



Simufact Additive 2022.1 Installation guide

Installation, Running on Linux,
Configuration of compiler,
Troubleshooting, Support and Licensing

Copyright © 2022 Hexagon AB and/or its subsidiaries. All rights reserved.

This documentation, as well as the software described in it, is furnished under license and may be used only in accordance with the terms of such license.

Hexagon reserves the right to make changes in specifications and other information contained in this document without prior notice. The concepts, methods, and examples presented in this text are for illustrative and educational purposes only, and are not intended to be exhaustive or to apply to any particular engineering problem or design. Hexagon assumes no liability or responsibility to any person or company for direct or indirect damages resulting from the use of any information contained herein. This notice shall be marked on any reproduction of this documentation, in whole or in part. Any reproduction or distribution of this document, in whole or in part, without the prior written consent of Hexagon is prohibited.

This software may contain certain third-party software that is protected by copyright and licensed from third party licensors. Additional terms and conditions and/or notices may apply for certain third-party software. Such additional third-party software terms and conditions and/or notices may be set forth in documentation and/or at <http://www.mssoftware.com/thirdpartysoftware> (or successor website designated by Hexagon from time to time).

PCGLSS 8.0, Copyright © 1992-2022 Computational Applications and System Integration Inc. All rights reserved. Portion of this software are owned by Siemens Product Lifecycle Management Software, Inc. © Copyright 2022.

Trademarks

The Hexagon logo, Hexagon, MSC, MSC Nastran, Adams, Dytran, Marc, Mentat, Patran, Simufact, Simufact Forming, Simufact Welding and Simufact Additive are trademarks or registered trademarks of Hexagon AB and/or its subsidiaries in the United States and/or other countries.

Windows is a trademark or registered trademark of Microsoft Corporation in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

3Dconnexion and SpaceDevice are registered trademarks of 3Dconnexion Inc.

Nastran is a registered trademark of NASA.

FLEXIm and FLEXNet Publisher are trademarks or registered trademarks of Flexera Software.

Parasolid is a registered trademark of Siemens Product Lifecycle Management Software, Inc.

Magics and Materialize are trademarks or registered trademarks of Materialize NV and/or its subsidiaries.

All other trademarks belong to their respective owners.

Patents:

U.S. Patent No. 10,740,910

Contact:

www.simufact.com/contact

Table of Contents

1. Installation instructions	1
1.1. Installation using Windows™	2
1.1.1. Attended installation	2
1.1.2. Silent installation	8
1.1.3. Uninstallation	10
1.1.4. Queueing	10
1.2. Installation using Linux™	10
1.2.1. Quickstart	10
1.2.2. Folder structure	11
1.2.3. User subroutines	11
1.2.4. Running Simulations on Linux	11
1.2.5. For administrators	13
1.2.6. Queueing	16
2. Installation and configuration of compilers	17
2.1. Intel® Parallel Studio or Intel® oneAPI?	18
2.2. Intel Parallel Studio XE for Fortran	18
2.2.1. Requirements	18
2.2.2. Installation and configuration	19
2.3. Intel® oneAPI Compiler	22
3. Troubleshooting	23
3.1. Troubleshooting on Windows™	24
3.1.1. General hints	24
3.1.2. Status "Waiting for License"	26
3.1.3. Problems with Shared Libraries (DLLs)	26
3.1.4. Windows Remote Desktop and 3DConnexion SpaceDevices	27
3.1.5. The GUI does not appear after launching the program	27
3.1.6. CAD import fails	28
3.1.7. Initial meshing does not work	28
3.1.8. Simulation fails when the project is stored on a network drive	28
3.1.9. Simulation using DDM fail to start	28
3.1.10. Windows error dialog "Program has stopped working"	29
3.2. Troubleshooting on Linux™	29
3.2.1. Wrong information in INI files	29
3.2.2. Jobs fail to start	29
3.2.3. DDM jobs do not start	30
3.3. Log files	31
3.3.1. Simufact Additive GUI	31
3.3.2. License client	31
3.3.3. Result manager	32
3.3.4. Simufact Additive memory dumps	32
3.4. Support	32
A. MSC Licensing: Configuring and managing licenses	33
A.1. Configuring the license client	33
A.2. Firewall considerations	33
A.3. Updating the license file	34
A.4. Releasing Licenses	35
A.5. Combination with other MSC products	36
A.6. Combination with 3rd party products	36
A.7. Restrictions	37
A.8. Troubleshooting	37
A.9. Miscellaneous	38
B. Firewall: New rules for opening ports	40
C. Maintaining Environment Variables	43
C.1. Microsoft Windows™ 10	43
C.2. Linux™	45

D. Password-less login with <code>ssh</code>	46
E. Basic commands of the editor <code>vim</code>	48

1

Installation instructions



Before installing **Simufact Additive 2022.1**, please carefully read the "Release Notes" and the "What's New". They contain additional information which is not included in this installation guide. In particular:

- the system requirements for installing this software,
- known issues that may occur during installation and usage of this software.



Please keep in mind that **administrative rights** are required for a successful installation!



Before starting the installation make sure that you have a valid **license file** for the products you like to install. The recommended steps of the installation are:

- 1) get a valid license file
- 2) install the license server (see notes below)
- 3) install **Simufact Additive** (described in this manual)

Only in rare special cases, which are explicitly instructed in the delivery letter of the license file, step 2) "Installation of the license server" is not needed.

For network licenses the installation of the license server software is only needed on the machine used as license server, this means on the machine with the host-ID specified in the license file. This machine may or may not be used to run **Simufact Additive**, too. For nodelocked licenses the license server software and **Simufact Additive** are installed on the same machine.

The required license server software is "MSC Licensing", a FLEXlm based licensing software. Please see the "Release notes" for the required version of MSC Licensing. Typically several version of **Simufact Additive** can be used with a constant version of MSC Licensing. Thus there may be no need to update an existing MSC Licensing.

MSC Licensing is not shipped with **Simufact Additive**. It is available as separate download in the MSC Solution Download Center (SDC) where you have downloaded **Simufact Additive**. MSC Licensing is not installed by the **Simufact Additive** installer but has to be installed separately. For this please follow the instructions shipped with MSC Licensing. You need to remember the port number and host name of the license server shown respectively configured during the installation of MSC Licensing.

The installation of a license server is also not needed if you have the license server software "MSC Licensing" already running for other MSC products on the machine licensed for **Simufact Additive**. In this case the license files for the different products need to be combined, see Appendix A at the end of this document.

For more information about configuring and managing your licenses please see Appendix A at the end of this document.



Please note that there is a comprehensive **troubleshooting** chapter at the end of this installation guide.

1.1. Installation using Windows™

1.1.1. Attended installation

The **Simufact Additive 2022.1** download package contains an easy-to-use installer: "Setup.exe". The installation needs to be done with administrative rights because the Microsoft™ C++ and Intel™ Fortran Redistributable packages included are installed as needed in the system directories as recommended by these software vendors. Thus right click on "Setup.exe" and select **Run as administrator ...** to open installation wizard that will guide you through the installation. Double-click on the executable is only appropriate if you are logged-in as administrator.

The installer checks whether it is run with the appropriate administrative rights and aborts with a message if this is not the case.

1.1.1.1. Language selection for installation

In the first step of the installation you can choose the language used during the installation. Select the language of your choice and click **Next**. Please note that this selection does not influence the language used in Simufact Additive. The language used in Simufact Additive can be chosen in the settings of Simufact Additive.

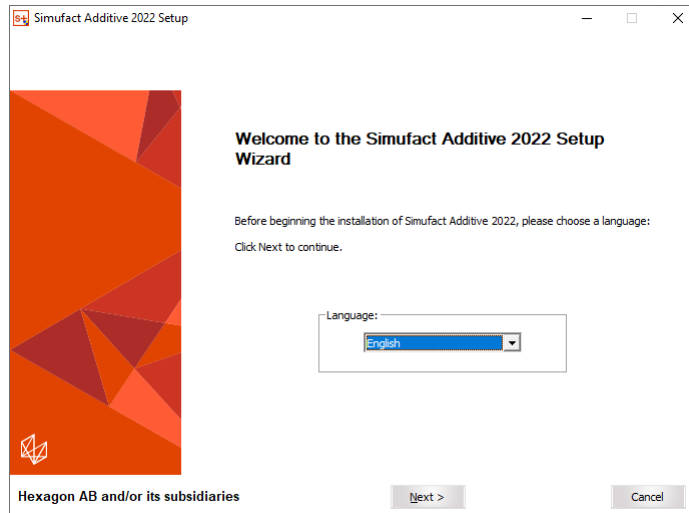


Figure 1.1. Language selection

1.1.1.2. License agreement

If you are installing a trial version of Simufact Additive:

To continue the setup, you must accept the Trial License Agreement by selecting **I ACCEPT** and clicking **Next**. If you do not accept the agreement, click **Cancel** to quit the setup. You can print the whole agreement with the **PRINT** button.

If you are installing a regular version of Simufact Additive:

To continue the setup, you must confirm the copyright notice by selecting **Acknowledged**. If you do not accept the agreement, click **Cancel** to quit the setup.

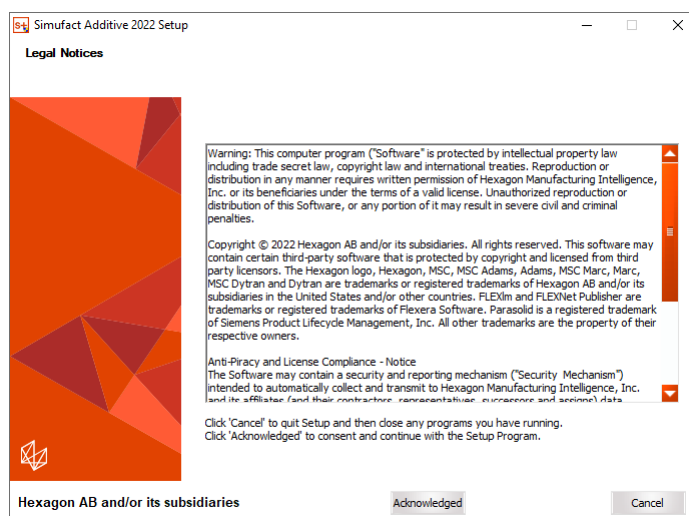


Figure 1.2. License agreement

1.1.1.3. Choose license location

You should have received a license file *before* running the setup or have a license server providing all necessary licenses. You need to specify the location of your license server you should have set up before. The location of the license is set in the system environment variable "MSC_LICENSE_FILE". On this page you can adapt the value of this variable.

For all counted licenses you have to set the right value to connect to your MSC Licensing license server. This is typically something like 'portnumber@hostname'. If you already have specified this with a previous setup you will see in the 'Current value' field previously entered data. If that is correct, you do not have to change anything. If the 'Current value' is empty or wrong, type in the correct value for MSC_LICENSE_FILE. If you do not have an MSC Licensing license server running on your system or in your network please install MSC Licensing on your machine used as license server or on your nodelocked licensed machine. MSC Licensing is a separate program, so a separate download and setup is necessary.

Only in rare special cases, which are explicitly instructed in the delivery letter of the license file, option 3) applies. **Only** in this cases use the **Browse...** button to directly select the license file on your system. The path and the name of the license file must not contain special characters like ; , : Å or ß, and must be shorter than 248 characters otherwise the license will not be found. One of this special cases are **Simufact Additive Trail Licenses** that have been issued using on a simplified, web based process. They can be identified by the word **uncounted** in the license file `license.dat`.

If the value of the environment variable MSC_LICENSE_FILE is not set correctly, you won't be able to start any Simufact program. If you're not sure what to do, consult your System Administrator. You can set the value also after the installation in the advanced system settings on your Windows™ operating system, compare Appendix C. For more information about configuring and managing your licenses please see Appendix A at the end of this document.

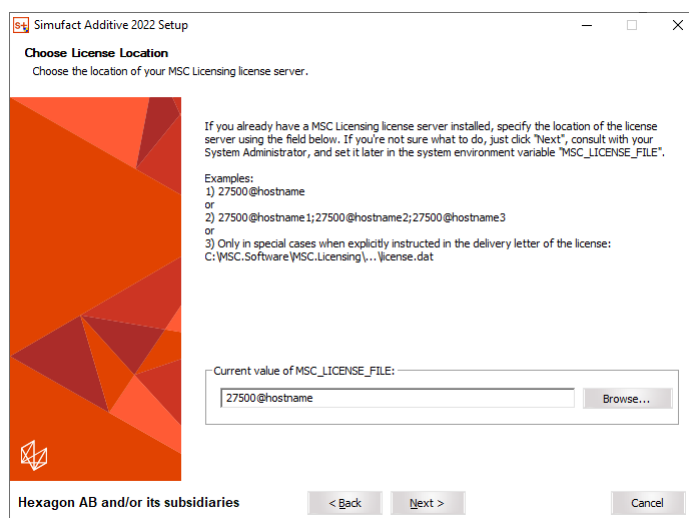


Figure 1.3. License location

1.1.1.4. Select target directory

Next you are asked to specify the location where you want to install the programs. If you have already installed an older version of Simufact Additive or other programs, then the destination folder is automatically set to the last installation directory. In this case, you should **not** change the folder to prevent conflicts.

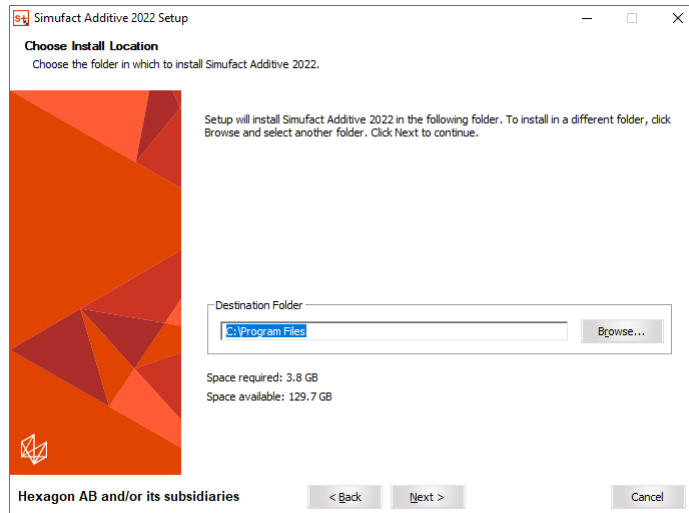


Figure 1.4. Installation location

You can select any folder on your computer. Spaces in the installation path are only supported if the optional additional 8.3 file names have been activated during the creation of the directory. If you see this message (Figure 1.5), please select another installation directory.

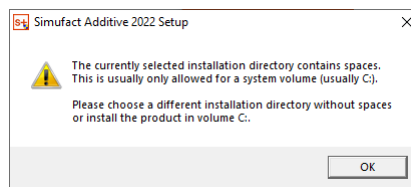


Figure 1.5. Spaces in installation path

1.1.1.5. Choose components to install

You will be asked which Simufact products you want to install.

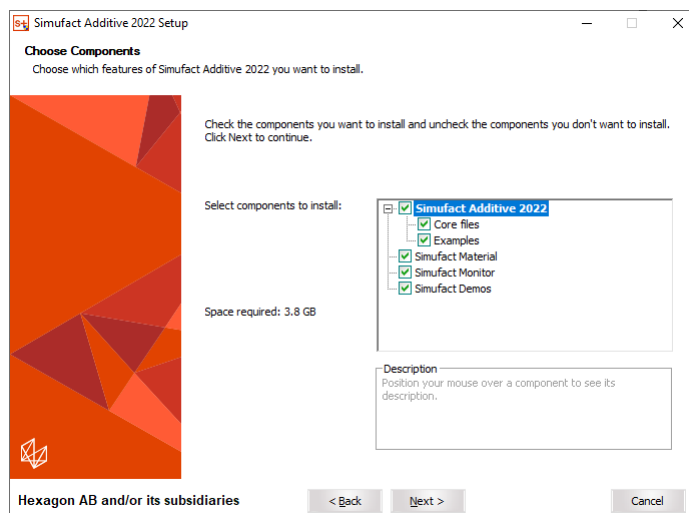


Figure 1.6. Choosing components

By default the following products are selected to be installed: Simufact Additive, Simufact Material and all Simufact Utilities. Additional programs or examples will not be installed if you deselect the corresponding option. After the installation the removal of specific components is possible as part of the uninstallation, see Section 1.1.3

The installation procedure will check if any programs are running that would conflict with the installation and will display a list of these programs. If you get this, close the mentioned programs before continuing the setup. Once the list of blocking programs is empty, press **Install** to start the installation.

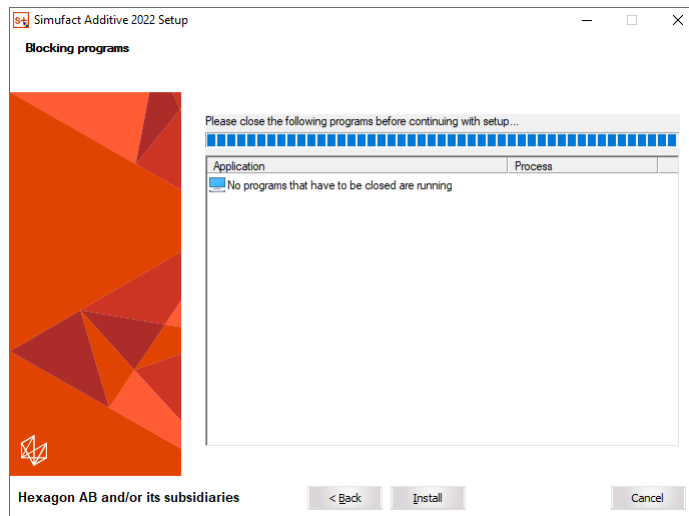


Figure 1.7. (Empty) list of blocking programs

Older versions of Simufact Material, Simufact Monitor and Simufact Demos in the same installation path are updated during the installation procedure. You will see a message about this, select **Yes** to continue. These utilities are shared between all Simufact programs and all versions of them. You should always use only the latest version. Do not try to run different versions of these utilities in parallel as this may result in different kinds of mal functionality.

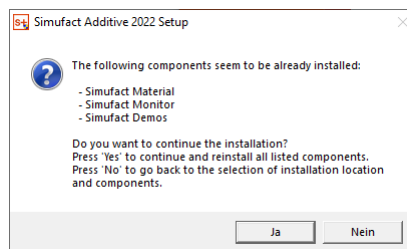


Figure 1.8. Reinstallation of utility program components

Simufact Additive 2022.1 can be used in parallel with previous versions. Please note that you cannot run different versions of Simufact Material, Simufact Monitor and Simufact Demos in parallel. Here we recommend using the latest version.

If you want to change something, you can go back to your previous selections by clicking **Back**. Otherwise start the installation by clicking **Install**. By clicking **Cancel** you can quit the installation.

1.1.1.6. Confirming the installation

The installation is fully automatic – the completion bar will inform you about the installation progress. The installation cannot be canceled anymore at this point.

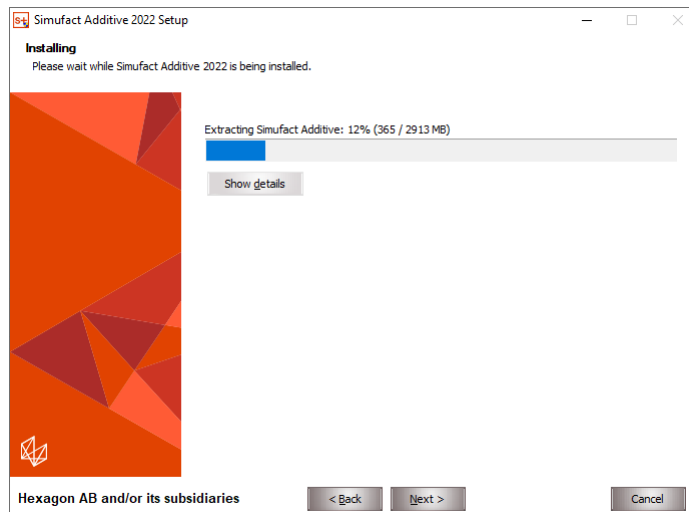


Figure 1.9. Installation progress

By default, the file extension `.amproj` is associated with Simufact Additive.

1.1.1.7. Complete setup

At the end of the installation you will be informed accordingly by the setup program. You only have to press **Finish** to end the setup.

The installation wizard closes and you can use the installed Simufact products. If the required Microsoft™ C++ and Intel FORTRAN™ Redistributable Packages have not already been installed on the machine before, the installer has installed them. **In this case you may be asked to reboot the machine once the installation has finished.**

If the required Intel FORTRAN™ Redistributable Packages have not already been installed on the machine before, the setup may not be able to perform all configurations completely. In this case a pop-up message will tell you "The installation of Simufact Additive finished. For further configuration please start Simufact Additive from the desktop after the setup is completed." If you get this message start Simufact Additive as administrator after the setup and potentially the reboot are completed and perform the settings described in Section 1.1.1.9 below for all users.

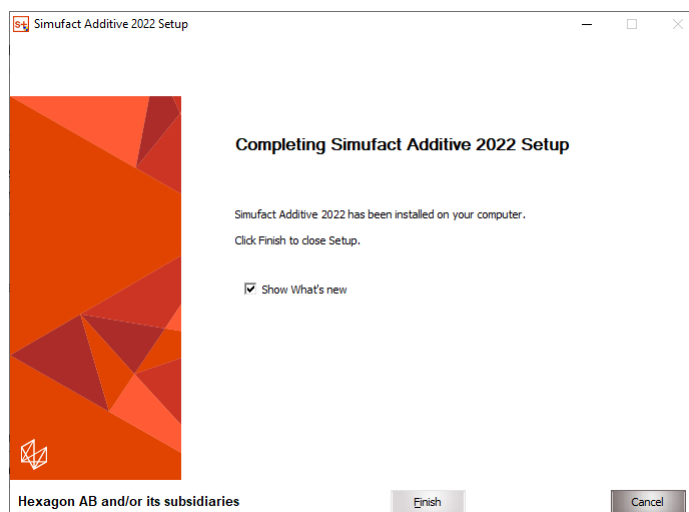


Figure 1.10. Installation finished

1.1.1.8. Show the What's new document

In case you haven't actively deselected the option **Show What's new**, the document `What's New` is shown after clicking **Finish**. It shows the fundamental program improvements. Enjoy reading!

1.1.1.9. User settings

When Simufact Additive or Simufact Material are started for the first time, they will ask you to perform initial setup actions. If you want to transfer the configuration of the previous version, just check the corresponding check box and the application will copy the existing settings into the .ini file of the new version.

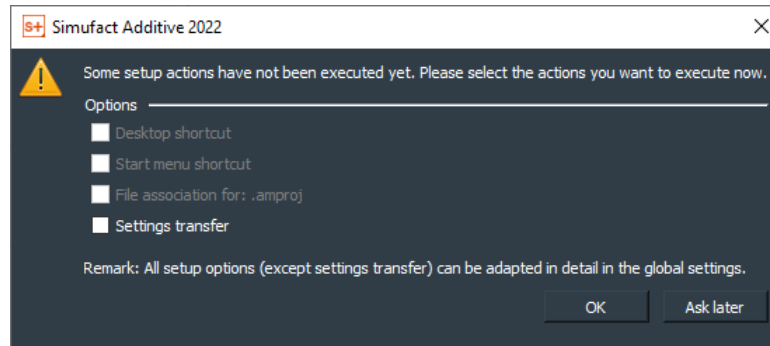


Figure 1.11. Setup options

Please note that version-dependent settings will be excluded. For example the location of a solver executable cannot be copied from the previous settings. The settings can be changed later in the global options of Simufact Additive, which may require to start the GUI as administrator to change or overwrite settings done for all users.

1.1.1.10. Configuration for 3DConnexion™ SpaceDevices

Simufact Additive supports the usage of 3DConnexion™ SpaceDevices. The sensitivity of the SpaceDevice can be adjusted in the setting of Simufact Additive. Typically no further configuration is required.

1.1.2. Silent installation

In order to install the software without any user interaction, for example on multiple machines using a script, the setup supports a silent installation mode. To use the silent mode, please open a command line and type the name of the setup followed by /S and the following valid parameters:

Table 1.1. Allowed parameters

Accept license notice	/AcceptLicenseNotice=yes
To install Simufact Additive	/Additive=yes
To install Examples	/Examples=yes
To install Simufact Demos	/Demos=yes
To install Simufact Material	/Material=yes
To install Simufact Monitor	/Monitor=yes
Installation path (must be the last parameter)	/D=c:\some folder

If the installation path is not provided, C:\Program Files will be used by default. If the parameter for a program is not specified, it will not be installed. The Microsoft Visual Studio™ and Intel FORTRAN™ Redistributable Packages will always be installed. Please note that the destination folder (for example C:\Program Files) must exist before the setup is started.

You must confirm the following copyright by adding /AcceptLicenseNotice=yes to install the software:

Warning: This computer program ("Software" is protected by intellectual property law including trade secret law, copyright law and international treaties. Reproduction or distribution in any manner requires written permission of Hexagon

Manufacturing Intelligence, Inc. or its beneficiaries under the terms of a valid license. Unauthorized reproduction or distribution of this Software, or any portion of it may result in severe civil and criminal penalties.

Copyright (C) 2022 Hexagon AB and/or its subsidiaries. All rights reserved. The Hexagon logo, Hexagon, and other logos, product and service names of Hexagon and its subsidiaries are trademarks or registered trademarks of Hexagon AB and/or its subsidiaries in the United States and/or other countries. All other trademarks belong to their respective owners. To confirm the copyright notice add the parameter `/AcceptLicenseNotice=yes` to the command line.

Examples:

- To install all programs to user dir:

```
setup.exe /S /AcceptLicenseNotice=yes /Additive=yes  
          /Examples=yes /Demos=yes /Material=yes /Monitor=yes  
          /D=C:\Users\\Program Files
```

- To only install Simufact Material:

```
setup.exe /S /AcceptLicenseNotice=yes /Material=yes
```

- To only install Simufact Utilities:

```
setup.exe /S /AcceptLicenseNotice=yes /Demos=yes /Monitor=yes
```

Normally the process `setup.exe` will return immediately if started from a script. You can circumvent that by starting it as follows:

```
start /WAIT setup.exe
```

with the options as given above.

A log file named `setupSimufact.log` will be created in your temporary directory (typically `C:\Users\\AppData\Local\Temp`) where the steps of the installation/uninstallation are recorded.

After the silent installation you need to review resp. set the environment variable `MSC_LICENSE_FILE` to configure your license. Typically it points to `portnumber@hostname`. See introduction remarks and Section 1.1.1.3 above as well as the Appendix A. To do so in a script you can use

```
setx /m MSC_LICENSE_FILE portnumber@hostname
```

to set this environment variable system wide (option `/m` requiring administrator privileges).

Another useful environment variable is `SF_LANGUAGE` which sets the language used in the graphical user interface. You may want to preset this for the user in the same way. Depending on the Simufact application used the following values may be supported: EN, DE, JA, PL, RU, TR, ZH-CN, ZH-TW.

Simufact applications have several configuration options, including for example the preferred unit system, custom databases used, colors, color legends, user defined result values or the possibility to display your company logo in the results. These and numerous other user preferences can be set in the graphical user interface and will be stored in ini-files in `%PROGRAMDATA%\Simufact` and in `%APPDATA%\Simufact`. Before you rollout the Simufact application you may want to configure a master installation manually and save the adapted ini-files to a template directory. The complete silent installation of Simufact Additive as part of your software rollout will then be done with a script that includes the following steps:

1. Run `setup.exe` in the silent mode.
2. Set `MSC_LICENSE_FILE`
3. Set `MSC_LICENSE_SF_WAITINGTIME` (compare Section 3.1.2)

4. Set SF_LANGUAGE
5. Copy the saved ini-files from your template directory to %PROGRAMDATA%\Simufact and %APPDATA%\Simufact.

1.1.3. Uninstallation

You can use the Windows Control Panel to uninstall **Simufact Additive**. Alternatively you can use the generated uninstaller `Uninstall Simufact Additive 2022.1.exe` located in the installation folder directly. You can select the components to be uninstalled.

The uninstaller supports a silent mode, too. For the silent mode start the uninstaller with the option `/S` using the command line. Please note that all components installed before will be removed. There are no options to select certain components in the silent mode of the uninstall.

Example:

```
"Uninstall Simufact Additive 2022.1.exe" /S
```

1.1.4. Queueing

If no more licenses are available, the simulations will not start or continue but reach the status "Waiting for License", see Section 3.1.2. This may be used to implement a simple, short-term quasi batch mode in all situations where licenses are shared between different users or machines. Note that this is *not a traditional queue*. The waiting simulations are collected in a pool and will be started in a random order that can not be influenced once a license is available again.

In a single user, single workstation environment the **Simufact Monitor** offers a simple but flexible batch queue for simulations, which is - in this environment - a better choice than using the status "Wait for license" as quasi batch mode.

1.2. Installation using Linux™



Only the solvers are part of the Linux version of **Simufact Additive**, you have to use the graphical user interface (GUI) on Windows to set-up your model, to write the input files for the solver and to do the post-processing. Please see Section 1.2.4 for how to start a simulation on Linux and for possibilities to couple the GUI on Windows with the solvers on Linux.

1.2.1. Quickstart

If you are a normal user, check with your administrator on step 3 and 4 and continue with step 5. Administrators start the installation with step 1.

1. Read the subsequent chapters of this document carefully.
2. Start the installation. For this, open a command shell (terminal), change into the directory that contains the **Simufact Additive** download package `simufact.additive.2022.1_linux.tar.tgz`. After extracting this archive, all required software components are already installed. You may want to extract this archive directly to the desired location:

```
tar xjvf simufact.additive.2022.1_linux.tar.bz2 -C /target/directory
```

Please check the owner and the permission of the installed files afterwards.

3. Set the environment variable `MSC_LICENSE_FILE` to point to your license. Typically it points to `portnumber@hostname`, but in special cases indicated in the delivery letter of the license file it may point directly to the license file. See introduction remarks and Section 1.1.1.3 above as well as Appendix A.

Use the means of your shell or your Linux distribution to permanently set the environment variable `MSC_LICENSE_FILE`. You may want to check the chapter "Invocation" (or similar) in the manpage of your shell.

System wide settings are preferred. For user specific settings typically shell specific initialisation files like `~/.profile`, `~/.kshrc` or `~/.bashrc` are used. The later one could for example include commands similar to:

```
MSC_LICENSE_FILE=portnumber@hostname1:portnumber@hostname2
export MSC_LICENSE_FILE
```

You have to reinitialise resp. to relogin to check the effect of modified shell initialisation files. The initialisation of environment variables can be tricky if Simufact simulations are started using your scheduling engine.

4. Optionally: Link the solver start scripts into a directory in the regular search PATH.
5. That's it! You should now be ready to use the software. As only the solvers are part of the Linux version of **Simufact Additive**, you have to use the graphical user interface (GUI) on Windows to set-up your model, to write the input files for the solver and to do the post-processing. Please see Section 1.2.4 for how to start are simulation on Linux and for possibilities to couple the GUI on Windows with the solvers on Linux.

1.2.2. Folder structure

After a successful installation, you can find the executable and the documentation in the following folders:

<code>simufact/additive/2022.1/sfMarc/tools/run_marc</code>	FE solver script
<code>Installation_guide_Simufact_Additive_en.pdf</code>	This guide

1.2.3. User subroutines

For creating user subroutines and integrating them to the solver used by **Simufact Additive** for FE simulations, you have to install the Intel(R) FORTRAN compiler in the version indicated in the [Release Notes](#). For **Simufact Additive** to be able to find your installation of the compiler, you may have to edit the file

```
.../simufact/additive/2022.1/sfMarc/tools/include_linux64
```

to match your system. Search for the definition of `INTELPATH` and adapt it as needed. If you have any `ifort` in your search PATH, this version will be used. Thus take care that the correct version or no version is found in PATH. See Chapter 2, too.

1.2.4. Running Simulations on Linux

As only the solvers are part of the Linux version of **Simufact Additive**, you have to use the graphical user interface (GUI) on Windows to set-up your model, to write the input files for the solver and to do the post-processing. There are 2 possibilities to start a simulation on Linux once the input files for the solver have been written by the GUI on Windows:

- **Manually** copying the input files to your Linux machine, manually starting the solver using the appropriate options and manually copying the result files back. The remainder of this chapter will focus on this. A small *solver input for testing is provided in the Linux installation and it is recommended that you do some manual tests* before you proceed with the next possibility.
- Use custom specific **Start Scripts** to couple the GUI on Windows with the Solvers on Linux. Samples are provided. See the relevant Infosheet which can be accessed from the help menu in the GUI. Start Scripts could for example be configured to send e-mails once a simulation finishes. BTW, Start Scripts work on a pure Windows environment, too, and could be useful on a single machine in special cases, too.



A small solver input called `am_testjob` for testing the FE solver is provided in `.../simufact/additive/2022.1/sfMarc/tools/test`.

The solver input consists of several files for each test.

To manually run a simulation on Linux you obviously need the input files for the solver. Initially this may be the just mentioned test files. For all real simulations you need to use the GUI on Windows to write the solver input files.

To do so start a job submission in the GUI but choose **Cancel** in the last dialogue. The solver input files are now in the `_Run_` directory of the process in the folder structure used to store the **Simufact Additive** project. You can use the option **Open process folder** in the right mouse menu of the process in the GUI to access it. A Simufact solver input typically consists of several files, thus copy the whole `_Run_` directory or *all* files in it to your Linux machine. After the simulation copy *all* files in the `_Run_` directory back while the GUI is closed. Once the GUI is opened again, the GUI will import the results and move them from `_Run_` to `_Results_`.

If the `_Run_` directory written by the GUI is already stored on your Linux machine using network drives on Windows, you can omit the copying and the GUI can remain open while the simulation is running. Results will be imported as they are calculated.

You can import simulation results in blank processes in the GUI by right clicking on the process and selecting **Import results**. You will be asked for a directory that contains the results directly or in sub-directories. While the results are imported, the needed components are added to the process. This possibility is especially useful to post-process simulation variants that have been created on the solver input file level.

The simulation itself is started using the solver start script `../simufact/additive/2022.1/sf-Marc/tools/run_marc`. Do not start the solver executable directly! The options needed by the solver start scripts can be taken from the file `run.bat` written by the GUI in the `_Run_` directory. They represent the settings needed by the model and the parallelisation set by the user in the GUI. They are independent of the operating system. Check the comments in the solver start scripts and the Technical Reference (VoA) about their meaning.

The most frequent used options of `run_marc` are:

- `-j jid`

`jid` ("jobID") is the name of the solver input dat-file without the extension `.dat`.

- `-ddm 1` (always 1 or completely omitted)

Activates DDM with global remeshing. Originally developed for **Simufact Forming**, but standard for all our applications. It should be used for **Simufact Welding** and **Simufact Additive**, too, because the Simufact QA is done with it. Requires `-nps`.

- `-nps d`

Sets the number of domains for DDM to `d`. All domains will run on the host used to start `run_marc` as long as not `-host hostfile` is used to determine some thing different. It is recommended to use `-host hostfile` only for very huge models and with a high number of domains because additional restrictions apply.

- `-nts t -nte t`

Define the total number of cores used over all domains for the matrix solution resp. the matrix assembly using shared memory parallelisation. Typically both numbers are the same. Can be use with or without `-nps`. If it is used with `-nps`, `t` must be a whole number multiplier of `d`. In the GUI the number of cores for each domain is entered. `t` is `d` time the number entered in the GUI. Typically the DDM is more effective as shared memory parallelisation. `-nts` is equivalent to `-nthread_solver`, `-nte` is equivalent to `-nthread_elements`.

- `-v y/n`

"Verbose" `y` (yes) is the default and will cause the `run_marc` script to ask for confirmation resp. completion of the input arguments. `n` (no) does not do this. Typically `-v n` is used in scripts.

- `-b y/n`

"Background" `y` (yes) will cause the simulation to detach form the calling shell and to log stdout and stderr to `jid.log`. For fast tests, in scripts and for queuing typically `n` (no) is used. This causes the job to run in the foreground and logging to stdout and stderr, which should be redirected to a file.

- `-dl y/n`

"Delete log", determines whether `jid.log` is deleted at the start of the simulation to remove the output of previous runs. `-dl n` should be used in combination with `-b n` and redirecting stdout and stderr to `jid.log` because otherwise `run_marc` will delete / clear the file written by its calling shell.

- `-iam sim`

Always required to indicate a Simufact simulation.

Example for starting simulations with increasing complexity are given in Section 3.2.2. The parallelisation options are discussed in more detail in Section 1.2.5.3. Simulations started by scripts or by a scheduling engine are typically started with some thing like

```
run_marc -j jid -v n -b n -dl n -ddm 1 -nps 4 \  
-nts 8 -nte 8 -iam sim > jid.log 2>&1
```

`\` and line-break inserted for readability, not to be included in the command line. `jid.log` is expected by the GUI with this name and thus should not be renamed to enable other output redirections or similar.

1.2.5. For administrators

1.2.5.1. Prerequisites and compatibility

For a complete installation of Simufact Additive 2022.1 you need to have the following basic packages installed in your system:

- The shells `/bin/sh`, `/bin/bash` and `/bin/ksh`.
- GLIBC v2.17 or higher.

You should be able to get Simufact Additive 2022.1 up and running on the *x86-64* (sometimes called *amd64* or just *x64*) versions of

- Redhat Enterprise Server 7.3

or later versions with a higher number.



For process parallelization with *Domain Decomposition (DDM)* in Simufact Forming the default MPI version requires that the `ssh` command has been set up such that it does not prompt for a password if the local host and - if distributed domains are used - all involved hosts are accessed. See Appendix D how this is established.

More information about MPI can be found on the Intel(R) website at <http://software.intel.com/en-us/articles/intel-mpi-library-documentation/>.

1.2.5.2. Basic installation

See Section 1.2.1

1.2.5.3. Scheduling of jobs

Simufact Additive 2022.1 for Linux™ can be used with a scheduling engine. This functionality was developed and tested with the Sun Grid Engine (SGE) in mind, but you should be able to get it running with other scheduling engines as well.



The configuration and scripts discussed in this chapter focus on the requirements of the functionalities used in the sample **Start Scripts** provided.

The basis of this functionality are two files. In order to enable scheduling on the server, you have to edit these two files. The most common changes for SGE are already in these files albeit commented out. If you're using SGE, you mostly just have to delete the “#” characters at the beginnings of the lines between “START SGE” and “END SGE”. Please take care of the environment variable “MSC_LICENSE_FILE” in `write_job`, and adjust it to match your local settings. If you want to use a different scheduling engine, you may have to adjust these sections to suit your needs. The files are as follows:

.../simufact/additive/2022.1/sfTools/sfScheduling/write_job

This small shell script writes the jobfile for the scheduling engine. The jobfile is submitted to the scheduling engine to run the job. The jobfile contains the actual start command for the solver and settings / configurations needed for the scheduling engine used. Please make sure to delete a file named `RUNNING` after you called the solver. Before starting the solver, the jobfile writes the `hostfile` specifying the hosts used to run a simulation distributed to different hosts / nodes of a cluster using parallelization with Domain Decomposition (DDM). If Domain Decomposition is used, but all domains remain on the same host, a `hostfile` is not needed. In this case take care that the script removes the `-host hostfile` from the command line of `run_marc` given by **Simufact Remote**. If `-host hostfile` remains in the command line, a reasonable `hostfile` must be created. It can contain only one host without problems. `write_job` uses the name of the jobfile as first argument and the command line to start the solver as second argument.

.../simufact/additive/2022.1/sfTools/sfScheduling/schedule_job

This shell script submits the job (using the jobfile written by `write_job`) to the scheduling engine. It will be entered in the queue. You may specify further options in this script to adjust the way new jobs are entered into the queue. Please make sure to create a file named `RUNNING` after you queued the job. If **Simufact Remote** is used, it may be needed to initialise the environment of your scheduling engine in this script, too. `schedule_job` uses the number of `cpu` requested / used as first argument and the name of the jobfile as second argument.

Additional information about these scripts can be found in the files themselves. To make it easier to understand how these files work together, see the following sample script for scheduling a simulation (the line breaks are only inserted for this documentation and the paths have been shortened). For manually scheduling simulations from the command line the example script `schedule_marc.sh` is provided in `.../simufact/additive/2022.1/sfTools/sfScheduling/testscripts`. They use the same combination of `write_job` and `schedule_job`.

```
#!/bin/sh
cd RUNDIR

".../simufact/additive/2022.1/sfTools/sfScheduling/write_job" \
  "RUNDIR/_jid.job" \
  ".../simufact/additive/2022.1/sfMarc/tools/run_marc" \
  -j jid -host hostfile -ddm 1 -nps d -nts t \
  -nte t -v n -b n -dl n -iam sim >> jid.log 2>&1"
".../simufact/additive/2022.1/sfTools/sfScheduling/schedule_job" \
  d "RUNDIR/_jid.job"

while [ -f RUNNING ]; do
  sleep 5
done
cd /
```

For the abbreviations used, see Section 1.2.4. The file `RUNNING` keeps the script running while it is waiting for the job to be started by the scheduling engine and while it is being calculated. This is needed to keep track of the job. `RUNNING` is written by `schedule_job` and deleted at the end of the jobfile written by `write_job`. If **Start Scripts** are used, the GUI will assume that the simulation has stopped if the Start Script exits for any reason. Thus your Start Script must not exit after the job is submitted to the scheduling engine but must wait for it to finish for example by checking for `RUNNING` periodically.

Now you're all set to run scheduled jobs on your Linux™ machine.

Another thing to pay attention to is licensing in combination with a scheduling engine. If you don't configure your scheduling engine to respect the number of available licenses, it may schedule jobs for which not enough licenses are available. To circumvent these issues, you may be able to configure the licenses as additional resources (comparable to computing slots or system load) which your scheduling engine manages. That way no jobs will be started if no more licenses are available.



If no more licenses are available, the simulations will frequently do not exit with a license failure, but go into status "Waiting for license", see Section 3.1.2. This will probably break the objective of the scheduling engine.

The scheduling scripts shipped with Simufact Additive and discussed before do not reflect the whole complexity of scheduling simulations with limited hardware and license resources. Here a summary of the things to be considered:

Basics:

- `write_job` has 2 arguments:
 1. the name of the jobfile, a script to be given to the scheduling engine later.
 2. the command line for the solver a resulting from the settings made in the GUI.
- `schedule_job` has 2 arguments:
 1. the requested number of domains to be given to the scheduling engine.
 2. the jobfile given to the scheduling engine for being started.
- `hostfile`
 - only needed for FE jobs with DDM distributed to different hosts. This is only reasonable for selected matrix solvers. The used matrix solver can not be determined from the command line.
 - written by the jobfile based on the output of the scheduler and the configuration in `write_job`.
 - contains lines of `hostname number_of_domains`.
 - if jobs are not distributed to different hosts, `write_job` can remove `-host hostfile` from the solver command line given as argument (but does not have to, if the option is given a correct host file must be provided in any case).

Job types and parallelisation options:

	Number of Domains	cores for shared memory parallelisation	-host hostfile needed?	number given to <code>schedule_job</code>	total number of cores used (maximum)	can be distributed to different hosts
FE, no parallelisation	-	-	no	1	1	no
FE, shared memory parallelisation	-	as per <code>-nts</code>	no	1	as per <code>-nts</code>	no
FE, domain decomposition (DDM)	as per <code>-nps</code>	-	yes (if distributed to different hosts)	same as per <code>-nps</code>	as per <code>-nps</code>	yes
FE, DDM and shared memory parallelisation	as per <code>-nps</code>	as per <code>-nts</code>	yes (if distributed to different hosts)	same as per <code>-nps</code>	as per <code>-nts</code>	yes, in packages of Number of Threads /

						Number of Domains cores
--	--	--	--	--	--	-------------------------------



Simufact's sample configuration and the discussion above focus on using the domain decomposition method (DDM) for parallelisation and assume the typically alternating cpu load of shared memory parallelisation as overload above the scheduled cpu cores. A more precise scheduling for a distributed environment needs to consider both, the number of domains (d) and the total number of cpu cores (n). In an environment not distributing one simulation on different hosts, you may consider n as request to your scheduler to use a more conservative load setting.

Short method to figure out the maximum number of cores to be considered by the scheduling engine:

- if `-nts` is given, use this number.
- if `-nts` is not given, but `-nps` is given, use this number
- if neither `-nts` nor `-nps` are given, use 1

Short method to figure out the licenses needed:

- 1 `SF_ADDITIVE_SOLVER` for each job
- "total number of cores used (maximum)" - 1 `SF_ADDITIVE_NODE` for jobs using parallelisation
- if used: licenses for additional modules, can not be determined from the command line. Typically Simufact Additive is distributed with the same number of solver licenses and licenses for additional modules.

Notes:

- Shared memory parallelisation typically shows varying CPU usage with up to the number of cores specified. The cores are typically not constantly loaded.
- Distributing a DDM job on different hosts adds extra complexity. Depending on the size of the models, the parallelisation used, the maximum reasonable parallelisation and the computer resources, it should be considered not to use this option. If it is used, a common working directory shared on all hosts under the same path is needed.
- The meshers and other auxiliary programs are always run on the host used to start the job (by the scheduling engine), not on the hosts specified in the `hostfile`.

1.2.5.4. Uninstallation

Simufact Additive 2022.1 is basically a self-contained package, i.e. all required files and executable are stored in the "simufact" folder created during the installation.

1.2.6. Queueing

If no more licenses are available, the simulations will not start or continue but reach the status "Waiting for License", see Section 3.1.2. This may be used to implement a simple, short-term quasi batch mode. Note that this is *not a traditional queue*. The waiting simulations are collected in a pool and will be started in a random order that can not be influenced once a license is available again. In a HPC environment the usage of a scheduling engine is more recommended, compare Section 1.2.5.3.

2

Installation and configuration of compilers

Simufact Additive 2022.1 supports user subroutines written in the FORTRAN programming language. The support is not given in the GUI, but only in the solver. Thus, in order to use user subroutines you have to manually start your simulation and add `-u FortranFile` as argument to `run_marc.bat` resp. `run_marc`. `FortranFile` is the name of the file containing your code without the extension `.f`. It is not necessarily the name of the user subroutine.

2.1. Intel® Parallel Studio or Intel® oneAPI?

To use user subroutines written in FORTRAN programming language, you need to install a FORTRAN compiler on your machine. The required compiler is neither shipped nor installed nor licensed with Simufact Additive. You have the choice between 2 different compilers:

- The Intel® Parallel Studio XE Composer for Fortran.

This compiler is used during development and build of Simufact Additive. All quality assurance is done with software using this compiler. Because of this it is the recommended compiler for user subroutines. The main disadvantage is that for each version of Simufact Additive only one specific compiler version is supported as specified below.

- The Intel® oneAPI Compiler.

This compiler is a more flexible alternative as always the latest version of it can be used. However there is less experience with it and no performance studies have been done with Simufact Additive when using this compiler.

The installation of both compilers is described below. If both, Parallel Studio XE Composer and oneAPI are installed, the Composer will be used to compile user subroutines.

To **test your compiler installation** you can use the small test routine `compile_test.f` shipped in `C:\Program Files\simufact\additive\2022.1\sfMarc\tools\test` and in the corresponding directory of the Linux installation. You can use this routine with all Simufact simulations. In **Simufact Forming** you can specify the user subroutine to be used in the Forming Control under **Advanced \ User-defined \ Used Program**. In **Simufact Additive** and **Simufact Welding** you need to manually copy `compile_test.f` into the `_run_` directory (the working directory of the solver) after the solver input is written by the GUI and before the solver is started. Additionally you need to edit the `run.bat` file in the `_run_` directory and add `-u compile_test` to the call of `run_marc.bat` (`run_marc` on Linux). If the test routine is compiled and used correctly, the simulation will immediately stop showing an exit 3020 and descriptive messages in the log and out file. If the simulation aborts with any other message or continues to run after the 1st increment, the compilation of the routine was not successful and needs to be debugged.

2.2. Intel Parallel Studio XE for Fortran

2.2.1. Requirements

To use user subroutines with Simufact Additive 2022.1, additional programs are required.

For Windows™:

- Intel® Fortran Compiler Version 19.1.3.311
- Microsoft™ Visual Studio™ 2019

For Linux™:

- Intel® Fortran Compiler Version 19.1.3.304

Visual Studio™ on Windows™ provides linker support during the compilation of subroutines. The required compiler version is released by Intel® as **Intel Parallel Studio XE 2020 Update 4** as mentioned in Table 2.1.

Table 2.1. Compiler required for user subroutines in Simufact Additive 2022.1

Intel Parallel Studio XE 2020	Windows™ version / build	Linux™ version / build
Update 4	19.1.3.311 (20200925)	19.1.3.304 (20200925)

The user can choose the compiler version from a drop-down menu from the Intel® compiler download web page.

2.2.2. Installation and configuration

On Windows™ platforms, it is recommended to first install Visual Studio™, followed by the Intel® Fortran compiler. This will help the Intel® compiler's installer script to detect the correct installation path of Visual Studio™, in case it is installed at a non-default location. After the installation of above listed program(s), user subroutines can be used to compile and run jobs in Simufact Additive 2022.1. Multiple versions of the compiler and / or Visual Studio™ can be installed on the same machine. This might require manual changes to the Simufact compiler's script as described in Section 2.2.2.1.1.

Install Visual Studio™ with the components as shown in the figure below. Manually select the C++ Universal Windows Platform tools.

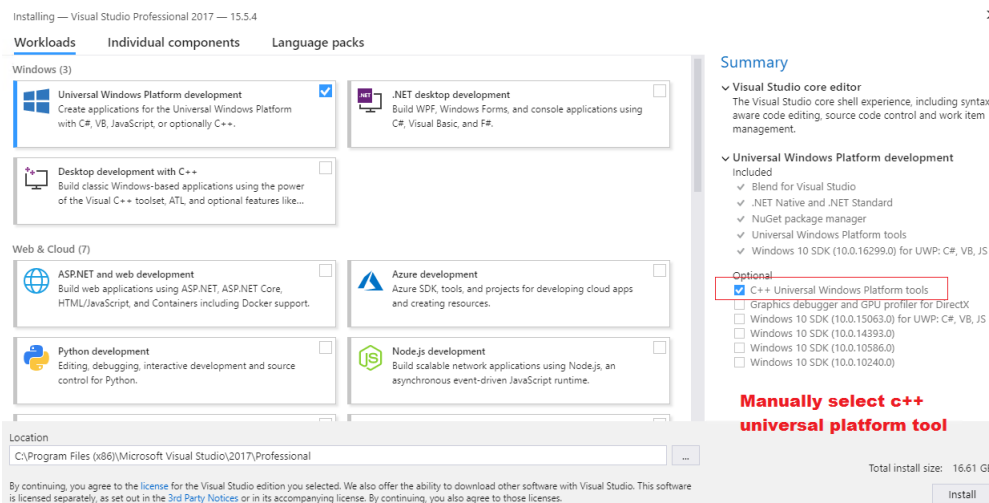


Figure 2.1. C++ Universal Windows Platform tools for Visual Studio™

When installing the Intel® Fortran compiler select **Customize** in the **Installation Summary** dialogue and select **Cluster support for Fortran** during the selection of the components to be installed. By default this option is unchecked and will not be installed. It is needed for parallel (DDM) running of Simufact jobs with user subroutine, even on a single workstation. Selecting only **Intel® 64** as target platform architecture is enough to compile user subroutines for Simufact applications.

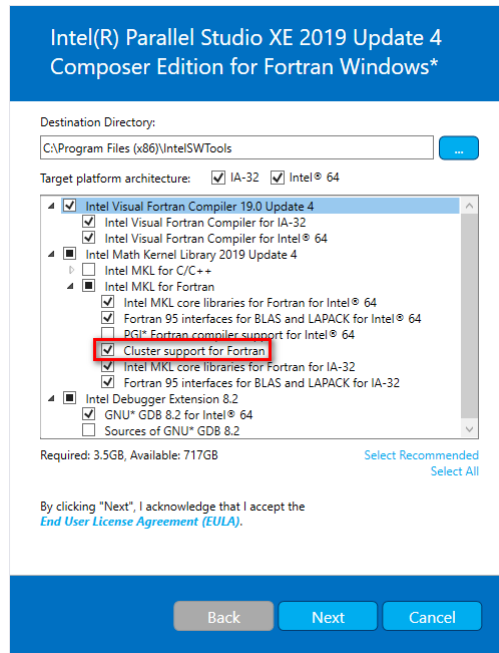


Figure 2.2. Cluster support for Intel®64 required for parallelization support

Keep the default installation folder, otherwise Simufact's compiler script needs to be adapted as described in Section 2.2.2.1.1. Make sure that the **Microsoft Visual Studio™ 2019 software integration** is selected. This should be the default, if not something went wrong.

Logout and re-login into Windows to complete the installation.

2.2.2.1. Debugging compiler problems

If the compilation of user subroutines does not work, please check the following point:

1. Make sure to keep the recommended order for installation:
 - i. Simufact Additive
 - ii. Visual Studio
 - iii. Compiler

It is recommended to use the default directories:

- C:\Program Files\simufact
 - C:\Program Files (x86)\Microsoft Visual Studio
 - C:\Program Files (x86)\IntelSWTools
2. Make sure to correctly customize the settings during installation:
 - For Visual Studio:
 - Activate C++ UWP
 - For Fortran Compiler:
 - Activate Cluster Support for Fortran
 - Activate Visual Studio Integration

3. Make sure to correctly adjust the `include_win64.bat` script in `C:\Program Files\simufact\additive\2022.1\sfMarc\tools`:

You have to correctly adjust the `INTELPATH` variable in this file, so that the compiler can be found.

4. Make sure to check the environment variables:

Recommended is the following setup:

- The linker not included in the path variable
- The compiler not included in the path variable (`INTELPATH` is already set in `include_win64.bat`, above).
- Installation directory of Visual Studio included in the environment variables, e.g. "VS2017INSTALLDIR" for Visual Studio 2017

Some typical problems are discussed in more detail in the next chapters.

2.2.2.1.1. Compiler not found

Intel® ifort Fortran compiler is installed however Simufact Additive 2022.1 reports it as missing. Trying to run a job with a subroutine will output the following message in the job log file:

```
'ifort' is not recognized as an internal or external command, operable
program or batch file.
```

For Simufact Additive to be able to find your installation of the compiler, you may have to edit the include-file

```
C:\Program Files\simufact\additive\2022.1\sfMarc\tools\include_win64.bat
```

on Windows resp.

```
.../simufact/additive/2022.1/sfMarc/tools/include_linux64
```

on Linux to match your system. Search for the definition of `INTELPATH` and adapt it as needed. On Windows you will additionally find that `%INTELPATH%\compilervars.bat` is called with the argument `vs2019`. This argument loads the Visual Studio™ 2019 environment variables.



If you have any `ifort` in your search `PATH`, this version will be used. It is recommended that the compiler is **not** included in `PATH`, otherwise `compilervars.bat` will be skipped, which may lead to a wrong compiler configuration.

2.2.2.1.2. Linker not found

If you get an output in the log file like:

```
'LINK' is not recognized as an internal or external command
```

This means that the linker could not be found after the executable was build. In some Visual Studio™ versions the path to the linker needs to be configured manually. Please also check if your Visual Studio™ version is supported by the script `compilervars.bat` (called by the include-file) and that the correct arguments are passed to it.

2.2.2.1.3. Link error: unresolved externals

Even if all the additional programs are installed, running a subroutine-based job might yield errors like:

```
mdsrc.lib(run_exe.obj) : error LNK2019: unresolved external symbol
__intel_sse2_strcpy ...
```

```
sfclib.lib(sfjoining.obj) : error LNK2019: unresolved external symbol
"__declspec(dlli ...
```

```
sfclib.lib(sfjoining.obj) : error LNK2019: unresolved external symbol
"private: static ...
```

These errors are generally due to one of the following reasons:

- An older or newer version of the Intel® Fortran compiler is loaded by the Simufact compiler's script. Make sure that the correct compiler version is installed and the correct path is mentioned in the include-file, see Section 2.2.2.1.1.
- A wrong version of the Visual Studio™ environment is loaded by the Simufact compiler's script. Installing the correct version of Visual Studio™ will help to get rid of the error.

Other versions of Visual Studio™ may also be used provided all the run-time dependencies are met by installing proper **Redistributable Packages**. But this approach is not supported by Simufact.

2.3. Intel® oneAPI Compiler

Please use the following step for the installation in the given order. It is recommended to use all proposed standard installation paths.

1. Install Simufact Additive 2022.1
2. Microsoft Windows™ only: Install Microsoft™ Visual Studio™ 2019. Select the **Universal Windows Platform Development** components to be installed. Additionally, manually select the optional **C++ Universal Windows Platform** tools to be installed.
3. Download the latest version of the Intel® oneAPI Base Toolkit. Only the installation of the **Intel® oneAPI Math Kernel library** package is needed. During the installation select to integrate with the Microsoft™ Visual Studio™ 2019 IDE. If Visual Studio is not yet installed, you need to run the oneAPI toolkit installer later again to integrate with the installed Visual Studio.
4. Download the latest version of the Intel® oneAPI HPC Toolkit. Select **Intel® Fortran Compiler & Intel® Fortran Compiler Classic** to be installed.

If the compilers are installed in a non-default path, then the script

```
C:\Program Files\simufact\additive\2022.1\sfMarc\tools\include_win64.bat
```

needs to be modified for the correct installation path:

```
if "%use_oneapi%"=="yes" (
call "C:\Program Files (x86)\Intel\oneAPI\setvars.bat" > NUL
for %%i in (ifort.exe) do set COMPILER=%%~$PATH:i
)
```

In case there is still a compiler missing error message, remove the `> NUL` in the code above and resubmit the simulation. The log-file or the `jobname_std.log` file of the simulation will now contain the messages from the initialisation of the Intel® oneAPI compiler, hopefully showing some meaningful hints for the reason of the compiler missing error message.

For further debugging information please see the chapter Section 2.2.2.1 above, too.

3

Troubleshooting



A list of known issues of this version of Simufact Additive is given in the [Release Notes](#). Please review this list as part of your troubleshooting activities.



For troubleshooting license issues, please see Appendix A which describes the configuration, managing and troubleshooting of MSC Licensing, the license system used for Simufact Additive.

3.1. Troubleshooting on Windows™

3.1.1. General hints

3.1.1.1. Issues with the display of the graphics, e.g. missing colors

As stated in the [Release Notes](#), Simufact Additive needs OpenGL to work correctly, otherwise different malfunctions of the graphical display may occur, for example missing colors, windows not opening, missing texts in the graphics windows, and similar. If you experience this issues, check the specification of your display adapter and check and update its driver. If your computer has multiple display adapters ensure that the `simufact_additive.exe` is always completely handled by the correct display adapter. In Windows 10™ this is configured under Settings / System / Display / Graphics settings / Graphics performance preference. The software of your display adapter may have additional settings.

Intel™ (U)HD onboard / onchip graphic adapters, which are frequently used in laptops, are known to cause various errors and are strictly not supported. The errors are frequently well hidden and hard to detect for new users. Beside other, the menus after right mouse click can be shorted not offering all options and the picking of points can be blocked. If you can not remove the Intel adapter from your system or deactivate it in the BIOS of the machine and you have a second display adapter in your system, configure it to be used (see above).

Typically Simufact Additive works smooth in Remote Desktop environments, in particular with the standard Windows Remote Desktop. However issues with the display of graphics are more likely to happen and harder to debug in these environments, especially when combined with virtual machines. Please contact your local IT when you experience issues in these environments.

If you can not solve the display issues by other means, you can try whether it helps to set the environment variable `QT_OPENGL` to `angle` (in small letters). This will enable an internal conversion of OpenGL to DirectX. Please expect a significant performance decrease with this. An alternative value of `QT_OPENGL` is `desktop`. Please see Appendix C for setting environment variables.

3.1.1.2. Windows are not shown on the screen

Under certain conditions it can happen that windows, like for example the clipping plane or the measurement dialogue, are not shown when they should open when the corresponding function is used in the GUI. Frequently this is related to changing screen resolutions or number of screens, interim remote desktop usage or similar. Typically the window is open, but "shown" outside the currently visible screen area. A few things can be done to display these windows correctly:

- Immediately after the not shown window has opened, press "Alt + Space". This will open the window menu. Press "arrow down" once to select **move** and confirm with "enter". Use the arrow keys to move the window to a visible screen area and confirm with "enter". It may be useful to test this Microsoft Windows™ function with a visible window first.
- Immediately after the not shown window has opened, press "Windows Key + Shift + (left or right) arrow key", repeat as needed. This will move the window from one screen to the next. It may be useful to test this Microsoft Windows™ function with a visible window first.

- In the settings of Simufact Additive navigate to **General - Message / dialog positions** and press **Reset all**. Try again afterwards.
- Close all GUIs of Simufact Additive and delete `view.ini` and `additive_windows.ini` from `C:\Users\\AppData\Roaming\simufact\simufact.additive_2022.1 (%appdata%)`. Open the GUI again and try again afterwards. Regarding ini files compare Section 3.1.1.5 below.

3.1.1.3. Energy saving mode

Some calculations can sometimes take place over night. So if the energy saving mode is activated, the machine will go into standby mode after a certain time before the calculation has finished and thus the job is aborted. The suspend energy saving mode may not abort simulations as they will recover once the computer wakes up again. But for sure there will be no progress of the simulation during the suspend mode. The energy saving modes are triggered only by the user activity, not by the CPU activity. It is recommended to deactivate the energy saving mode completely on a machine used to run simulations.

3.1.1.4. Windows™ updates

It is recommended to disable automatic system updates, because the related reboot aborts all running Simufact simulations. In Windows 10 Pro™ you can use the Group Policy Editor for this:

1. Press Windows™ key + R to gain access to the Run dialog and type `gpedit.msc`.
2. Navigate to the following directory: `Computer Configuration \ Administrative Templates \ Windows Components \ Windows Update`.
3. Double-click on "Configure Automatic Updates" in the right hand panel.
4. Set the configuration to Enabled on the left-hand side and then choose "Notify for download and notify for install" from the drop-down list under the Options header.

3.1.1.5. INI files

INI files are created for Windows™ in

- `C:\Users\\AppData\Local\Simufact (%localappdata%)`,
- `C:\Users\\AppData\Roaming\Simufact\ (%appdata%)`,
- `C:\ProgramData\Simufact\ (%programdata%)`.

They contain paths (e.g. for your company logo to be displayed in the results window), settings and other information. In case of problems with these settings, please check the contents of these INI files and - if necessary - edit them manually. If your setting files have been seriously corrupted for some reason, you can delete them or rename them to keep copies. If Simufact Additive is started without INI files, INI files with all default settings will be created.

If the Simufact Remote service is running as user SYSTEM, which is typically the case, INI files can be found in the subdirectories of `C:\Windows\System32\config\systemprofile\AppData` and of `C:\Windows\SysWOW64\config\systemprofile\AppData`.



The INI files are not deleted during un-installation and will not be reset during a re-installation of Simufact Additive. Thus, if you consider re-installing Simufact Additive to fix some issues, consider deleting the INI-files, too.

3.1.1.6. Firewall, ports and MPI

The firewall blocks some ports which are needed for Simufact™ services such as the remote service, licensing and the MPI service. So if you do not want not deactivate the firewall completely, please make sure that the ports 9985 (Simufact Monitor), 9987, 9988 and 9990 (Simufact Remote) as well as the ports used for MSC Licensing are not blocked. Compare, Appendix B too.

3.1.1.7. Antivirus software

Antivirus software sometimes can hinder some subprograms of Simufact™ to be installed, so it is strongly recommended to deactivate all antivirus software during the installation.

Antivirus software may slow down simulations considerably. Especially avoid that all files read and written on the local disk are scanned for viruses. The solvers and the GUI have a huge file I/O.

Some times antivirus software blocks files while they are checked. This may lead to random crashes of Simufact Additive, in particular of the solvers. If you experience such problems, deactivate your antivirus software.

3.1.1.8. Dell™ backup and recovery

Beside its high usage of CPU, the Dell backup stops some services (possibly the remote service or the license server) during the backup process, so it is recommended to deactivate this software.

Some versions of the Dell backup and recovery software conflict with Simufact applications causing these to crash, especially during "file open ..." dialogues. It is recommended to uninstall the Dell software in this case. See the *Release Notes* for more details.

3.1.2. Status "Waiting for License"

If a license can not be granted Simufact Additive and especially the solver will frequently not exit with a license failure, but go into the status "Waiting for license". The environment variable `MSC_LICENSE_SF_WAITINGTIME` is used to control the maximum duration of this status. The value of this variable is the time in minutes after which the solver exits with license failure if no license is granted in the meantime. The permissible range are intergers between 0 and 43200 minutes (30 days). The default is 4320 minutes (3 days). See Appendix C for how to set and change environment variables.

While this prevents from simulations to abort for example in the case of network troubles and while this may be used as a simple batch queue, it may prevent license problems to become obvious at once. In the status line of the GUI the status "Waiting for license" will be displayed. The log-file of the solver will show repeated messages about the license query. In the working directory of the solver a file named `jobname_pid.loc` will indicate the status "Waiting for license". The content of this files shows the required licenses. Deleting `jobname_pid.loc` will stop the simulation. This is done during a smart stop from the GUI, too.

In a single user, single workstation environment the Simufact Monitor offers a simple but flexible batch queue for simulations, which is - in this environment - a better choice than using the status "Wait for license" as quasi batch mode.

3.1.3. Problems with Shared Libraries (DLLs)

Simufact applications relay on Microsoft™ C++ and Intel™ Fortran Redistributable packages providing a wide range of DLLs. These redistributable packages are installed during the installation of Simufact Additive. If you experience problems that mention some DLL, it is likely that this installation has failed or that the redistributables have been overwritten by the later installation of a 3rd-party application.

Note: Not in all cases error messages mentioning DLL related problems may be displayed if internally subprograms, like the meshers, are called. You can call the subprograms as stand-alone programs to check on this. Or you just proceed with the hints given below if you suspect issues related to DLLs.

The Microsoft C++ Redistributables install into the system directories.

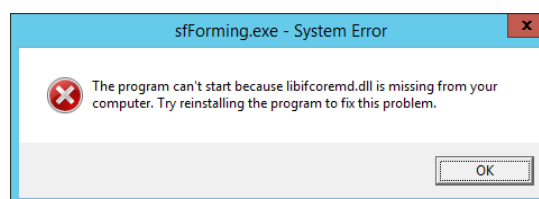


Figure 3.1. Error due to not found Fortran Redistributables

Figure 3.1 shows a typical error message occurring when **Simufact Additive** is started but the required Intel Fortran Redistributables can not be found. Frequently this is not because the installation failed, but due to configuration issues with the search PATH for programs and libraries. The Intel Fortran Redistributables install in a special directory and are referenced by the environment variable PATH. PATH is defined using the environment variable INTEL_DEV_REDIST set by the installer. The directory with the redistributables is not included literally in the definition of PATH. If the expansion of the nested variables fails, the Fortran Redistributables will not be found. You can test this if you execute `set` on the command prompt: INTEL_DEV_REDIST must not be contained in PATH literally, but must be expanded to its value. Some times a reboot helps to archive this. If not, check the definition of the environment variables in the system control. It may help if both, PATH and INTEL_DEV_REDIST, are either user or system variables. If this does not help, redefine PATH in the system control by replacing %INTEL_DEV_REDIST % with its value. Reinstalling the redistributables or even **Simufact Additive**, will *not* help. See Appendix C for the review and configuration of environment variables.

The directory `3rd-party` of the **Simufact Additive** download contains the needed redistributables. If needed, you can reinstall them from there directly without re-installing **Simufact Additive**.

Some times 3rd-party applications install their DLLs in the system directories, but this DLLs conflict with DLLs **Simufact Additive** includes within its installation. As the system directories are checked first, errors can occur. In this case delete the DLLs from the system directories and place them in the installation of the 3rd-party application. One example for this is the `tbb.dll`.

3.1.4. Windows Remote Desktop and 3DConnexion SpaceDevices

The Windows Remote Desktop does *not* support the usage of SpaceDevices from 3DConnexion™. This is a limitation of the Windows Remote Desktop, not of the **Simufact** application. But some more advanced 3rd-party Remote Desktop systems support the the usage of SpaceDevices from 3DConnexion™.

3.1.5. The GUI does not appear after launching the program

- Make sure that the license feature to open the GUI on your machine is available and valid. You should get an error message with the possibility to review the licenses if no licenses are available for the GUI.



During start-up **Simufact Additive** will query all license servers specified in the environment variable `MSC_LICENSE_FILE` for all potentially existing license features to be able to configure it self based on the available licenses. If one of the license server is not available or responds very slow, it may take very long before the GUI either opens based on the found licenses or displays an error message about missing licenses. If you suspect this issue, check the connection to all your license servers or reduce `MSC_LICENSE_FILE` to include only one license server known for fast response. The solver is not affected by this as it checks only for the licenses actually needed and stops when getting the first license.

If it takes more than 5 seconds for a license server to respond, an informative warning message is displayed.

- Check the hints given in Section 3.1.1.2 above.
- Make sure that the target "`...\simufact\additive\2022.1\bin\simufact additive.exe`" of the shortcut to launch the program is correct.
- Open a command prompt, navigate to the installation directory and start `...\simufact\additive\2022.1\bin\simufact additive.exe` from within the command prompt. Watch for errors.
- Check in the Windows™ Taskmanager whether `simufact additive.exe` is running. Note: The Windows™ pop-up messages indicating an application crash may be disabled on your system.

- **Simufact Forming only:** Check the Windows™ Registry for Computer\HKEY_CURRENT_USER\SOFTWARE\Simufact User\Simufact Forming 2022.1\Settings and delete the key WindowPos while all GUIs are closed.

3.1.6. CAD import fails

- Check the installation path and also the path and name of the CAD file. They should not contain special characters such as @, ü, ø, etc. Try to import different files, use some shipped with the Demos & Examples of Simufact Additive.
- CAD import is done using a temporary directory in %TEMP%\simufact.additive. This should be on a local drive with a drive letter. Some times the user names, that get part or %TEMP%, can cause problems. Consider to define an other location for %TEMP% by redefining this environment variable in the Windows™ system control under system.
- Check and redefine the setting ScratchDirectory in the [Environment] section of C:\Users\\AppData\Roaming\Simufact\simufact.additive_2022.1\simufact.additive.ini.
- Execute the redistributable packages setup in the folders ... \3rd-party and ... \3rd-party\Microsoft Visual C++ ... of the Simufact Additive download to re-install the needed redistributables.
- Check whether the operating system supports the creation of 8.3 file names. If not, change the settings of your operating system. For Windows™ see <http://support.microsoft.com/kb/121007/en-us>.

3.1.7. Initial meshing does not work

- Check the installation path and also the name of the geometry. They should not contain special characters such as @, ü, ø, etc. Try to mesh different geometries using different meshers, use some examples of the Demos & Examples of Simufact Additive.
- Initial meshing is done using a temporary directory in %TEMP%\simufact.additive. This should be on a local drive with a drive letter. Some times the user names, that get part or %TEMP% can cause problems. Consider to define an other location for %TEMP% by redefining this environment variable in the Windows™ system control under system.
- Check and redefine the setting ScratchDirectory in the [Environment] section of C:\Users\\AppData\Roaming\Simufact\simufact.additive_2022.1\simufact.additive.ini.

3.1.8. Simulation fails when the project is stored on a network drive

Simulation jobs with DDM may fail to run and exit immediately when the Simufact Additive project is stored on a network drive. Parallel calculations using DDM require a local working directory for the solver. The usage of network drives or network paths is frequently not possible. Thus, to be able to use DDM, the Simufact Additive project needs to be stored on a local hard drive (if neither Simufact Remote nor individual start scripts are used). Otherwise conflicts with the access rights to files will arise because of the control of the solvers by the MPI service causing an immediate abortion of the simulation. If your IT infrastructure or your IT policies do not allow this, you can configure the "local server" of Simufact Remote with a local working directory on your local PC and start Simufact Additive simulations stored on a network drive using "pseudo remote" to your own PC.

3.1.9. Simulation using DDM fail to start

Simulations using the Domain Decomposition Method (DDM) for parallel computing may fail to start writing only a short stub of a log-file. This may be because of too restrictive firewall settings. DDM uses the Intel™ Process manager for MPI applications service to couple the domains during the simulation. This service uses network

protocols and is started and ended with the simulation. If the communication of this service is blocked by your firewall the simulations will abort or may hang at start-up. Change your firewall configuration to solve this. The user may have to accept a firewall warning when the first simulation is started. The name of the service executable is `mpiexec.exe`. Even with network protocols used, all communication is done only on the machine used for the simulation.

3.1.10. Windows error dialog "Program has stopped working"

If an application aborts unexpectedly resp. crashes Microsoft Windows™ typically displays an error dialog indicating this. While this is useful in most cases because it shows that the application was not closed by the user unintentionally or reminds that the application was running on an unwatched computer (for example over night) and generally indicates a major issue, the error dialog can prevent fall back mechanisms to work correctly. For example the meshers of Simufact will retry a failed remeshing during the analysis with settings a couple of times before giving up. If the mesher crashes during one of this trails, the "Program has stopped working" dialog needs to be confirmed before the next fall back is done. Similar issues may occur during automated optimization runs or in other batch operations. Thus you may want to turn off or on this Windows error dialog as needed.

To switch off the Windows error dialog "Program has stopped working" you need to change in the Windows Registry in the key `HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting` the setting `DontShowUI` from 0 to 1. You may have to create this setting first.

3.2. Troubleshooting on Linux™

Please check the troubleshooting hints for Windows™, too. Some typically problems are independent from the operating system. Especially regarding Section 3.1.2. For sure the location of the INI files mention is different, compare Section 3.2.1.

3.2.1. Wrong information in INI files

INI files are created for Linux™ in `~/ .config/Simufact` and `/etc/xdg/Simufact`. They contain paths and other information. In case of problems with these settings, please check the contents of these INI files and - if necessary - edit them manually. If your setting files have been seriously corrupted for some reason, you can delete them or rename them to keep copies.

3.2.2. Jobs fail to start

To check the functionality of the Simufact Additive solvers, a step-by-step debugging should be done as indicated below.

- a. Login the SERVER as the mentioned users and get a shell access resp. a terminal window. Create a temporary directory for testing.
- b. Try to run an FE job manually, start with the minimum options and without parallelisation before increasing the complexity:
 - copy the files `am_testjob*` from `.../simufact/additive/2022.1/sfMarc/tools/test` into the temporary directory, change into that directory.
 - execute the solver start script like this:

```
.../simufact/additive/2022.1/sfMarc/tools/run_marc -j am_testjob -v n -b n -iam sim
```

This will start the testjob in the foreground displaying the content of the log file in the shell. The testjob should finish after some minutes, but you can use `Ctrl+C` to abort the job at any time (which you should not do before the first remeshing was done to check whether remeshing is working, too). If the job does not run, check the output for errors: License? ...

- If you have parallelisation licenses, you can now start the testjob using

```
.../simufact/additive/2022.1/sfMarc/tools/run_marc -j am_testjob -ddm 1 -
nps 2 -v n -b n -iam sim
```

to check whether DDM is working. A typical error would be `ssh` not being well configured, compare Section 1.2.5. It is recommended to delete all files in the temporary directory between the tests and to copy the input files again for the next test.

- If you have a cluster and want to distribute the domains of the parallelisation with DDM to multiple nodes of the cluster, create a file called `hostfile` with the content

```
hostname1 1
hostname2 1
```

for sure, use your host names instead of `hostname1` and `hostname2`. Initially `hostname1` should be the name of your current machine. Now try running the testjob using

```
.../simufact/additive/2022.1/sfMarc/tools/run_marc -j am_testjob -ddm 1 -
nps 2 -host hostfile -v n -b n -iam sim
```

- If you work with user subroutines, you can test the configuration by adding `-u subroutine` to the call of `run_marc`, with `subroutine` being the name of your Fortran file without the extension. A small test user subroutine is provided in `.../simufact/additive/2022.1/sfMarc/tools/test` named `compile_test.f`. It works with all Simufact simulations, including the testjob. For testing copy `compile_test.f` in the temporary directory and add `-u compile_test` to the call of `run_marc`. The test is successful if the simulation immediately stops showing an exit 3020 and descriptive messages in the log and out file. If the simulation aborts with any other message or continues to run after the 1st increment, the compilation of the routine was not successful and needs to be debugged. For details please see the comments in `compile_test.f`.

- c. If you use a scheduling engine, repeat b. with using scheduling. See Section 1.2.5.3 and use an adapted version of the script `schedule_marc.sh` provided in `.../simufact/additive/2022.1/sfTools/sfScheduling/testscripts`.

If all tests to run simulations manually locally have been successful, you can proceed with testing **Start Scripts** from a CLIENT. Again, increase the complexity step-by-step.

3.2.3. DDM jobs do not start

DDM jobs may hang immediately with the following output in the `log`-file:

```
[mpiexec@server] HYDT_dmxu_poll_wait_for_event ...
[mpiexec@server] HYDT_bscu_wait_for_completion ...
[mpiexec@server] HYDT_bsci_wait_for_completion ...
[mpiexec@server] HYD_pmci_wait_for_completion ...
[mpiexec@server] main (/ui/mpich/mpiexec.c:287): process manager error ...
```

In that case HYDRA MPI cannot connect to the single domains. To solve this problem, you might disable `Strict Host Key Verification` for the executing user in his `ssh` configuration. Talk to your IT department how this is handled in your special case. The background of this is that for DDM jobs to run `ssh` must be set up such that it does not prompt for a password or asks for any confirmations (like "unknown host") if the local host and - if distributed domains are used - all involved hosts are accessed. Even if this may work when logged in interactively, the different environment of jobs started by the remote service, which internally uses `sudo`, may lead to a different behaviour. You may want to try to get all possible host names to be known before you disable `Strict Host Key Verification`.

As **unsupported** alternative you may place a comment at the start of the line `INTELMPI_VERSION=HYDRA` in `.../simufact/additive/2022.1/sfMarc/tools/include_linux64`

3.3. Log files

The software writes several log files that provide additional information that might be helpful in case of problems:

- **Solver:** The solver write extensive information in the `log-` and `out-`files in the `_Run_` directory of each simulation. These files contain logging information of the **meshers**, too. They are the first place to check about not starting or aborting simulations. They can be used to judge the simulation progress, too.
- **Simufact Additive GUI:** Keeps track of what the user does and what the program does internally.
- **License client:** keeps track of granted and denied licenses.
- **Result manager:** Keeps track of the result import from the `_Run_` directory into the result repository (directory `_Results_`).
- **Simufact Additive memory dump:** If the GUI or an auxiliary program crashes, this file contains detailed information that might help our programmers to locate and fix the causes that led to the crash.

Generally the log files will clean up them self. Only a limited number or a limited size is stored. Normally no user interaction is needed to delete old log files.

3.3.1. Simufact Additive GUI

Path: `%localappdata%/Simufact/log`

Contents: Keeps track of what the user does and what the program does internally.

```

USE 1026 13:05:55.530463 1292 General      Check license
USE 1026 13:05:55.746518 1292 General      Open main window
USE 1026 13:05:58.052950 1292 General      Create new project
DET 1026 13:05:58.055965 1292 General      Units: mm, ton, C, s, deg., kN, MPa, J, kg/m3, rotation/min, mm/s, J, mm3
USE 1026 13:05:58.071084 1292 General      Show object catalog
USE 1026 13:05:58.072685 1292 General      Insert new process
USE 1026 13:06:01.504554 1292 Settings     Created process type in application module 'Cold forming' and with proces
WAR 1026 13:06:02.233372 1292 General      QFileInfo::absolutePath: Constructed with empty filename
USE 1026 13:06:02.238461 1292 General      Create new process preview for 'FormingFe3D'
DET 1026 13:06:02.239461 1292 General      Create frame 0x9bc0ef0
DET 1026 13:06:02.248467 1292 General      Create view 0x9b24840
USE 1026 13:06:02.258164 1292 General      Created new Qt view wrapper (FormingFe3D - Model view): (ID 4096)
WAR 1026 13:06:02.258464 1292 General      QLayout: Attempting to add QLayout "" to QGroupBox "", which already has
WAR 1026 13:06:02.261167 1292 General      QLayout: Attempting to add QLayout "" to QWidget::AnimationToolBar "animationT
PRO 1026 13:06:13.201246 1292 General      Create autohshape cube (xyz = 0.002 / 0.003 / 0.005, refine = yes, minimal
PRO 1026 13:06:13.201246 1292 General      Insert new item into object catalog: Cube
PRO 1026 13:06:15.298223 1292 General      Assign item in tree view: item Cube, parent Workpiece
PRO 1026 13:06:15.298724 1292 General      Process tree: rename Workpiece as Cube
USE 1026 13:06:54.586088 1292 General      Open the <Translate> dialog
DET 1026 13:07:02.835882 1292 General      New transformation matrix of Cube:
DET 1026 13:07:02.835882 1292 General      1          0          0          0
DET 1026 13:07:02.835882 1292 General      0          1          0          0
DET 1026 13:07:02.835882 1292 General      0          0          0          1
DET 1026 13:07:02.835882 1292 General      0.0023    0          0          1

```

Figure 3.2. Simufact Additive GUI log file

Format:

- message type (USE = user, WAR = warning, PRO = progress, DET = details)
- date
- time (with microseconds)
- thread ID
- topic
- message

3.3.2. License client

The messages of the license client get part of the messages in the `log-` and `out-`files of the solver and of the meshers. The GUI does not log this.

Note: The MSC Licensing license server has its own log-file `lmgrd.log` showing useful information and among this the granted licenses.

3.3.3. Result manager

Path: `<process directory>/_Log/<process name>_ResultManager.log`

Contents: Keeps track of the result import from the `_Run_` directory into the result repository (directory `_Results_`).

```
Okt 26 16 12:02:42.578 [7 DEBUG] Solver @ increment 27, state: Running
Okt 26 16 12:02:42.579 [7 INFO ] Importing increment 25
Okt 26 16 12:02:42.581 [7 DEBUG] Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16 12:02:42.581 [7 DEBUG] Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16 12:02:42.582 [7 DEBUG] Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16 12:02:42.583 [7 DEBUG] Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16 12:02:42.584 [7 INFO ] Adding surface set to C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_C
Okt 26 16 12:02:42.589 [7 DEBUG] Took 5ms.
Okt 26 16 12:02:42.589 [7 DEBUG] Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16 12:02:42.590 [7 DEBUG] Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16 12:02:42.591 [7 DEBUG] Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16 12:02:42.592 [7 DEBUG] Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16 12:02:42.595 [- DEBUG] Merge new increments
Okt 26 16 12:02:42.595 [- DEBUG] Fetched increment 25
Okt 26 16 12:02:43.110 [7 DEBUG] Solver @ increment 38, state: Running
```

Figure 3.3. Result manager log file

3.3.4. Simufact Additive memory dumps

Path: `%LOCALAPPDATA%\CrashDumps` (the standard Windows™ location), may differ if old versions have been installed before.

Contents: If the GUI or an auxiliary program crashes, this file contains detailed information provided by the operating system (error message, call stack etc.) that might help our programmers to locate and fix the causes that led to the crash. You may be asked by our support for this file. The memory dump files contain no personal or model specific information and indicate only the internal routine that crashed. Additional information about what was done when the crash happened is needed. A maximum of 10 files is kept.

Memory dumps are only created if the Windows Registry keys for the respective Simufact executable shown in the figure below exist. They are created during the installation of the respective Simufact application, but may become deleted by Windows Updates or other means. You may have to recreate them manually.

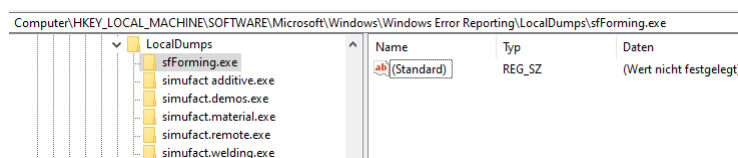


Figure 3.4. Window Registry keys needed for memory dumps

3.4. Support

Although the simple and intuitive use of our software is one of our main focuses, the design of some complex processes requires additional training courses. Troubleshooting can also become difficult and time-consuming for unexperienced users. Generally it makes a lot of sense that the user solves these problems by himself in order to gain experience, but often a tight schedule does not allow this. For this reason Simufact offers an extensive support to our customers.

Please note that only the current version and the previous main version of the product in question can be supported.

Please direct your support requests always to your local Simufact or MSC representative where you bought the software. For our global sales & support network please see: <https://www.simufact.com/global-sales-support.html>

In the help section of the Simufact Additive menu you can find an Infosheet that describes how to contact the Simufact support and what information needs to be provided to enable an efficient solution of your problem.

Appendix A. MSC Licensing: Configuring and managing licenses



MSC Licensing, the FLEXlm based licensing software used for Simufact and MSC products, is shipped with its own documentation. A lot of additional information is available, too. This appendix is not intended to be a comprehensive information about MSC Licensing, only the most common tasks and situations when configuring and managing licenses are described and some more in depth topics are shortly mentioned. In case of deviations between this documentation and the MSC Licensing resp. FLEXlm documentation the later shall prevail.



The license system consists of a client, this is the Simufact application you want to run, and a server, that is the software service that provides the licences based on a license file. Additionally there are graphical and command line utilities for configuration and status information purposes.



If you are using Simufact Remote (Simufact Forming only) you need to to stop and start again the Simufact Remote service ("Local server") after you have changed the license configuration or after you have updated the license file (do not use the "restart" option of the operating system for this, use stop and then start). Otherwise the license changes may not be recognised by Simufact Remote, which may result in unexpected errors.

A.1. Configuring the license client

The license client is configured only using the environment variable `MSC_LICENSE_FILE`. This variable tells the Simufact application where to look for the license. `MSC_LICENSE_FILE` is typically something like `'portnumber@hostname'` pointing to a license server, which applies to nodelocked licenses, too. Only in rare special cases, which are explicitly instructed in the delivery letter of the license file, `MSC_LICENSE_FILE` directly points to the license file. `MSC_LICENSE_FILE` may be a list with multiple entries separated by ";" (Windows™) or ":" (Linux™). These entries are queried in the order given.

Please note that `MSC_LICENSE_FILE` may be a system wide environment variable - recommended in most cases - or a user specific environment variable. For the configuration of environment variables please see Appendix C.

The value of `MSC_LICENSE_FILE` is displayed in the settings dialog of the Simufact application under Licenses and in all cases of a license failure. Further more it can be reviewed using the means provided by our operation system. Frequently using the command line is simple and fast for this: `set` shows a list of all environment variables, `echo %MSC_LICENSE_FILE%` (Windows™) resp. `echo $MSC_LICENSE_FILE` (Linux™) display just this variable.

`MSC_LICENSE_FILE` can be set during the installation (Windows™) or configuration (Linux™, only Simufact Forming) of your Simufact application, but changing it afterwards is not supported within the Simufact applications or within MSC Licensing. Use the means provided by our operation system for this, see Appendix C.

A.2. Firewall considerations

The license server of MSC Licensing uses two services: The FLEXlm main service called "lmgrd" and the vendor service called "MSC". Both use different ports. Thus if you have a firewall between license server and client, you need to open both ports. The port used by the main service is configured during the installation of MSC Licensing and is specified on the clients within the environment vari-

able `MSC_LICENSE_FILE`. As default the port used by the vendor service is randomly chosen during startup. But it can be configured in the license file, which enables a constant firewall configuration.

To configure the port to be used by the vendor service, add `port=your_number` to the end of the `DAEMON` line in the top of the license file (see example blow) and restart the license server service either using the buttons provided in the graphically `MSC Licensing Utility (LMTOOLS)` on the **Start/Stop/Reread** tab or using the means provided in the service control of your operating system.

The port to be used by the main service is given as last argument in the `SERVER` line in the top of the license file, thus it can be reviewed and changed there. Please see Appendix B for how to open ports on a firewall.



Resulting from this and other configurations there are differences between the license file as it is shipped to you and as it is used by the license server.

```
SERVER ING08 1813731e3d93 27500
DAEMON MSC "C:\Program Files\MSC.Software\MSC Licensing\Helium
\msc.exe" port=27501
#
# MSC License Reference ID: WXYZ
#
# Simufact
#
FEATURE SF_FORM_GUI MSC 2018.0309 09-mar-2018 2 F92DAC800190 \
VENDOR_STRING=PID:10653 ISSUED=11-mar-2017 ck=103 \
```

A.3. Updating the license file

After a renewal of your maintenance or lease period and on other occasions you will receive an updated license file. Please note that you will typically not receive a new license file once a new release of your Simufact applications has been published. To update the license file used by your license server, follow the following steps:

1. Locate the used license file in the installation directory of `MSC Licensing` - typically `license.dat` in `C:\Program Files\MSC.Software\MSC Licensing\Helium` - and create a backup copy of it.
2. Stop the license server service, either using the button provided in the graphically `MSC.Licensing Utility (LMTOOLS)` on the **Start/Stop/Reread** tab or using the means provided in the service control of your operating system.
3. Copy the received new license file `license.dat` into the installation directory of `MSC Licensing` replacing the old already existing file.
4. Manually edit the new `license.dat` as shown below. The automatic configurations done in the license file during the installation of `MSC Licensing` have to be reproduced, potential manual configurations in the license file (for example for a firewall, see above) have to be reproduced, too.
5. Restart the license server service, either using the button provided in the graphically `MSC.Licensing Utility (LMTOOLS)` on the **Start/Stop/Reread** tab or using the means provided in the service control of your operating system.

The top of a license file as it is shipped typically looks like:

```
SERVER this_host 1813731e3d93 27500
DAEMON MSC /your_path/msc
#
# MSC License Reference ID: WXYZ
#
```

```
# Simufact
#
FEATURE SF_FORM_GUI MSC 2018.0309 09-mar-2018 2 F92DAC800190 \
  VENDOR_STRING=PID:10653 ISSUED=11-mar-2017 ck=103 \
```

If for example your license server machine has the IP-name ING08, you have chosen to use port 1700 for the main license service, you have installed in the default location and you have - which is optionally - manually set a port for the vendor service, you have to edit the new license file to look like this:

```
SERVER ING08 1813731e3d93 1700
DAEMON MSC "C:\Program Files\MSC.Software\MSC Licensing\Helium
\msc.exe" port=1701
#
# MSC License Reference ID: WXYZ
#
# Simufact
#
FEATURE SF_FORM_GUI MSC 2018.0309 09-mar-2018 2 F92DAC800190 \
  VENDOR_STRING=PID:10653 ISSUED=11-mar-2017 ck=103 \
```

Please note that the above modifications are an example which has to be adapted to your local settings. **It is recommend to review the old, working license file to check which adaptations are needed in the new license file.**

A.4. Releasing Licenses

If the Simufact application or the Simufact solver ends, the license is released and can be used for the next simulation or GUI. This works pretty reliable, even if the application or solver has no clean exit resp. crashes. However in rare circumstances, especially when the operating system crashes or if the network connection is changed back and forth between LAN and WLAN or if VPN is (dis-)connected, licenses may not be released correctly. Because of this it can be useful to configure the license server in a way that licenses are released after a certain timeout if the connection to the license client is lost.

For this create a file `msc.opt` and place it in the installation directory of the license server where the license file `license.dat` is located, too. The content of `msc.opt` is for example:

```
TIMEOUTALL 1000
```

1000 is the configurable time in seconds after which the license server will release licenses when there is no contact with the license client any more. Timeouts below 900 seconds will be ignored, thus the license will never be released. Additionally this option file needs to be referenced in the DAEMON line in the license file:

```
SERVER ING08 1813731e3d93 27500
DAEMON MSC \
  "C:\Program Files\MSC.Software\MSC Licensing\Helium\msc.exe" \
  "C:\Program Files\MSC.Software\MSC Licensing\Helium\msc.opt" \
  port=27501
#
# MSC License Reference ID: WXYZ
#
# Simufact
```

Note that the DAEMON line is one logical line with the \ as line wrap character to keep the text line together. `port=...` is optionally, see Section A.2. **Restart the MSC Licensing service after these changes to activate the usage of the option file.**

To release individual licenses the command line utility `lmutil.exe` shipped and installed with MSC Licensing can be used. This should work on the license server and if the access to `lmremove` and

lmdown has not be restricted to the host or user running the license server (there is an option for this in the graphically MSC Licensing Utility (LMTOOLS)) from every host for every user that has access to lmutil.exe. The syntax to release an individual license is:

```
lmutil lmremove -c port@License_server_name LicenseFeature User  
ClientMachine ClientDisplay
```

The required information for this can be gathered using:

```
lmutil lmstat -a -c port@License_server_name -f LicenseFeature
```

The output will generally be User, ClientMachine, ClientDisplay, version, For the LicenseFeature names see the license file license.dat. For their current usage see the license overview in the settings of the Simufact application. For more options of lmutil.exe see Section A.8.

A restart of the license server service will release all licenses. The Simufact application and the Simufact solver normally recover from this when they can contact the license server again and can re-checkout the required licenses.

A.5. Combination with other MSC products

If you have licensed Simufact and other MSC products on the same license server, MSC Licensing is needed only once on this machine, actually it must only be used once on this machine. But you will receive separate license files for Simufact and other MSC products at separate times and from separate senders. You have to combine these license files for the usage by the license server.

The recommended method to do so is:

1. Install and configure MSC Licensing either for the Simufact application or for the MSC application in the normal way using only the corresponding license file. Test your installation.
2. Stop the license server service, either using the button provided in the graphically MSC.Licensing Utility (LMTOOLS) on the **Start/Stop/Reread** tab or using the means provided in the service control of your operating system.
3. Open the license file in the installation directory of MSC Licensing - typically license.dat in C:\Program Files\MSC.Software\MSC Licensing\Helium - with a text editor and copy all FEATURE lines of the 2nd license file into it. Additionally you may add some comment lines starting with # to separate the licenses for the different products. Please note that one FEATURE line typically consists of several lines of text. The \ at the end of a line of text indicate that the FEATURE line continuous in the next line of text. Please take care to copy all FEATURE lines complete and unchanged. Do not copy the SERVER and the DAEMON line of the 2nd license file.
4. Restart the license server service, either using the button provided in the graphically MSC.Licensing Utility (LMTOOLS) on the **Start/Stop/Reread** tab or using the means provided in the service control of your operating system.

Once you receive an updated license file for one of your Simufact or MSC products, repeat the steps above, but do not add FEATURE lines during copying but replace the existing FEATURE lines with the updates lines. Frequently 1st deleting all FEATURE lines of a product in the license file in the installation directory and then copying the new lines will be a good approach.

As an alternative advanced users may configure and maintain separate license files for different products. See the tech article KB8021145 in the MSC SimCompanion Knowledge Portal for details.

A.6. Combination with 3rd party products

MSC Licensing is based on the FLEXlm license management software of Flexera Software (www.flexerasoftware.com). FLEXlm is widely used of a lot of applications of different software

vendors. Typically no special care is needed when running applications of different vendors that use the FLEXlm license management system. Conflicts are rare. Nevertheless here some hints about this.



Try to avoid the usage of the general FLEXlm environment variable `LM_LICENSE_FILE`. In rare conditions of license server and license client machines cross referencing each other for different applications this may result in conflicts including non-functionality. Use the vendor specific environment variables like `MSC_LICENSE_FILE`, `HPQ_LICENSE_FILE` or similar instead. If conflicts can not be solved with this, encapsulate the concerned license server and / or applications into their own environment.

If the same machine is used as license server for both, Simufact and MSC applications as well as for a 3rd party application that uses a FLEXlm based licensing system, we recommend to keep the installations separate and to run each vendors FLEXlm license server as a separate process on the same machine. The resources needed for this are very little. Keeping the installations separate simplifies the maintenance of the licenses and avoids conflicts, for example resulting from different required FLEXlm versions.

However, if desired, a common "lmgrd" main license service can be used. In this case the license files of all vendors would need to be combined including the required `DAEMON` and `VENDOR` lines as well as the other options that may be present. For more information, documentation and help for this option, please see the relevant FLEXlm resources. This is outside of the scope of the support of Simufact and MSC.

A.7. Restrictions

Within one license file each `FEATURE` can only be included once. Within one license file shipped to you all features will be of the same type (nodelocked or networking) and will have the same maintenance end date and the same expiry date. Furthermore a license file shipped to you may not include all your licensed features, but only the latest additions and changes. Thus, in case you purchase or lease additional Simufact products or modules during your maintenance or lease period, you will have to combine the new license file shipped to you and the already running licenses manually similar to the procedure explained in Section A.5 above.

In cases like a short term lease of additional simulation capacity or a test of the effectivity of for example more parallel options where the additional licenses expire before your regular license, you will probably receive a license file with `FEATURE` lines containing the sum of your licensed quantity, but expiring at the earliest date. In this case you first have to backup your used license file. Then you have to delete these features from your used license file and to add the new received `FEATURE` lines with the sum of your licensed quantity. After the expiry of the short term licenses you have to switch back to using your original backed-up license file.

Remember to stop and to start your license server service after each change of your license file.

A.8. Troubleshooting

- If a license can not be granted, Simufact applications frequently do not exit with a license failure, but go into status "Waiting for license", see Section 3.1.2.
- The license server service writes a log file, typically `C:\MSC.Software\MSC Licensing\Helium\LOG\lmgrd.log`. This file contains useful debugging information. Read it carefully if there is a license issue. Besides information about the license server, every client request should be echoed in this file.
- During start-up the GUI will query all license servers specified in the environment variable `MSC_LICENSE_FILE` for all potentially existing license features to be able to configure it self based on the available licenses. If one of the license server is not available or responds very slow, it may take very long before the GUI either opens based on the found licenses or displays an error

message about missing licenses. If you suspect this issue, check the connection to all your license servers or reduce `MSC_LICENSE_FILE` to include only one license server known for fast response, exclude license servers that serve only other MSC products. The solver is not affected by this as it checks only for the licenses actually needed and stops when getting the first license.

If it takes more than 5 seconds for a license server to respond, an informative warning message is displayed. This may be because a license server is unavailable, invalid or responding very slow.

- Take care that the value of `MSC_LICENSE_FILE` does not contain any spaces, including spaces at the end of the license server name.
- MSC Licensing is shipped with the powerful command line debugging tool `lmutil`. Check `lmutil -help` for the available options. Some typical tasks are:

`lmutil lmhostid` to get a list of valid host-IDs of the machine used. In rare circumstances this may deviate from the output of `getmac /v` or `ipconfig /all` (both on Windows™) that are typically used when determining the host-ID before applying for a license.

`lmutil lmcksum -c license.dat` to check the syntax and the checksum of the license file, which is useful if issues due to the transmission or the editing of the license file are suspected.

`lmutil lmstat -a -c portnumber@hostname` to check the status of the license server and the vendor services including a list of license features that can be served. As the output may be bigger, you may want to redirect it to a file by adding `> status.txt` at the end of the command line.

`lmutil lmdiag -c portnumber@hostname -n > diag.txt` to review diagnostic information about all license features served including whether they can be checked out and reasons why they can not be checked out. As the output of this command is typically quite large, it is recommended to redirect it to a file.

- Some times network license servers respond too slow to license requests causing license failures without obvious reasons. Typical examples are large networks, wide area networks, VPN configurations and some times virtual machines, even in small networks. “Too slow” is some thing in the internal communication, the license failure appears (for the GUI) nearly quite as fast as a denied license, much faster than it appears if the network connection is gone. These issues got more relevant starting with Simufact Welding 8.0, Simufact Additive 4.0 and Simufact Forming 16.0. The same license server may work for one client, but not for another. The typical error message is: `FlexNet Licensing error:-96, ...` Setting the environment variable `FLEXLM_TIMEOUT` on the license client to for example 5000000 can help in these cases. The value may have to be increased further. It is in microseconds. For the configuration of environment variables see Appendix C.

A.9. Miscellaneous

- The license server service writes a log file, typically `C:\MSC.Software\MSC Licensing\Helium\LOG\lmgrd.log`. This file shows `OUT:` and `IN:` messages of the license usage. This can be used to monitor the license usage. For more advanced usage reporting Flexera Software offers extra-cost tools directly to end-users. This is outside of the scope of the support of Simufact and MSC.
- Network licenses can be served by redundant license servers where one out of three may be offline. A special license file is needed for this. If needed, please ask your local Simufact sells representative about this. Please note: While new license queries will be very stable in this configuration, long running license occupancies (like solver runs) may still show some instability when a license server goes offline. See the tech article KB8020361 in the MSC SimCompanion Knowledge Portal for details how to configure this set-up.
- MSC Licensing offers a lot of advanced configurations. For example the access to network licenses can be restricted and controlled flexibly including reserving features for certain users. This is done

using an additional option file. Please see the relevant FLEXlm resources. This is outside of the scope of the support of Simufact and MSC.

Appendix B. Firewall: New rules for opening ports

Here you can read how to change the firewall settings under Windows 10™ to allow messages through the network to the port 9987 for the *Transmission Control Protocol (TCP)*. In an environment with firewalls you need to open the ports needed for MSC Licensing (see Section A.2), the Simufact Monitor uses port 9985, Simufact Remote uses ports 9987, 9988 and 9990.

1. Open the **Control panel** and then **Windows Defender Firewall with Advanced Security**. The dialog displayed in Figure B.1 will open.

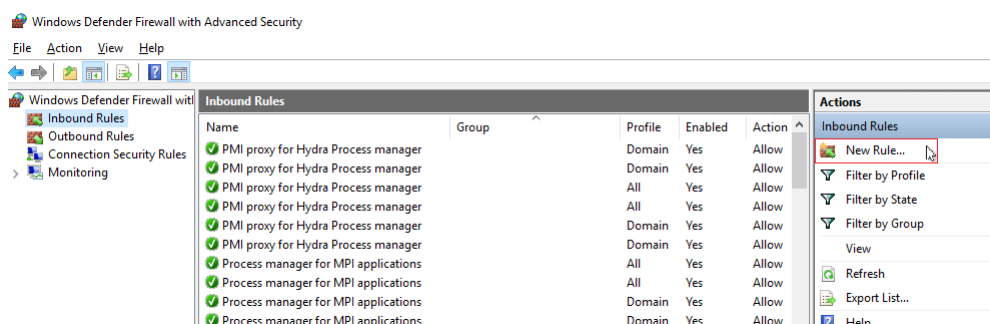


Figure B.1. New firewall rules for incoming messages

2. Select **Inbound Rules** and then **New Rule...** A new dialog appears (Figure B.2).

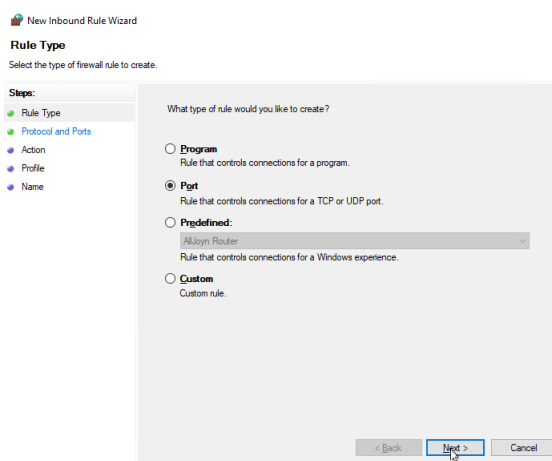


Figure B.2. New firewall rule: rule type

3. Then choose **Port** and click **Next**. The next dialog page Figure B.3 will appear.

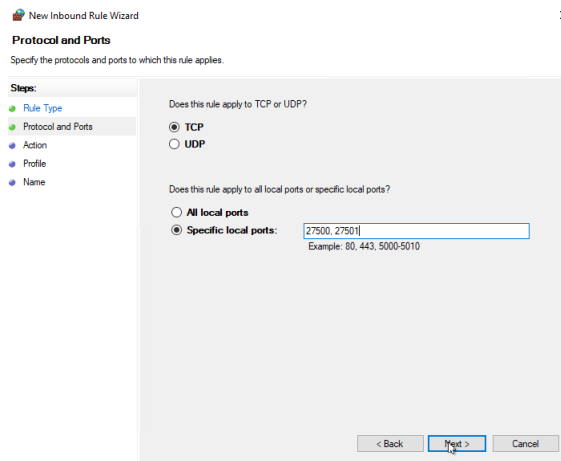


Figure B.3. New firewall rule: protocol and port number

4. Select the protocol **TCP** and enter the desired port numbers (here shown for typical port numbers for MSC Licensing) in the input field **Specific local ports** and click **Next**. The next dialog page Figure B.4 will appear.

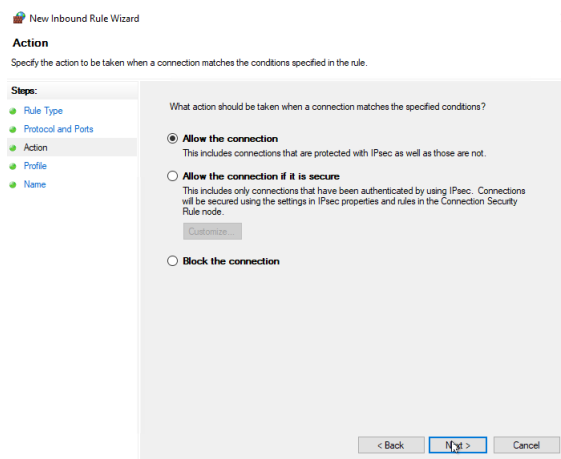


Figure B.4. New firewall rule: action

5. Select **Allow the connection** and **Next**. The next dialog page Figure B.5 will appear.

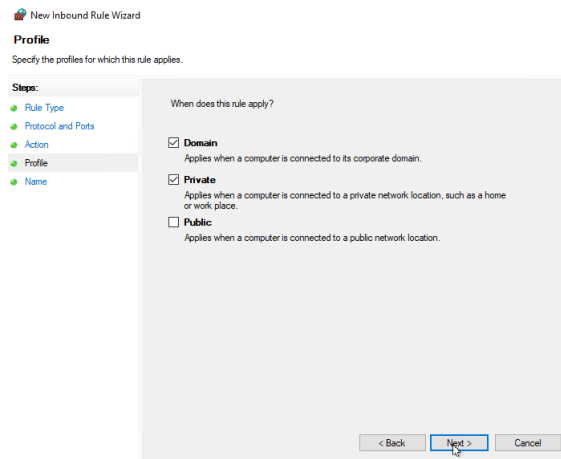


Figure B.5. New firewall rule: profile

- Depending on your environment, select **Domain** or **Private** and click **Next**. The last dialog page Figure B.6 will appear.

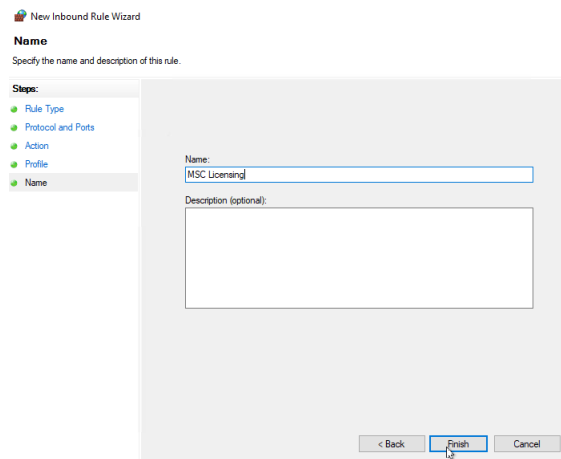


Figure B.6. New firewall rule: name

- Enter a name for the rule (e.g. *MSC Licensing*) and click **Finish**.

Appendix C. Maintaining Environment Variables

Here you can read how to review, set, and change environment variables using the means provided by your operating system. Besides registry settings and ini-files, environment variables are used to configure applications. Environment variables are passed from the calling application to the called applications, thus for example from the operating system to the Simufact Additive GUI and from there to the solver. Or from the command line to the program started from there. Applications can only read environment variables, but never set or change change them to become effective immediately. After a change of a environment variable the application always needs to be restarted to get the change active. This is true for services resp. daemons, too.

Environment variables can be configured separately for each user or as system wide variables. The usage of system wide variables is preferred. Environment variables can be nested.



Always restart the Simufact Service (part of Simufact Remote (Simufact Forming, only)) after `MSC_LICENSE_FILE` or any other environment variable was changed! Always restart the Simufact applications after environment variables have been changed! A reboot of the machine is typically not needed, but in doubt, do it.

Besides the usual system environment variables like `PATH`, Simufact uses the following environment variables:

- `SF_LANGUAGE`
- `MSC_LICENSE_FILE`
- `MSC_LICENSE_SF_WAITINGTIME`

For debugging or in special cases more environment variables may be applicable.

C.1. Microsoft Windows™ 10

Starting with Microsoft Windows™ 10 the configuration of environment variables can not be accessed using the Windows menus only. The configuration of environment variables can now be accessed using the following alternative methods:

- In the Windows Search (Windows Key + S), search for `environment variables` and pick the first and most relevant hit. The correct search result leads directly to the **Environment Variables** configuration window shown below. The search may need to be done using your local language.
- Open the run program prompt (Windows Key + R) and start `sysdm.cpl`. This opens the **System Properties** dialogue, where you select the **Advanced** tab and there **Environment Variables ...** to access the **Environment Variables** configuration window shown below.
- For those who still think about Windows 7: Windows Search (Windows Key + S), search for `control panel` and open the **Control Panel**. There select **System** and then **Advanced system settings** to open the **System Properties** dialogue and proceed like before.
- For those who like the command prompt: `set` and `setx` are your friends.

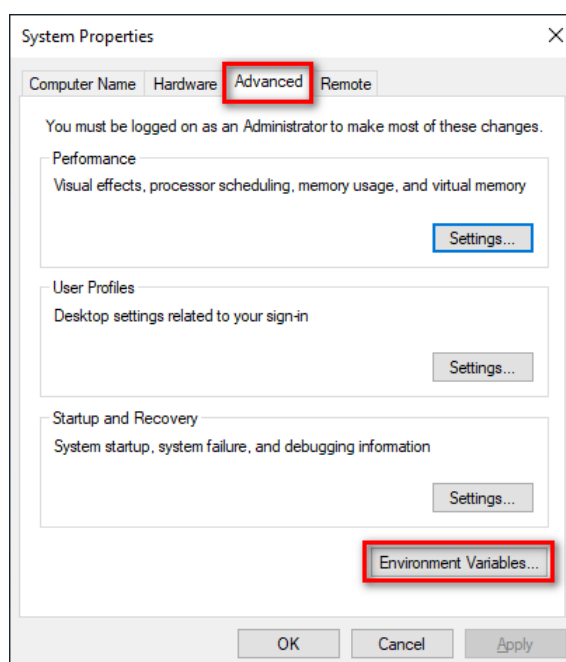


Figure C.1. System Properties

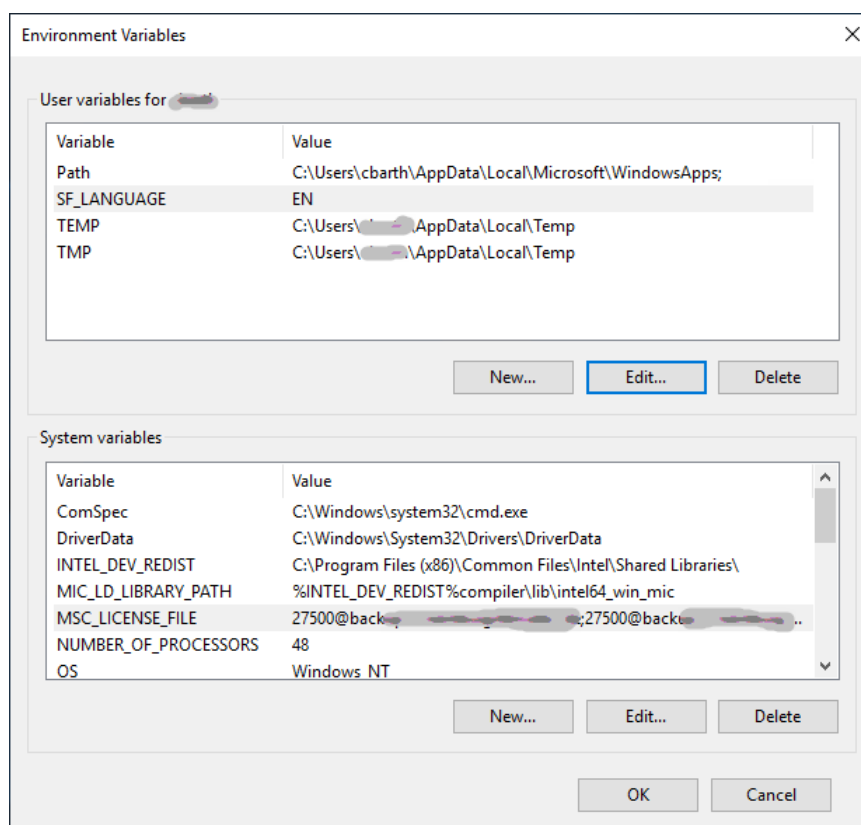


Figure C.2. Configuration of Environment Variables

In the **Environment Variables** configuration window select the variable to be changed and press **Edit ...**. Preferably use only **System variables**. Avoid to define the same variable twice as **User variable** and **System variable**. If needed select **New ...** to define a new environment variable.

To review the effective environment variables, open a command prompt and execute `set`.

C.2. Linux™

Use the means of your shell considering the initialisation files of your (login) shell, see the relevant man pages. For the solver consider to set the needed environment at the beginning of the solver start scripts mentioned in Section 1.2.4. **Simufact Forming** only: Rerun the configuration utility for this.

Appendix D. Password-less login with ssh

In Marc™ 2014.0.0 and later, on Linux™ the default MPI version is Intel(R) MPI. This MPI version requires that the `ssh` command has been set up so that it does not prompt for a password. Here is a description on how to set this up.

- Make sure there is a directory called `.ssh` in the home directory (note that in this home directory only you need write permissions):

```
cd $HOME
```

```
ls .ssh
```

- If it does not exist, create it:

```
mkdir .ssh
```

```
chmod 700 .ssh
```

```
cd .ssh
```

- Execute the command:

```
ssh-keygen -t rsa -f id_rsa -P ''
```

Please note that it is `-P` followed by two single quotes. This will create two files: `id_rsa` and `id_rsa.pub`.

- Copy `id_rsa` to a file called `identity`:

```
cp id_rsa identity
```

- Append `id_rsa.pub` to a file called `authorized_keys`

```
cat id_rsa.pub >> authorized_keys
```

```
chmod 600 authorized_keys
```

The directory `$HOME/.ssh` should now contain the four files `id_rsa`, `id_rsa.pub`, `identity` and `authorized_keys` (and possibly more files). If all went well, it should now be possible to do:

```
ssh localhost
```

without getting a prompt for the password. You can replace `localhost` by the hostname of the current machine.

In order to be able to connect to other Linux™ machines without being prompted for a password (for example for running parallel network jobs), use the following steps:

- Copy the file `id_rsa.pub` that was created above to the other machine. Make sure it has a directory `$HOME/.ssh`. Append the file `id_rsa.pub` to the files `$HOME/.ssh/authorized_keys` and `$HOME/.ssh/authorized_keys2` and give them the appropriate permission:

```
cat id_rsa.pub >> $HOME/.ssh/authorized_keys
```

```
cat id_rsa.pub >> $HOME/.ssh/authorized_keys2
```

```
chmod 600 $HOME/.ssh/authorized_keys $HOME/.ssh/authorized_keys2
```

```
chmod 700 $HOME/.ssh
```

- The first time you log in with `ssh` to the second system you will get a warning and asked if you want to continue. Type `yes` to accept and the remote host will be added to the file `$HOME/.ssh/known_hosts` and the next time you will not be prompted.

Based on your local environment additional configuration may be needed to archive password-less login with `ssh`. For example, **SELinux** with NFS mouted home directories requires a `setsebool -P use_home_dirs 1` to enable root to access the stored keys. Restrictions may apply. If needed, contact your local administrator for help.

Appendix E. Basic commands of the editor vim

`visudo` will probably open `/etc/sudoers` in `vim`, a non-graphical text editor operated by key strokes. For your reference here we offer a list of the very most basic key strokes needed to work with `vim`. See the man page and the online help of `vim` for much more.

arrow keys	navigate the cursor
i	change to the insertion mode before the current character
a	change to the insertion mode after the current character
ESC-key	change back to the navigation mode
ZZ	save and exit
:q	exit with query about saving
:q!	exit without saving
:w	save
Ctrl + f	go down one screen
Ctrl + b	go up one screen
1G	go to the start of the file
G	go to the end of the file
\$	go to the end of the line
0	go to the beginning of the line
ma	set the marking a
'a	go to the marking a
d'a	delete between the cursor and the marking a
x	delete the character below the cursor
dd	delete the current line
/xyz	search for xyz (case-sensitive)
n	repeat last search
N	repeat last search upwards
:set [no]wrap	switch the line break on or off

Key strokes starting with `:` or `/` need to be followed by the enter key.



Hexagon is a global leader in digital reality solutions, combining sensor, software and autonomous technologies. We are putting data to work to boost efficiency, productivity, quality and safety across industrial, manufacturing, infrastructure, public sector, and mobility applications. Our technologies are shaping production and people-related ecosystems to become increasingly connected and autonomous – ensuring a scalable, sustainable future.

Hexagon's Manufacturing Intelligence division provides solutions that use data from design and engineering, production and metrology to make manufacturing smarter. For more information, visit hexagonmi.com.

Learn more about Hexagon (Nasdaq Stockholm: HEXA B) at hexagon.com and follow us [@HexagonAB](https://twitter.com/HexagonAB).