib/x86_64-linux-gnu/libm.so.6
libpthread.so.0	/lib/x86_64-linux-gnu/libpthread.so.0
librt.so.1	/lib/x86_64-linux-gnu/librt.so.1
libstdc++.so.6	/usr/lib/x86_64-linux-gnu/libstdc++.so.6
libuuid.so.1	/lib/x86_64-linux-gnu/libuuid.so.1
libz.so.1	/lib/x86_64-linux-gnu/libz.so.1
linux-vdso.so.1	(Virtual dynamic shared object)

3.4.2 CentOS 7 shared libraries

Required shared library name	Full library name on our CentOS 7.9.2009 test system
/lib64/ld-linux-x86-64.so.2	/lib64/ld-linux-x86-64.so.2
libc.so.6	/lib64/libc.so.6
libdl.so.2	/lib64/libdl.so.2
libgcc_s.so.1	/lib64/libgcc_s.so.1
libm.so.6	/lib64/libm.so.6
libpthread.so.0	/lib64/libpthread.so.0
librt.so.1	/lib64/librt.so.1
libstdc++.so.6	/lib64/libstdc++.so.6
libuuid.so.1	/lib64/libuuid.so.1
libz.so.1	/lib64/libz.so.1
linux-vdso.so.1	(Virtual dynamic shared object)

3.4.3 Red Hat Enterprise Linux 7 shared libraries

Required shared library name	Full library name on our RHEL 7.9 test system
/lib64/ld-linux-x86-64.so.2	/lib64/ld-linux-x86-64.so.2
libc.so.6	/lib64/libc.so.6
libdl.so.2	/lib64/libdl.so.2
libgcc_s.so.1	/lib64/libgcc_s.so.1
libm.so.6	/lib64/libm.so.6
libpthread.so.0	/lib64/libpthread.so.0
librt.so.1	/lib64/librt.so.1
libstdc++.so.6	/lib64/libstdc++.so.6
libuuid.so.1	/lib64/libuuid.so.1
libz.so.1	/lib64/libz.so.1
linux-vdso.so.1	(Virtual dynamic shared object)

3.4.4 Ubuntu 18 shared libraries

Required shared library name	Full library name on our Ubuntu 18.04.6 test system
/lib64/ld-linux-x86-64.so.2	/lib64/ld-linux-x86-64.so.2
libc.so.6	/lib/x86_64-linux-gnu/libc.so.6
libdl.so.2	/lib/x86_64-linux-gnu/libdl.so.2
libgcc_s.so.1	/lib/x86_64-linux-gnu/libgcc_s.so.1
libm.so.6	/lib/x86_64-linux-gnu/libm.so.6
libpthread.so.0	/lib/x86_64-linux-gnu/libpthread.so.0
librt.so.1	/lib/x86_64-linux-gnu/librt.so.1
libstdc++.so.6	/usr/lib/x86_64-linux-gnu/libstdc++.so.6
libuuid.so.1	/lib/x86_64-linux-gnu/libuuid.so.1
libz.so.1	/lib/x86_64-linux-gnu/libz.so.1
linux-vdso.so.1	(Virtual dynamic shared object)

3.4.5 SUSE Linux Enterprise 15 shared libraries

Required shared library name	Full library name on our SUSE Enterprise 15.3 test system
/lib64/ld-linux-x86-64.so.2	/lib64/ld-linux-x86-64.so.2
libc.so.6	/lib64/libc.so.6
libdl.so.2	/lib64/libdl.so.2
libgcc_s.so.1	/lib64/libgcc_s.so.1
libm.so.6	/lib64/libm.so.6
libpthread.so.0	/lib64/libpthread.so.0
librt.so.1	/lib64/librt.so.1
libstdc++.so.6	/usr/lib64/libstdc++.so.6
libuuid.so.1	/usr/lib64/libuuid.so.1
libz.so.1	/lib64/libz.so.1
linux-vdso.so.1	(Virtual dynamic shared object)