Simufact Forming 2022.1 Installation guide

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

HEXAGON

Trademarks

Patents:

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

Contact:

www.simufact.com/contact

For the installation, configuration and usage of Simufact Remote please see Chapter 2 and for Linux Section 1.2.5.3.

For troubleshooting please check Chapter 4.

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 14
1.2.6. Some notes on configuration and security of the Remote Server 20
1.2.7. Queueing 22
2. Simufact Remote 23
2.1. Installation 24
2.2. Configuration 25
2.2.1. Introduction 25
2.2.2. Server configuration 25
2.2.3. Client configuration 27
2.2.4. Starting the analysis on the server from the client 29
2.3. Queueing 31
3. Installation and configuration of compilers 32
3.1. Intel® Parallel Studio or Intel® oneAPI? 33
3.2. Intel Parallel Studio XE for Fortran 33
3.2.1. Requirements 33
3.2.2. Installation and configuration 34
3.3. Intel® oneAPI Compiler 36
4. Troubleshooting 38
4.1. Troubleshooting on Windows™ 39
4.1.1. General hints 39
4.1.2. Status "Waiting for License" 41
4.1.3. Problems with Shared Libraries (DLLs) 41
4.1.4. Windows Remote Desktop and 3DConnexion SpaceDevices 42
4.1.5. The GUI does not appear after launching the program 42
4.1.6. CAD import fails 43
4.1.7. Initial meshing does not work 43
4.1.8. Simulation fails when the project is stored on a network drive 43
4.1.9. Simulation using DDM fail to start 43
4.1.10. Windows error dialog "Program has stopped working" 44
4.1.11. Remote problems 44
4.2. Troubleshooting on Linux™ 46
4.2.1. Wrong information in INI files 46
4.2.2. The Simufact Remote server doesn't respond anymore 47
4.2.3. The Simufact Remote server does not start 47
4.2.4. The Simufact Remote server does accept simulations 47
4.2.5. Jobs submitted by Simufact Remote fail to start on the SERVER 48
4.2.6. DDM jobs started by the Remote server hang 50
4.3. Log files 51
4.3.1. Simufact Forming GUI 51
4.3.2. License client 52
4.3.3. Result manager 52
4.3.4. Remote service 52
4.3.5. File synchronizer 52
4.3.6. Simufact Forming memory dumps 53

4.4. Support ... 53
A. MSC Licensing: Configuring and managing licenses ... 54
A.1. Configuring the license client ... 54
A.2. Firewall considerations ... 54
A.3. Updating the license file ... 55
A.4. Releasing Licenses ... 56
A.5. Combination with other MSC products ... 57
A.6. Combination with 3rd party products ... 57
A.7. Restrictions ... 58
A.8. Troubleshooting ... 58
A.9. Miscellaneous ... 59
B. Firewall: New rules for opening ports ... 61
C. Maintaining Environment Variables ... 64
C.1. Microsoft Windows™ 10 ... 64
C.2. Linux™ ... 66
D. Password-less login with `ssh` ... 67
E. Basic commands of the editor `vim` ... 69

List of Examples

2.1. Path mapping 29

1 Installation instructions

Before installing Simufact Forming 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 Forming (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 Forming, too. For nodelocked licenses the license server software and Simufact Forming 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 Forming 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 Forming. It is available as separate download in the MSC Solution Download Center (SDC) where you have downloaded Simufact Forming. MSC Licensing is not installed by the Simufact Forming 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 Forming. 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™

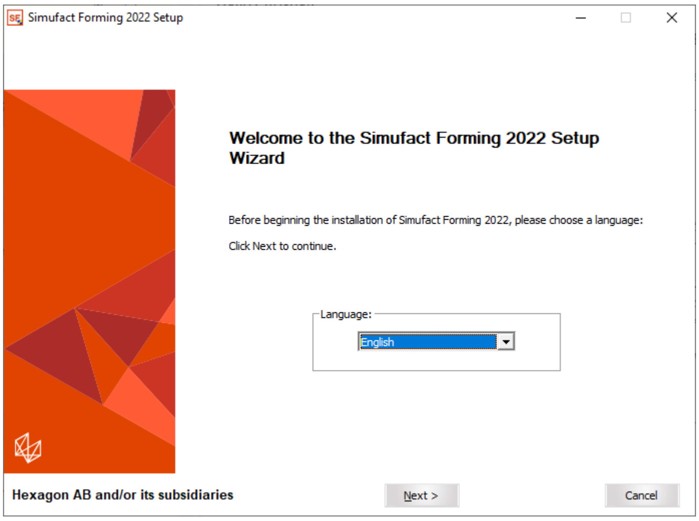
1.1.1. Attended installation

The Simufact Forming 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 frist 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 Forming. The language used in Simufact Forming can be chosen in the settings of Simufact Forming.



Simufact Forming 2022 Setup

Welcome to the Simufact Forming 2022 Setup Wizard

Before beginning the installation of Simufact Forming 2022, please choose a language:

Click Next to continue.

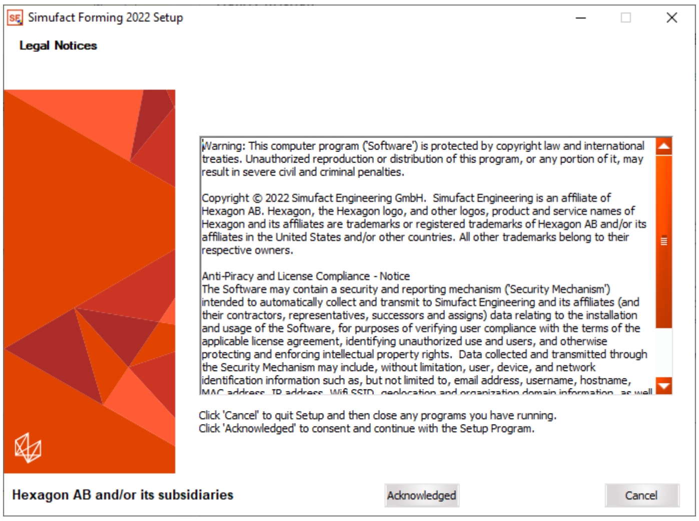
Language: English

Hexagon AB and/or its subsidiaries

Figure 1.1. Language selection

1.1.1.2. License agreement

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.



Simufact Forming 2022 Setup

Legal Notices

Warning: This computer program ('Software') is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties.

Copyright © 2022 Simufact Engineering GmbH. Simufact Engineering is an affiliate of Hexagon AB. Hexagon, the Hexagon logo, and other logos, product and service names of Hexagon and its affiliates are trademarks or registered trademarks of Hexagon AB and/or its affiliates in the United States and/or other countries. All other trademarks belong to their respective owners.

Anti-Piracy and License Compliance - Notice
The Software may contain a security and reporting mechanism ('Security Mechanism') intended to automatically collect and transmit to Simufact Engineering and its affiliates (and their contractors, representatives, successors and assigns) data relating to the installation and usage of the Software, for purposes of verifying user compliance with the terms of the applicable license agreement, identifying unauthorized use and users, and otherwise protecting and enforcing intellectual property rights. Data collected and transmitted through the Security Mechanism may include, without limitation, user, device, and network identification information such as, but not limited to, email address, username, hostname, MAC address, IP address, Wifi SSID, geolocation and organization domain information, as well

Click 'Cancel' to quit Setup and then close any programs you have running.
Click 'Acknowledged' to consent and continue with the Setup Program.

Hexagon AB and/or its subsidiaries

Figure 1.2. License agreement

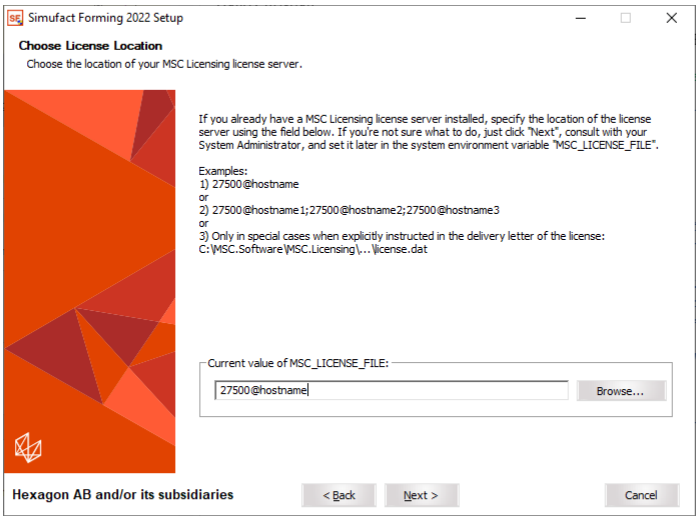
1.1.1.3. Choose license location

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 ; , : Ä or ß, and must be shorter than 248 characters otherwise the license will not be found.

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.



Choose License Location

Choose the location of your MSC Licensing license server.

If you already have a MSC Licensing license server installed, specify the location of the license server using the field below. If you're not sure what to do, just click "Next", consult with your System Administrator, and set it later in the system environment variable "MSC_LICENSE_FILE".

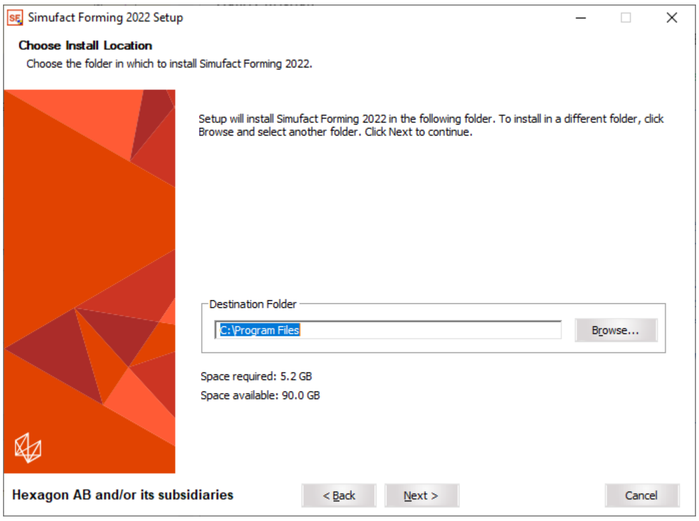
Examples:
1) 27500@hostname
or
2) 27500@hostname1;27500@hostname2;27500@hostname3
or
3) Only in special cases when explicitly instructed in the delivery letter of the license:
C:\MSC.Software\MSC.Licensing\...\license.dat

Current value of MSC_LICENSE_FILE: 27500@hostname

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 Forming 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.



Simufact Forming 2022 Setup

Choose Install Location

Choose the folder in which to install Simufact Forming 2022.

Setup will install Simufact Forming 2022 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Program Files

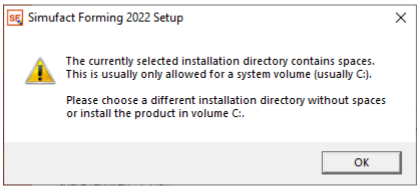
Space required: 5.2 GB

Space available: 90.0 GB

Hexagon AB and/or its subsidiaries

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.



Simufact Forming 2022 Setup

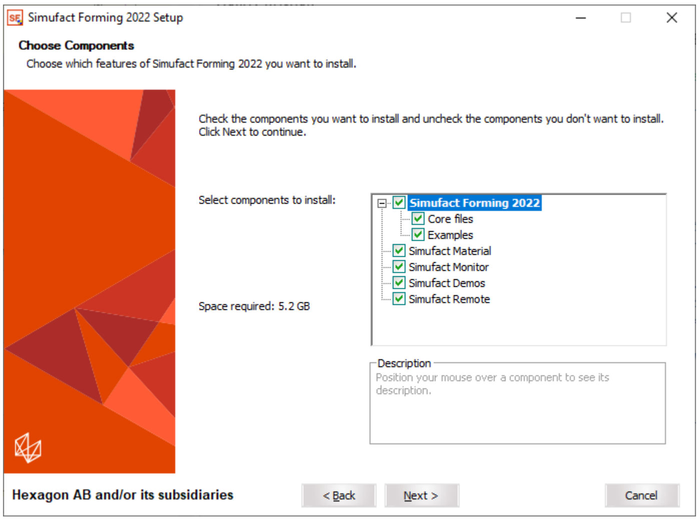
The currently selected installation directory contains spaces. This is usually only allowed for a system volume (usually C:).

Please choose a different installation directory without spaces or install the product in volume C:.

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.



Simufact Forming 2022 Setup

Choose Components

Choose which features of Simufact Forming 2022 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- ☑ Simufact Forming 2022
  - ☑ Core files
  - ☑ Examples
- ☑ Simufact Material
- ☑ Simufact Monitor
- ☑ Simufact Demos
- ☑ Simufact Remote

Space required: 5.2 GB

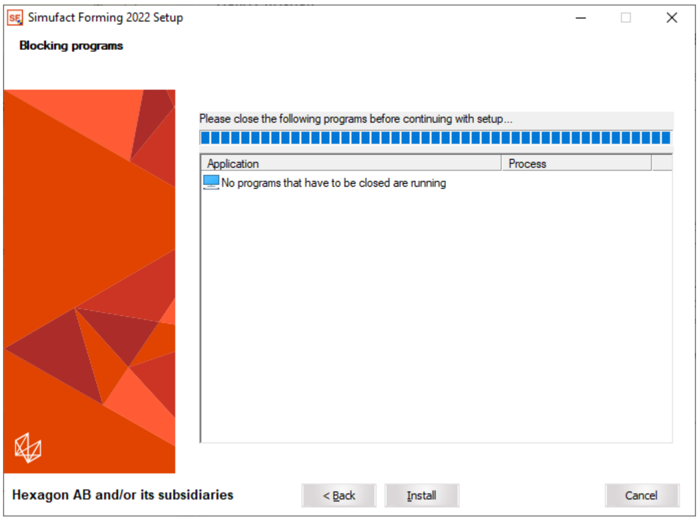
Description: Position your mouse over a component to see its description.

Hexagon AB and/or its subsidiaries

Figure 1.6. Choosing components

By default the following products are selected to be installed: Simufact Forming, 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.



Simufact Forming 2022 Setup

Blocking programs

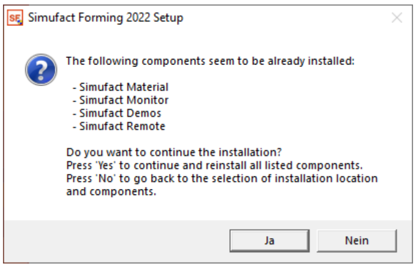
Please close the following programs before continuing with setup...

Application	Process
No programs that have to be closed are running	

Hexagon AB and/or its subsidiaries

Figure 1.7. (Empty) list of blocking programs

Older versions of Simufact Material, Simufact Monitor, Simufact Demos and Simufact Remote 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.



Simufact Forming 2022 Setup

The following components seem to be already installed:

- Simufact Material
- Simufact Monitor
- Simufact Demos
- Simufact Remote

Do you want to continue the installation?
Press 'Yes' to continue and reinstall all listed components.
Press 'No' to go back to the selection of installation location and components.

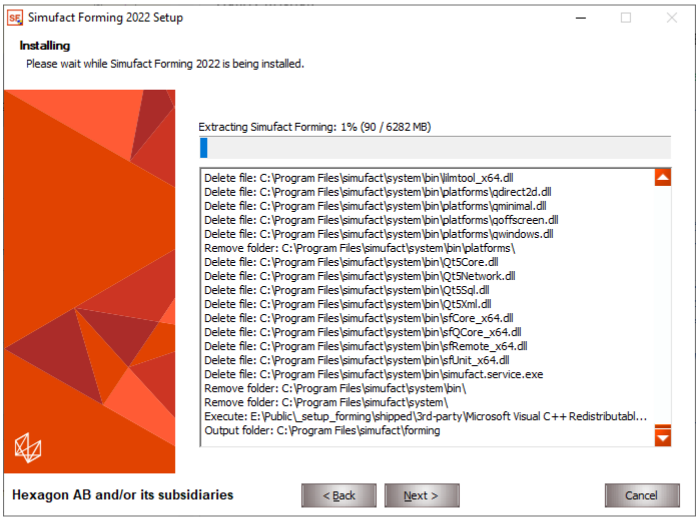
Figure 1.8. Reinstallation of utility program components

Simufact Forming 2022.1 can be used in parallel with previous versions. Please note that you cannot run different versions of Simufact Material, Simufact Monitor, Simufact Demos and Simufact Remote 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.



Simufact Forming 2022 Setup

Installing

Please wait while Simufact Forming 2022 is being installed.

Extracting Simufact Forming: 1% (90 / 6282 MB)

```
Delete file: C:\Program Files\simufact\system\bin\ilmtool_x64.dll
Delete file: C:\Program Files\simufact\system\bin\platforms\qdirect2d.dll
Delete file: C:\Program Files\simufact\system\bin\platforms\qminimal.dll
Delete file: C:\Program Files\simufact\system\bin\platforms\qoffscreen.dll
Delete file: C:\Program Files\simufact\system\bin\platforms\qwindows.dll
Remove folder: C:\Program Files\simufact\system\bin\platforms\
Delete file: C:\Program Files\simufact\system\bin\Qt5Core.dll
Delete file: C:\Program Files\simufact\system\bin\Qt5Network.dll
Delete file: C:\Program Files\simufact\system\bin\Qt5Sql.dll
Delete file: C:\Program Files\simufact\system\bin\Qt5Xml.dll
Delete file: C:\Program Files\simufact\system\bin\sfCore_x64.dll
Delete file: C:\Program Files\simufact\system\bin\sfQCore_x64.dll
Delete file: C:\Program Files\simufact\system\bin\sfRemote_x64.dll
Delete file: C:\Program Files\simufact\system\bin\sfUnit_x64.dll
Delete file: C:\Program Files\simufact\system\bin\simufact.service.exe
Remove folder: C:\Program Files\simufact\system\bin\
Remove folder: C:\Program Files\simufact\system\
Execute: E:\Public\_setup_forming\shipped\3rd-party\Microsoft Visual C++ Redistributabl...
Output folder: C:\Program Files\simufact\forming
```

Figure 1.9. Installation progress

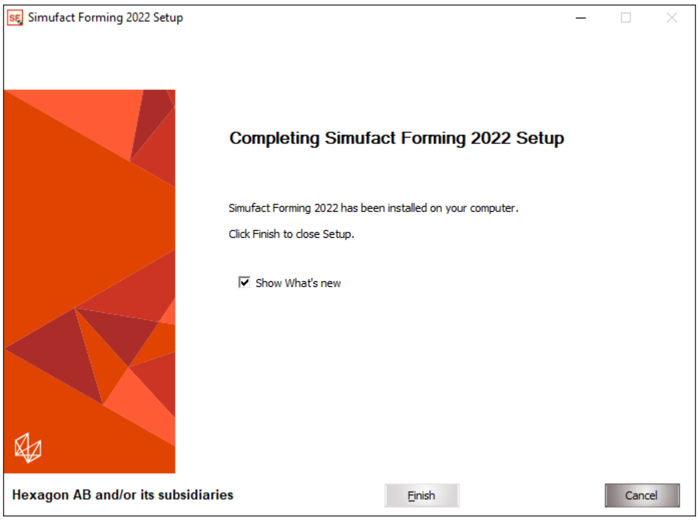
By default, the file extension `.sfp` is associated with Simufact Forming.

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 massage will tell you "The installation of Simufact Forming finished. For further configuration please start Simufact Forming from the desktop after the setup is completed." If you get this message start Simufact Forming 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.



Simufact Forming 2022 Setup

Completing Simufact Forming 2022 Setup

Simufact Forming 2022 has been installed on your computer.

Click Finish to close Setup.

☑ Show What's new

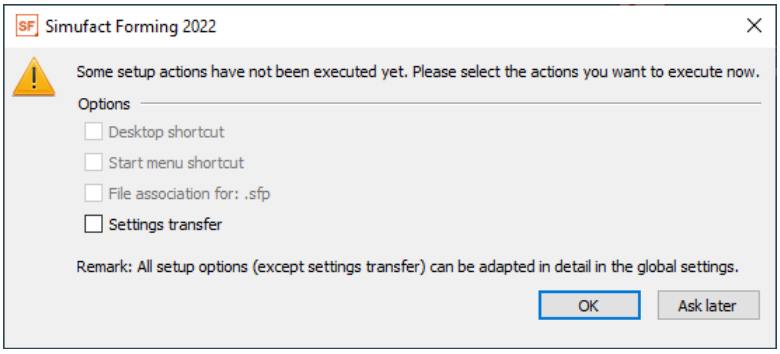
Figure 1.10. Installation finished

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 Forming 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.



Simufact Forming 2022

Some setup actions have not been executed yet. Please select the actions you want to execute now.

Options

- □ Desktop shortcut
- □ Start menu shortcut
- □ File association for: .sfp
- □ Settings transfer

Remark: All setup options (except settings transfer) can be adapted in detail in the global settings.

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 Forming, 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 Forming supports the usage of 3DConnexion™ SpaceDevices. The sensitivity of the SpaceDevice can be adjusted in the setting of Simufact Forming. 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 Forming	/Forming=yes
To install Examples	/Examples=yes
To install Simufact Demos	/Demos=yes
To install Simufact Material	/Material=yes
To install Simufact Monitor	/Monitor=yes
To install Simufact Remote	/Remote=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 copyright law and
international treaties. Unauthorized reproduction or distribution of this pro-
gram, 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 /Forming=yes
          /Examples=yes /Demos=yes /Material=yes /Monitor=yes
          /Remote=yes /D=C:\Users\<username>\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 /Remote=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\<username>\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 Forming 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 4.1.2)

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 Forming. Alternatively you can use the generated uninstaller `Uninstall Simufact Forming 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 Forming 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 4.1.2. 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 Forming, 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.

1.2.1. 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 Forming download package `simufact.forming.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 xzvf simufact.forming.2022.1_linux.tar.tgz -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. There are two alternative options for this:

- Use option "**1**" of the configuration script `.../simufact/config.forming/2022.1/config.exe`. This will set MSC_LICENSE_FILE in the solver start scripts `run_marc` and `sfDytran` in a way that it becomes only effective if the environment does not already contain MSC_LICENSE_FILE. If this option is used and the environment contains MSC_LICENSE_FILE, only the value from the environment will be used. The value set in the scripts will be ignored in this case. If you have multiple license servers, use **:** to string them together in MSC_LICENSE_FILE

- 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 Simufact Remote and / or your scheduling engine. Simufact Remote uses `sudo` to change user names and may be started using `sudo` during the system start-up. Simufact Remote is discussed later in this installation guide (Chapter 2).

4. Optionally: Link the solver start scripts into a directory in the regular search PATH. You can use option "**2**" of the configuration script `.../simufact/config.forming/2022.1/config.exe` for this. This will create soft links including the version number.

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 Forming, 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:

```
simufact/forming/2022.1/sfMarc/tools/run_marc       FE solver script
   "                  sfDytran/bin/sfDytran       FV solver script
   "     system/bin/*                             Simufact Remote service
   "     config.forming/2022.1/config.exe           Configuration script
   "                        /Ins...guide....pdf   This guide
```

1.2.3. User subroutines

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

```
.../simufact/forming/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 3, too.

1.2.4. Running Simulations on Linux

As only the solvers are part of the Linux version of Simufact Forming, 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 3 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 one of the next possibilities.
- Use Simufact Remote to couple the GUI on Windows with the Solvers on Linux. This offers a comfortable method to submit jobs and to monitor their progress. Simufact Remote is discussed in the next chapter and later in this installation guide (Chapter 2). It is described in the relevant Infosheets, too. Simufact Remote can interact with your scheduling engine.
- Use custom specific **Start Scripts** to couple the GUI on Windows with the Solvers on Linux. This offers more flexibility than Simufact Remote, but requires you to write them. 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 `sf_testjob_fe` for testing the FE solver is provided in `.../simufact/forming/2022.1/sfMarc/tools/test`.

A small solver input called `testjob_fv` for testing the FV solver is provided in `.../simufact/forming/2022.1/sfDytran/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 Forming 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 may have to use the option **Retrigger result import** form the **Extras** menu once to get this started.

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/forming/2022.1/sfMarc/tools/run_marc` respectively `.../simufact/forming/2022.1/sfDytran/bin/sfDytran` depending on the solver type used. Do not start the solver executables directly! The options needed by the solver start scripts can be taken from the file `runsf.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. For the FE solver check the comments in the solver start script and the Technical Reference (VolA) about their meaning. For the FV solver call the solver start script without any arguments to receive a summary of possible arguments.

The most frequent 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 4.2.5. The parallelisation options are discussed in more detail in Section 1.2.5.4. FE 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.

The most frequent options of `sfDytran` are:

- `jid=jobID`

  `jobID` is the name of the solver input dat-file without the extension `.dat`.

- `dmp=yes/no/both`

  Specifies the type of parallel simulation. `no` is the default and enables SMP (Shared Memory Parallel), `yes` enables DMP (Distributed Memory Parallel), `both` enables both.

  - Please note that there a limitations in the possible combinations of Solver type, Parallelization options, the Number of domains and the Number of cores used. Stick with the options in runsf.bat written by the GUI.

- `ncpus=N`

  `N` specifies the number of cores used for parallel computing. Depending on `dmp` this is the number of domains, too.

  - Please note that there a limitations in the possible combinations of Solver type, Parallelization options, the Number of domains and the Number of cores used. Stick with the options in runsf.bat written by the GUI.

- `intelmpi=yes/no`

  On Windows only: Use `yes` to activate the Intel MPI for `dmp=yes`

- `bat=yes/no`

  `yes` will start the simulation in the background detaching from the calling shell. `no` will start the simulation interactively in the foreground with the calling shell waiting for the simulation to end. For fast tests, in scripts and for queuing typically `no` is used.

- `ask_rem=yes/no/rename`

  `yes` is the default and will interactively ask whether to overwrite existing result and output files of previous runs of the same jobID. `no` will silently remove / overwrite these files. `rename` will automatically rename these files. Do not use rename as the GUI may not recognize renamed files.

Example for starting simulations with increasing complexity are given in Section 4.2.5. The parallelisation options are discussed in more detail in Section 1.2.5.4. FV simulations started by scripts or by a scheduling engine are typically started with some thing like

```
sfDytran jid=jobID bat=no ask_rem=no \
         dmp=yes ncpus=8 > jobID_std.log 2>&1
```

`\`and line-break inserted for readability, not to be included in the command line. `jobID_std.log` is used by the GUI for this, too and should be used to keep consistent.

1.2.5. For administrators

1.2.5.1. Prerequisites and compatibility

For a complete installation of Simufact Forming 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.
- Python in a version 2.x (not 3.x!).

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

- Redhat Enterprise Server 7.9

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. Installing and Managing the Simufact Remote server

Simufact Remote couples the GUI on Windows with the Solvers on Linux. This offers a comfortable method to submit jobs and to monitor their progress. In this chapter the installation and configuration of the Simufact Remote server on Linux is discussed. More information, especially about the Windows side, can be found later in this installation guide (Chapter 2) and in the relevant Infosheets, too. Simufact Remote can interact with your scheduling engine (if used).

Installing the Simufact Remote server ("Simufact Service") requires you to have administrative rights (=root permissions). So, log in as *root* (“su” or “sudo -i”) and start the configuration script `.../simufact/config.forming/2022.1/config.exe`. Select the option "**3**" for the Simufact Service. Continue with sub-option "**5**" to configure the Simufact Service. Follow the given instruction to register the service to the `init.d` system, to configure the working directory for the simulations and to set the user used to run the service. Concluding the installation you can start the service. Once the installation process has finished, you can quit the script by entering "**q**" in any menu. During the installation a start/stop script with the name `sfserviced` is written into the system folder `/etc/rc.d/` (or `/etc/init.d/`). You can repeat this process any time if you want to reset the contents of the written config files and the server start script.

Afterwards option "**3**" of the configuration script offers the possibility to start / stop the Simufact Service and to subscribe / unsubscribe it from the the `init.d` system by utilising the sub-options "**1**" to "**4**". You can repeat the installation / configuration any time if you want to reset the contents of the written config files and of the server start script using sub-option "**5**".

The Simufact Remote server will run as a process called `simufact.service`. The Simufact Remote server runs as Linux™ daemon in the background, just like other system services, e.g. `nfsd` or `smbd`. They are controlled by a special run script that is written to the `/etc/rc.d` or `/etc/init.d` folder during the installation. Please do not try to run the server application as standalone executable. For starting and stopping or querying its status, always use the script `sfserviced` in `/etc/rc.d` or `/etc/init.d`, respectively. Being *root*, call the scripts with one of the options `start`, `stop` or `status`. If you want to test whether the remote server is up and running, you can use the command:

```
/etc/rc.d/sfserviced status
```

Additionally, the configuration script `config.exe` provides a much easier way to control the daemon. The sub-options for the Simufact Service, option "**3**", offers the basic functionalities for this service.

For the location of the ini-files storing the settings of the Simufact Service please see the Troubleshooting chapters of this guide.

If `simufact.service` is not run as `root` (running as `root` is not recommended), `/etc/rc.d/sfserviced` uses `sudo` to start the daemon as `$USER` (with `$USER` being the username used to run `simufact.service`). Depending on your system settings, this may require further configuration, especially in systemd environments. You may have to add some thing like

```
Defaults>$USER !requiretty
```

to the end of the `sudoers` file using `visudo`. Please check the manpages for `sudo` and `visudo` for more information.

- Behind the curtains, Simufact Remote uses the `sudo` command to start processes (the simulations) under a different user account. So you have to ensure that users can start the FE/FV solver executable and meshers via `sudo`, and that the `requiretty` option is deactivated in your current `sudo` configuration.

  To deactivate this for all users, add a line like

```
Defaults !requiretty
```

to the end of the `sudoers` file using `visudo`. If you are running the service under an account without root privileges, you can also just deactivate this setting for that account:

```
Defaults:$USER !requiretty
```

with `$USER` being the username used to run `simufact.service`. In this case you also have to allow `$USER` to use `sudo` in the first place. Please see the `sudoers` file for more information on how to do that.

Please check the manpages for `sudo` and `visudo` for more information.

- If a firewall is used, the ports 9987, 9988 and 9990 are to be enabled for Simufact Remote. The Simufact Monitor uses port 9985 for communication. MSC Licensing uses additional ports (see Section A.2)
- Ensure that the environment variable MSC_LICNESE_FILE is configured correctly for the user(s) used to run the simulations. Otherwise the licensing may not work correctly. Consider setting MSC_LICENSE_FILE in the solver start scripts using `config.exe`.
- Behind the curtains, Simufact Remote uses the `sudo` command to start processes. This influences the effective environment variables, too. MSC_LICENSE_FILE and potentially some environment variables of your scheduling engine (if used), may require special configuration because of this. You can test your configuration by reviewing your environment after `sudo -u user /bin/sh`. Consider setting MSC_LICENSE_FILE in the solver start scripts using `config.exe`.
- For more details see Section 1.2.6, too.

1.2.5.4. Scheduling of jobs

Simufact Forming 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 basic task of scheduling a job is to create the correct command line for the simulation as discussed in Section 1.2.4 and to submit it to the scheduling engine to be executed. If needed a `hostfile` has to be created based on the output of the scheduler before the execution of the command line actually starts.

The configuration and scripts discussed in this chapter focus on the requirements of Simufact Remote and the functionalities used in the sample **Start Scripts** provided. If you are not using Simufact Remote, you may feel free to use your own configuration.

The basis of scheduling jobs in combination with Simufact Remote or with the sample **Start Scripts** provided 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/forming/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 the 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 scripts 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/forming/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 `runsf.bat` file as it is used by Simufact Remote (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 scripts `schedule_sfdytran.sh` and `schedule_marc.sh` are provided in `.../simufact/forming/2022.1/sfTools/sfScheduling/testscripts`. They use the same combination of `write_job` and `schedule_job`.

```
#!/bin/sh
cd RUNDIR

".../simufact/forming/2022.1/sfTools/sfScheduling/write_job" \
   "RUNDIR/_jid.job" \
   ".../simufact/forming/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/forming/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 `runsf.bat` running while it is waiting for the job to be started by the scheduling engine and while it is being calculated. This is needed for the Simufact Remote server 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 `runsf.bat` exits for any reason, the Simufact Remote server assumes that the simulation has stopped. 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 in a similar way.

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

If Simufact Remote is used to start simulations from the GUI on Windows machines, please keep in mind that only remote profiles with the scheduling options enabled will make use of your scheduling engine. If custom **Start Scripts** are used for this, they need to adapted to scheduling, too.

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 4.1.2. This will probably break the objective of the scheduling engine.

The scheduling scripts shipped with Simufact Forming 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.

- Finite Elements (FE) or Finite Volume (FV)?

  Check the command line for the solver: `run_marc` = FE, `sfDytran` = FV

- `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 `schedule_job`	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 `-nts`	no	1	as per `-nts`	no
FE, domain decomposition (DDM)	as per `-nps`	-	yes (if distributed to different hosts)	same as per `-nps`	as per `-nps`	yes
FE, DDM and shared memory parallelisation	as per `-nps`	as per `-nts`	yes (if distributed to different hosts)	same as per `-nps`	as per `-nts`	yes, in packages of `Number of Threads / Number of Domains` cores

FV, no parallelisation	-	-	no	1	1	no
FV, shared memory parallelisation	-	as per ncpus	no	1	as per ncpus	no
FV, distributed memory parallelisation	as per ncpus	-	no	same as per ncpus	as per ncpus	no

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 of FE simulations 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_FORM_SOLVER` for each job
- for FE simulations "total number of cores used (maximum)" - 1 `SF_FORM_NODE` for jobs using parallelisation
- for FV simulations "total number of cores used (maximum)" - 4 `SF_FORM_NODE` for jobs using parallelisation (if greater 0)
- if used: licenses for additional modules, can not be determined form the command line. Typically Simufact Forming 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`.

1.2.5.5. Uninstallation

Simufact Forming 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.

So, for a complete uninstallation you should use the `config.exe` script to stop and unsubscribe your running servers first (see also Section 1.2.5.3). Then you can safely remove the server script `sfserviced` in `/etc/rc.d/` (or `/etc/init.d/`) and finally delete the folder `simufact/` or if other version or other Simufact software is installed the relevant sub-folder only.

1.2.6. Some notes on configuration and security of the Remote Server

The Simufact Remote server, thus the program simufact.service, enables users on Windows™-based clients to run simulations on a Linux™ machine. Depending on your environment and your security needs, there are a lot of possible configuration options. For sure, the server and the client configuration have to match together, thus the configuration given in the remote manual for the clients may or may not match your configuration. Basic variables of the configuration are the user used to run the remote server, the user(s) used to run the simulations and the data exchange method between client and server.

The current remote server of version will work seamlessly simultaneously with older versions on the client side. The server side should always be updated first and have at least the same version as the clients used. Server versions older than the client are not supported and known to be buggy.

It is recommended to create a special user used to run the remote server, especially in a multi-user environment. In this manual this user will be called *sf_remote*.

The remote server is **not** designed and tested to run as *root*. It should always run as an unprivileged user to avoid possible security holes. The user used to run the remote server needs a home directory to store the remote configuration and depending on the configuration he needs to be able to log in, too.

Within the forming GUI on the client the user can specify a custom solver executable that is copied to and executed on the remote server when a remote job is submitted. This enables the user used to run the simulation to execute arbitrary code on the server. Thus the remote server should never run as *root* and the user used to run the simulation should be selected with care. User subroutines are compiled and executed on the server, this enables the user used to run the simulation to execute arbitrary code on the server, too.

There are 2 different methods for the data exchange between the remote server and the client. The server offers both methods, they are chosen on the client. These methods are:

- **File synchronizer**

  When the file synchronizer is used, the data exchange is done completely by the Simufact programs on the client and the server side. The only required configuration is the directory used on the server for the temporary storage of the simulation files during the calculation. You are asked for this by the configuration script `config.exe` (see Section 1.2.5.2). There should be plenty of free disk space, the directory must be fully accessible (read, write, execute) for *sf_remote* and the user(s) used to run the simulation. The advantage of the file synchronizer is that it is easy to configure. The drawback is that the access to the files in the working directory of a simulation is limited from the client side (if not additionally enabled by other means). But the main information is available within Simufact Forming on the client. The **load on the network** is considerably higher when the File synchronizer is used compared to the usage of a Mapped network drive, especially when the Simufact Forming project is stored on the remote server, too

  When the file synchronizer is used, the data exchange is done by the user *sf_remote*, not by the user(s) used to run the simulations. Thus *sf_remote* must be able to read and delete the files created during the simulations by the user(s) used to run the simulations. On the other hand, the user(s) used to run the simulations must be able to create new files in directories created by *sf_remote*. Take care to set umask, group membership and primary group appropriate.

- **Mapped network drive**

  When a mapped network drive is used, the Simufact programs on the client and on the server side only exchange control information. The data exchange is done by Simufact Forming using your Microsoft™ Network. For this to work, the Windows™ clients have to map a network drive with a drive letter to a share on your Linux™ remote sever machine, which has to be configured to enable this. Therefore you need to install and configure a 3rd party software on your Linux™ machine, for example Samba (www.samba.org [https://www.samba.org/]). This software is not shipped with Simufact, not checked or configured by the configuration script `config.exe` (see

Section 1.2.5.2) and not supported by Simufact. Normally your Linux™ distribution should include a reasonably working and configured version of Samba. Each client can map a different share or directory. Typically the users home directory or subdirectories are used. The used directory needs plenty of free disk space, the user used to run the simulation needs full access to it and *sf_remote* needs to be able to change into it, but does not require read or write permissions (thus only execution permissions are fine). The advantage of using a mapped network drive is that there is full and easy access to all files in the working directory of a simulation from the client side, which will in particular be useful for more experienced users. The drawback is some additional configuration effort.

For the Simufact Forming Graphical User Interface (GUI) to monitor the simulation status and to import the simulation results while the simulation is running, complete data consistency between the Windows™ client machine and the Linux™ server machine is needed. File updates must not be delayed or hindered by file locking or caching. If you are using Samba on the Linux™ machine for the SMB/CIFS file shares in the Windows network consider setting `kernel oplocks = yes` in the share definition in `smb.conf` to achieve this. See the man page of `smb.conf` for details.

The remote server does not offer a configuration option to prevent or to enforce the clients to use one or the other method for the data exchange. If you want to prohibit the use of the file synchronizer, you can set the directory of the temporary storage on the server to */dev/null*. If you want to prohibit the use of the mapped network drive, you can either not configure shares that can be mapped at all or prevent access to the shared directories for *sf_remote* or drastically limit the storage space in the shared directories.

If nothing is specified on the client, the user used to run the remote server and the user used to run the simulations are identical. Thus all simulations will run as *sf_remote*. In a lot of cases this will be fine, especially when the file synchronizer is used in a trusted environment. But there are good reasons to use individual users to run the simulations, for example:

- Security concerns with shared data access, especially in combination with using mapped network drives.
- The wish to have meaningful user names in the usage statistics.
- The wish to have meaningful user names in a job queue.
- To enable users to kill (not responding) simulations without the risk to kill other simulations. The users need shell access to the server for this.
- To allow the users safe but full access to all files in the working directory of their simulations. Besides monitoring and influencing running jobs (which is in particular useful for more experienced users), this can be useful to clean up the files in cases where the file synchronizer or the GUI fail to do so at the end of the simulation. Shell access to the server may be useful for this.

Licensing needs to be configured correctly for all users used to run the simulations. See Section 1.2.1 number 3) for possible options.

The Simufact remote server should work with users managed by a Windows™ domain controller and provided to the Linux™ remote server using winbind or a similar 3rd party software not shipped, configured or supported by Simufact. In this case specify the user used to run the simulation as *domain\user* on the client and take care about the required user configuration mentioned before.

If different users are used to run the simulations, the user used to run the remote server `simufact.service` - here called *sf_remote* - must be able to start processes under these user names. Behind the curtains the `sudo` command is used to do this. The required `sudo` configuration is not done by the configuration script `config.exe` (see Section 1.2.5.2) but has to be done by you manually using `visudo` to edit the configuration file `/etc/sudoers` (see Appendix E). For detailed documentation see the related man pages. There are 2 required configurations:

- Allow *sf_remote* to use `sudo`, for example using:

```
sf_remote ALL=(ALL, !root)  ALL
```

This will allow *sf_remote*, thus the user used to run the remote server, to run any commands for all users except root on all machines using this shared `sudo` configuration. It is not easily possible to limit the allowed commands as the remote server executes a script in the automatically generated working directory of each simulation. But you can limit the users and the machines if you want to.

- Allow *sf_remote* to use a remote connection when `sudo` asks for a password:

  ```
  Defaults:sf_remote !requiretty
  ```

  This will allow *sf_remote*, thus the user used to run the remote server, to submit a password to `sudo` without allocating a tty. You probably do not want to do this for all users.

The normal behavior of `sudo` is to ask for the password of the invoking user to verify him and then to enable the execution of predefined commands under a different user without knowing the password of the other user. This is pretty handy to enable some users to do things that normally only *root* can do, for example to manage printers. For the remote server this would mean that the client has to be configured with the name of the user used to run the simulation, but with the password of *sf_remote*. On the one hand this would confusing, on the other hand this is a **security hole**: Once a user has shell access to the server, he can easily find out the user name of *sf_remote*, he knows the password from the client configuration, ... and can do anything under anybody's name. Thus you probably want to

- Change the password behavior for *sf_remote* using `sudo`:

  ```
  Defaults:sf_remote targetpw
  ```

  This will force *sf_remote*, thus the user used to run the remote server, to supply the password of the user used to run the simulation - generally of the user used to execute a command with `sudo` - instead of his own password. In this case the client has to be configured with the name of the user used to run the simulation and his password. User and password have to be valid on the Linux™ remote server, they may differ form the Windows™ user and password used on the client. If applicable, consider the requirements for the mapped network drive. Be careful not to switch to `targetpw` for all users, as this may break other things on your system.

The remote server does not offer a configuration option to force the clients to specify a user name. To prevent the clients from running their simulations using *sf_remote*, you can remove the execution right from the solver scripts `simufact/forming/2022.1/sfMarc/tools/run_marc` and `simufact/forming/2022.1/sfDytran/bin/sfDytran`. An other option to prevent simulations to be run by `sf_remote` is to add an option file to MSC Licensing using the `EXCLUDE` direction to prevent sf_remote access to the needed licenses.

1.2.7. Queueing

If no more licenses are available, the simulations will not start or continue but reach the status "Waiting for License", see Section 4.1.2. 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 1.2.5.4.

2 Simufact Remote

Simufact Forming 2022.1 offers the possibility to run the Graphical User Interface (GUI) used for interactive pre- and postprocessing and the solvers used to run the simulations in the background on different computers. This enables to use powerful central computer resources for the simulations. Especially for long running simulations it can increase the stability to use dedicated computers. There are two possibilities to automatically link the GUI and the solvers on different computers:

- Simufact Remote, a Simufact auxiliary client and server application specially designed for this purpose. After little configuration it offers a comfortable method to submit simulations from the GUI to the solver running on a different computer and to monitor their progress. While the simulations are running, the GUI can be closed and even the computer used for it can be shutdown. Simufact Remote is described in this chapter.

- Custom specific **Start Scripts**. This offers more flexibility than Simufact Remote and enables special configurations, but requires you to write them. Samples are provided. See the relevant Infosheet. Typically the Start Scripts require that the computer used for the GUI is running while the simulation is running. Start Scripts can be used to starts simulations locally, too, for example to couple them into a local 3rd party scheduling engine. Start Scripts may for example also be configured to send e-mails once a simulation has finished.

In the following, the computer where the analysis is started with the Simufact Forming GUI will be referred to as the CLIENT. The name of the client is irrelevant here. The computer where the solver runs will be referred to as the SERVER. The name of the SERVER in this guide is usually SWK. Some pictures might be easier to understand with this knowledge in mind.

Both, Simufact Remote and the **Start Scripts** assume that the data storage is done by the GUI on its CLIENT computer. Only the solvers and their temporary files are shifted to SERVER computer, not the management of the simulation data. **Both are NOT some thing like a remote application, remote desktop or similar system.** Care needs to be taken not to overload computers and networks with data traffic. Simufact Remote is made for stable and fast local networks. Especially configurations with the Simufact Forming project stored on a network share of the **Simufact Remote** SERVER can cause huge network traffic, in particular if the File Synchroniser is used.

Before starting, please ensure that you have administrative rights on both CLIENT and SERVER.

2.1. Installation

Simufact Remote is installed as part of the normal installation of Simufact Forming. On Windows™ it can deselected during the installation. Thus, if you want to use Simufact Remote but you can not find it, it may not be installed. In this case rerun theSimufact Forming setup and ensure that Simufact Remote keeps selected. You can deselect already installed components during the rerun of the setup. They will remain untouched in this case.

The version of Simufact Remote (which is typically related to the installed version of Simufact Forming) used on the SERVER should always be the same or a newer version as the one used on the CLIENTs. The SERVER will work seamlessly simultaneously with older versions on the CLIENT. But versions on the SERVER older than on the CLIENTs are not supported and known to be buggy. Therefore it is recommended to install new versions on the SERVER and on all CLIENTs at the same time or at least to update the SERVER first.

If this constrain for the version of Simufact Remote are fulfilled, multiple versions of Simufact Forming can installed on the CLIENTs and on the SERVER and be used with Simufact Remote simultaneously. Simufact Remote is an auxiliary program shared by different versions of Simufact Forming. If the default settings are used and the same installation path is used for all version of Simufact Forming it will be updated as needed if a new version of Simufact Forming is installed.

The installed version of Simufact Remote can be checked for local and remote installations in the **Simufact Remote Manager** introduced later.

2.2. Configuration

2.2.1. Introduction

This configuration guide is meant for Windows™ administrators who have installed Simufact Forming and Simufact Remote on the SERVER and at least one CLIENT computer and want to configure these programs for remote usage. For the configuration on Linux™ (SERVER only) see Section 1.2.5.3.

2.2.2. Server configuration

1. Log in to the SERVER as the user *Administrator*.

2. Open the firewall settings.

   Open the ports 9985, 9987, 9988 and 9990 for inbound and for outbound connections. Port 9985 is for the Simufact Monitor, Simufact Remote itself uses the ports 9987, 9988 and 9990. See Appendix B.

3. Create a working directory on a drive with sufficient free space, for example *D:\work.*

   If the CLIENT uses a Mapped network drive (Figure 2.4), the drive needs to be mapped as a network drive by the CLIENT. The CLIENT needs access through the network to this working directory, so the directory has to be shared in the network. Grant read and write permissions to all users of the CLIENT who want to use the SERVER.

   If the Simufact Forming project file is stored on a network share of the SERVER, it is highly recommended to to use Simufact Remote with Mapped network drive as method for data exchange. Otherwise the network traffic will be significantly increased.

   If the CLIENT uses the File synchronizer, no more changes are required.

   For each analysis started from the CLIENT a subdirectory with a unique name will be created in the working directory. That subdirectory will temporarily contain all input and output of the analysis.

4. Install Simufact Forming, configure its license and start an analysis of an example project locally on the server as a test.

5. On the server, start Simufact Forming as Administrator and open the Simufact Remote Manager (see Figure 2.1).

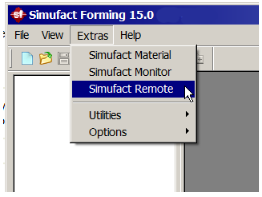
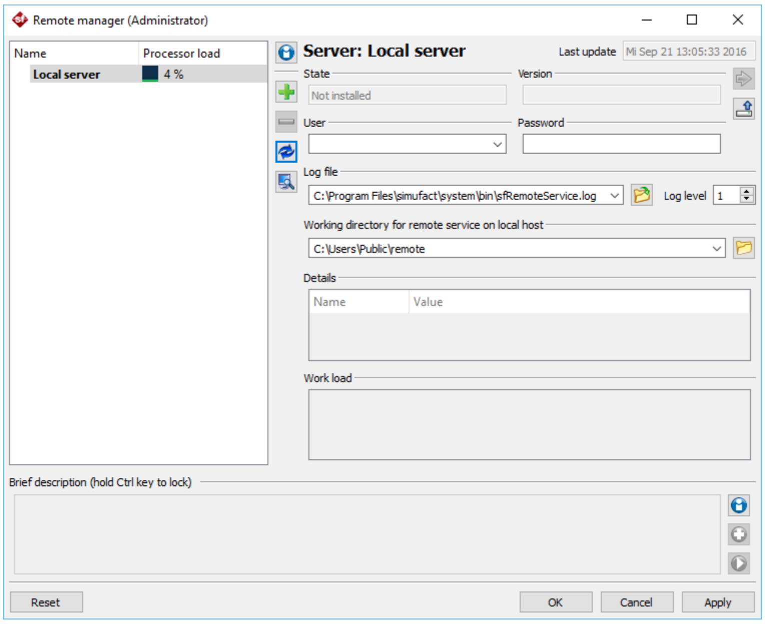


Figure 2.1. Opening the remote manager in Simufact Forming

6. In the Remote Manager, stop the local server under **Local server** with the ○ button in case it is running. The status text and the buttons for installing, starting, and stopping show if the service is running. The local server is the `simufact.service.exe` running as a service on the SERVER enabling the CLIENTs to start simulations on the SERVER.

7. Select the working directory you created in step 3.

8. Start the remote service by clicking the ➡ button. If the service has not been installed (Figure 2.2), you have to click the installation button ⬆.



Remote manager (Administrator)

Name	Processor load
Local server	4 %

Server: Local server

Last update: Mi Sep 21 13:05:33 2016

State: Not installed

Version:

User:

Password:

Log file: C:\Program Files\simufact\system\bin\sfRemoteService.log

Log level: 1

Working directory for remote service on local host: C:\Users\Public\remote

Details

Name	Value

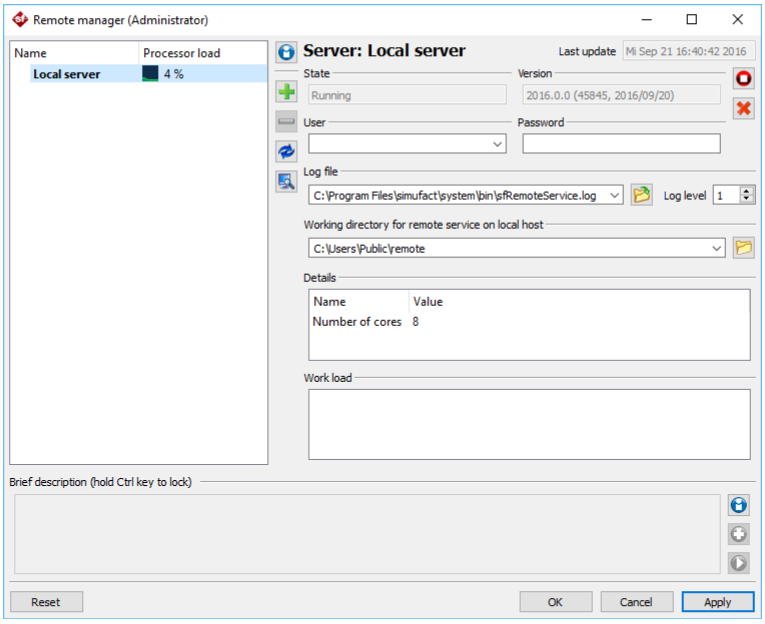
Work load

Brief description (hold Ctrl key to lock)

Figure 2.2. Local server not installed

Do not enter a user or password, so the service will be installed for your local user SYSTEM. After the installation of the service it will be started automatically for that user.

Please make sure that the status of the service shows `Running` since otherwise Simufact Remote cannot work properly (Figure 2.3).



Remote manager (Administrator)

Name	Processor load
Local server	4 %

Server: Local server

Last update: Mi Sep 21 16:40:42 2016

State: Running

Version: 2016.0.0 (45845, 2016/09/20)

User:

Password:

Log file: C:\Program Files\simufact\system\bin\sfRemoteService.log

Log level: 1

Working directory for remote service on local host: C:\Users\Public\remote

Details

Name	Value
Number of cores	8

Work load

Brief description (hold Ctrl key to lock)

Figure 2.3. Local server running

If you change your license configuration, you need to stop and start again the Simufact Remote service ("Local server") afterwards (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.

For the location of the ini-files storing the settings of Simufact Remote please see the Troubleshooting sections of this guide.

2.2.3. Client configuration

Execute these steps on the CLIENT from where you want to start analyses on the SERVER.

1. Open the window with the firewall settings as an administrator. Open the ports 9985, 9987 and 9988 for inbound and for outbound connections. Port 9985 is for the Simufact Monitor, Simufact Remote itself uses the ports 9987 and 9989. See Appendix B.

2. Log in to the CLIENT as your real user. You have to execute all of the following steps for every user on this CLIENT that wants to use the SERVER.

3. If you want to use the setting Mapped network drive Figure 2.4, you have to map the working directory of the SERVER as a network directory (see Section 2.2.2, step 3 and step 7). After successful mapping you should be able to access the working directory of the server through the local path (here: R:\). Make sure that your user has write permissions on the mapped working directory. This step is not necessary if you only want to use the remote File synchronizer.

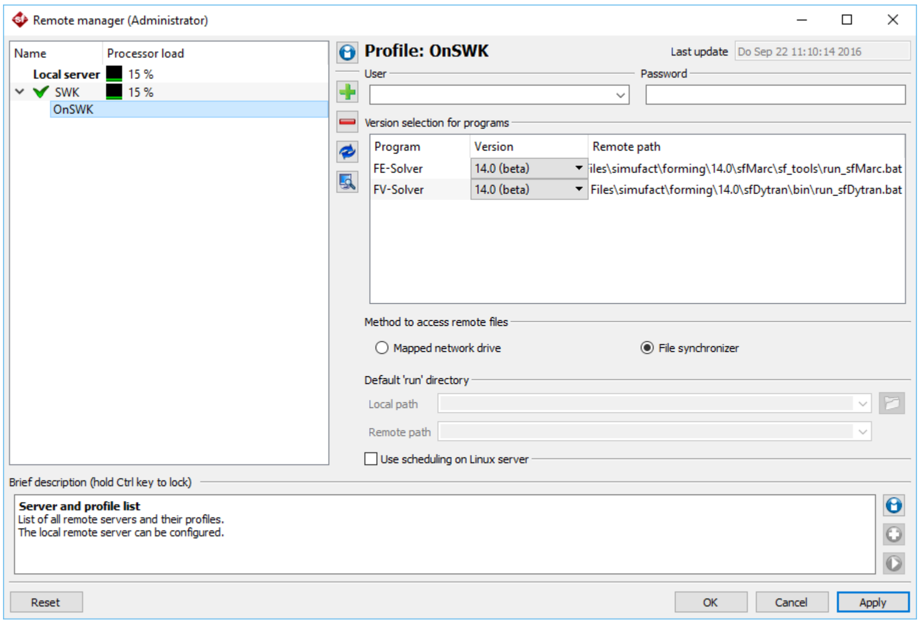
4. Start Simufact Forming and open the remote manager (Figure 2.1).

5. Add the SERVER to the server and profile list by clicking ✚ on the right of the processor load. In the selected field of the server list enter either

   - the computer name of the SERVER (here: SWK) or

   - the IP address of the SERVER.

6. Select the profile of the SERVER you just entered (here: OnSWK) and change the settings as follows (Figure 2.4):



Remote manager (Administrator)

Name	Processor load
Local server	15 %
✓ SWK	15 %
OnSWK	

Profile: OnSWK

Last update Do Sep 22 11:10:14 2016

User

Password

Version selection for programs

Program	Version	Remote path
FE-Solver	14.0 (beta)	iles\simufact\forming\14.0\sfMarc\sf_tools\run_sfMarc.bat
FV-Solver	14.0 (beta)	Files\simufact\forming\14.0\sfDytran\bin\run_sfDytran.bat

Method to access remote files

○ Mapped network drive ● File synchronizer

Default 'run' directory

Local path

Remote path

□ Use scheduling on Linux server

Brief description (hold Ctrl key to lock)

Server and profile list
List of all remote servers and their profiles.
The local remote server can be configured.

Figure 2.4. Adjusting the profile of the new server in the remote manager

- **User** and **password** Optional fields. If supported by the operating system the remote service on the SERVER uses these credentials to launch the analysis by creating a process which is owned by the given user. If configured correctly, Linux™ SERVER support this. SERVER with a Windows™ operation system do not support this. Do not enter a user or a password if the operating system of the SERVER does not support this option. If no user name and password are given, the analysis is started with the same user as the `simufact.service.exe` is running with.
- Choose the **Method to access remote files**:
  - When Mapped network drive is used, the Simufact programs on the client and on the server side only exchange control information. The data exchange is done by Simufact Forming using your Microsoft™ Network. For this to work, the CLIENTs have to map a network drive with a drive letter to a share on your SERVER, which has to be configured to enable this. The advantage of using a mapped network drive is that there is full and easy access to all files in the working directory of a simulation from the client side, which will in particular be useful for more experienced users. The 2nd advantage of using a mapped network drive is the possibility to use more restricted file permissions in the working directory on the SERVER, which may be useful in multiuser environments. The drawback is some additional configuration effort. If Simufact Forming is started as administrator - which is generally not recommended and normally not needed - remote simulations using Mapped network drive may fail because the administrator may not have access rights to the share.

    If you want to use the setting Mapped network drive, you have to enter the paths **Local path** (see step 3) and **Remote path** (see Section 2.2.2 step 3). You should also read Example 2.1 for this.

    It is highly recommended to use Mapped network drive as method to access remote files if the Simufact Forming project file is stored on a network share of the remote SERVER, too.

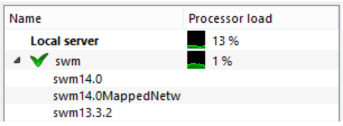
  - When the File synchronizer is used, the data exchange is done completely by the Simufact programs on the CLIENT and the SERVER side. The advantage of the file synchronizer is that it is easy to configure. The drawback is that the access to the files in the working directory of a simulation is limited from the client side (if not additionally enabled by other means). But the main information is available within Simufact Forming on the client. **The load on the network is considerably higher when the File synchronizer is used** compared to the usage of a Mapped network drive, especially when the Simufact Forming project is stored on the remote SERVER, too.

    If you want to use the File synchronizer, you do not have to change any other settings.

7. Click the update button ⟳ to update the profile or the list of programs that can be used on the SERVER and are displayed under **Version selection for programs**.

   Make sure that the list of programs is not empty. A profile with an empty program list is invalid and will not be selectable when you start the analysis.

8. Select the desired solver version in **Version selection for programs** for this remote profile.
9. You can predefine more than one remote profile.



Name	Processor load
Local server	13 %
✓ swm	1 %
swm14.0	
swm14.0MappedNetw	
swm13.3.2	

Figure 2.5. Using several remote profiles

Select the existing profile of a server and click + to add a further profile for this server and adapt the settings as explained in the step 6 and 8 above. Give the new profile a meaningful name using the F2 key to edit the profile name.

Having more than one remote profile for a server makes sense for example if you want to start analyses with different solver versions.

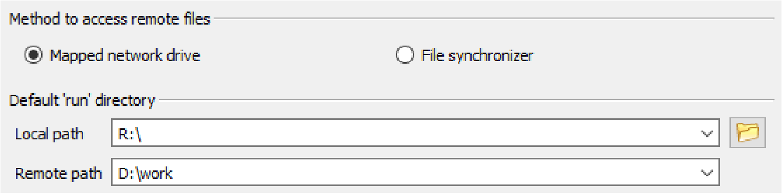
10. If your SERVER has a Linux operating system you have, depending on the configuration, the possibility to use a scheduling engine. This allows the server to place the jobs into a waiting queue and start them as soon as all needed resources are available. Activate the option **Use scheduling on Linux** if you want to use such a scheduling engine. Please consult your local administrator if you should, can or have to use this option.

Example 2.1. Path mapping

The remote system (SERVER) has the path *D:\work\*, where the 'Run' directories of all remote jobs are to be stored. The local computer (CLIENT) needs read and write access to this path. So the remote path *D:\work\* has to be mapped to your local computer.

Under Windows™ you have to add *D:\work\* as a network directory with a network drive letter on your local computer. In this example the remote path *D:\work\* is mapped to the local drive *R:\*.

The figure shows the correct specification of the paths **Local path** and **Remote path**.



Method to access remote files

● Mapped network drive ○ File synchronizer

Default 'run' directory

Local path: R:\

Remote path: D:\work

Figure 2.6. Path mapping configuration

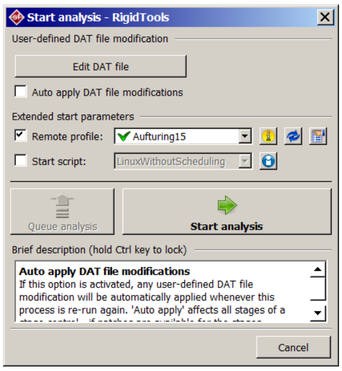
2.2.4. Starting the analysis on the server from the client

After the previous steps you should be able to start an analysis on the server with Simufact Forming with your user on the client.

1. Use one the example projects of the standard installation for the test. Start Simufact Forming, select Open example... on the File menu and select any project.

2. Click the start button ➔ and the analysis start dialog will appear.

3. Select the Remote profile in the dialog to start the analysis on the SERVER specified in the profile, and confirm by clicking the Start analysis button.



Start analysis - RigidTools

User-defined DAT file modification

□ Auto apply DAT file modifications

Extended start parameters

☑ Remote profile: Aufturing15

□ Start script: LinuxWithoutScheduling

Brief description (hold Ctrl key to lock)

Auto apply DAT file modifications
If this option is activated, any user-defined DAT file modification will be automatically applied whenever this process is re-run again. 'Auto apply' affects all stages of a

Figure 2.7. Starting the analysis on the SERVER from the CLIENT

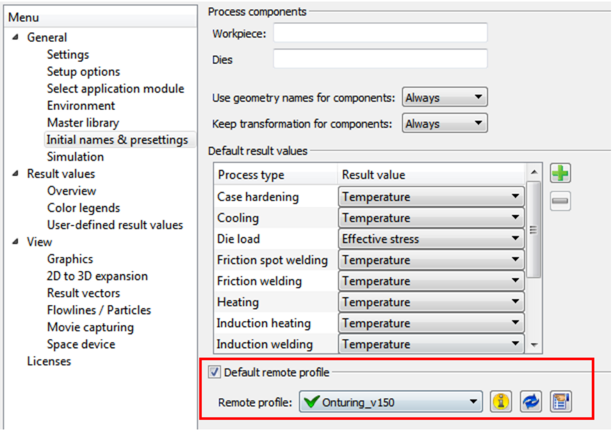
When you start the analysis, you will be able to select the Remote profile created in Section 2.2.3 in the dialog Figure 2.7. If the Remote profile does not show a ✓ but a ●, press the button ⟳ to refresh. If the Remote profile shows a ⊘ see the **Brief description** at the bottom of the dialog for possible reasons why this Remote profile can not be used.

If you are start the analysis the following will happen automatically:

- In the working directory on the server a new run directory will be created for this simulation.
- The input data are moved from the CLIENT (local run directory) to the new run directory (SERVER).
- The CLIENT sends a message through the network to the SERVER. This message tells the SERVER to start the analysis in the new run directory on the SERVER.
- The Simufact Forming GUI follows the analysis status on the CLIENT as in local analyses, the progress bar is updated, available results are imported automatically, and you can check the results, stop the analysis etc.

Default remote profile

For each process Simufact Forming stores the last used remote profile. To avoid the need to manually select the remote profile for each new process, a default remote profile can be defined in the **general settings** under **Initial names & presettings**. When a simulation is started the model check of Simufact Forming compares the settings of for parallel computing with the available licenses and issues a warning if more licenses are required than available. If defined - and not overruled by a remote profile stored in the process - the default remote profile is used for this comparison. This avoids unnecessary warnings in certain license configurations.



Process components

Workpiece:

Dies

Use geometry names for components: Always

Keep transformation for components: Always

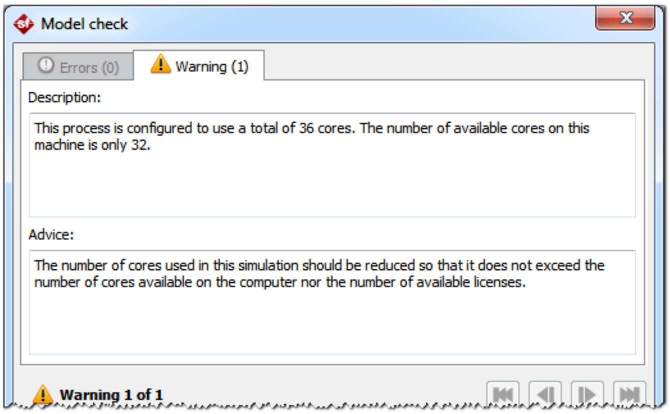
Default result values

Process type	Result value
Case hardening	Temperature
Cooling	Temperature
Die load	Effective stress
Friction spot welding	Temperature
Friction welding	Temperature
Heating	Temperature
Induction heating	Temperature
Induction welding	Temperature

☑ Default remote profile

Remote profile: ✓ Onturing_v150

Figure 2.8. Configuration of a default remote profile



Description:

This process is configured to use a total of 36 cores. The number of available cores on this machine is only 32.

Advice:

The number of cores used in this simulation should be reduced so that it does not exceed the number of cores available on the computer nor the number of available licenses.

Figure 2.9. Configuration of a default remote profile

2.3. Queueing

If no more licenses are available, the simulations will not start or continue but reach the status "Waiting for License", see Section 4.1.2. This may be used to implement a simple, short-term quasi batch mode in combination with Simufact Remote and in all situations where licenses are shared between users and 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. The maximum waiting time before a simulation aborts with license failure is controlled by the environment variable MSC_LICENSE_SF_WAITINGTIME (in minutes). For the configuration of environment variables see Appendix C.

The queueing functionality of the Simufact Monitor can not be used in combination with Simufact Remote.

3 Installation and configuration of compilers

Simufact Forming 2022.1 supports user subroutines written in the FORTRAN programming language.

3.1. 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 Forming. You have the choice between 2 different compilers:

- The Intel® Parallel Studio XE Composer for Fortran.

  This compiler is used during development development and build of Simufact Forming. 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 Forming 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 Forming 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\forming\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.

3.2. Intel Parallel Studio XE for Fortran

3.2.1. Requirements

To use user subroutines with Simufact Forming 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 3.1.

Table 3.1. Compiler required for user subroutines in Simufact Forming 2022.1

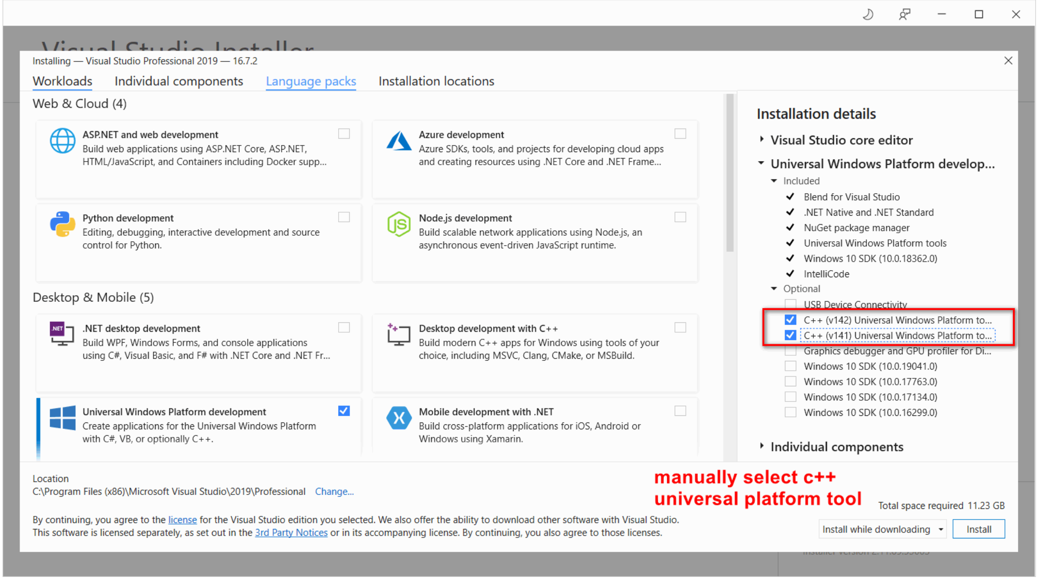
Intel Parallel Studio XE 2020	Windows™ version / build	Linux™ version / build
Update 4	**19.1.3.311** (20200925)	**19.1.3.304** (20200925)

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

3.2.2. 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 Forming 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 3.2.2.1.1.

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



Visual Studio Installer

Installing — Visual Studio Professional 2019 — 16.7.2

Web & Cloud (4)

- □ ASP.NET and web development — Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...
- □ Azure development — Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET Core and .NET Frame...
- □ Python development — Editing, debugging, interactive development and source control for Python.
- □ Node.js development — Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

- □ .NET desktop development — Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET Core and .NET Fr...
- □ Desktop development with C++ — Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.
- ☑ Universal Windows Platform development — Create applications for the Universal Windows Platform with C#, VB, or optionally C++.
- □ Mobile development with .NET — Build cross-platform applications for iOS, Android or Windows using Xamarin.

Installation details

- Visual Studio core editor
- Universal Windows Platform develop...
  - Included
    - ✓ Blend for Visual Studio
    - ✓ .NET Native and .NET Standard
    - ✓ NuGet package manager
    - ✓ Universal Windows Platform tools
    - ✓ Windows 10 SDK (10.0.18362.0)
    - ✓ IntelliCode
  - Optional
    - □ USB Device Connectivity
    - ☑ C++ (v142) Universal Windows Platform to...
    - ☑ C++ (v141) Universal Windows Platform to...
    - □ Graphics debugger and GPU profiler for Di...
    - □ Windows 10 SDK (10.0.19041.0)
    - □ Windows 10 SDK (10.0.17763.0)
    - □ Windows 10 SDK (10.0.17134.0)
    - □ Windows 10 SDK (10.0.16299.0)
- Individual components

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional Change...

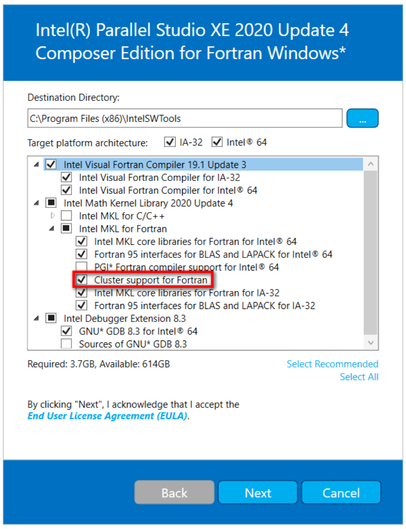
manually select c++ universal platform tool

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 11.23 GB

Figure 3.1. 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.



Intel(R) Parallel Studio XE 2020 Update 4
Composer Edition for Fortran Windows*

Destination Directory:
C:\Program Files (x86)\IntelSWTools

Target platform architecture: ☑ IA-32 ☑ Intel® 64

- ☑ Intel Visual Fortran Compiler 19.1 Update 3
  - ☑ Intel Visual Fortran Compiler for IA-32
  - ☑ Intel Visual Fortran Compiler for Intel® 64
- ■ Intel Math Kernel Library 2020 Update 4
  - □ Intel MKL for C/C++
  - ■ Intel MKL for Fortran
    - ☑ Intel MKL core libraries for Fortran for Intel® 64
    - ☑ Fortran 95 interfaces for BLAS and LAPACK for Intel® 64
    - □ PGI* Fortran compiler support for Intel® 64
    - ☑ Cluster support for Fortran
    - ☑ Intel MKL core libraries for Fortran for IA-32
    - ☑ Fortran 95 interfaces for BLAS and LAPACK for IA-32
- ■ Intel Debugger Extension 8.3
  - ☑ GNU* GDB 8.3 for Intel® 64
  - □ Sources of GNU* GDB 8.3

Required: 3.7GB, Available: 614GB

By clicking "Next", I acknowledge that I accept the *End User License Agreement (EULA)*.

Figure 3.2. 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 3.2.2.1.1. 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.

3.2.2.1. 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 Forming

   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\forming\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. VS2019INSTALLDIR for Visual Studio 2019.

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

3.2.2.1.1. Compiler not found

Intel® ifort Fortran compiler is installed however Simufact Forming 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 Forming to be able to find your installation of the compiler, you may have to edit the include-file

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

on Windows resp.

```
.../simufact/forming/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 arguments 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.

3.2.2.1.2. 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.

3.2.2.1.3. 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 3.2.2.1.1.
- 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.

3.3. 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 Forming 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\forming\2022.1\sfMarc\tools\include_win64.bat
```

needs to be modified for the correct installation path:

```
rem try oneAPI
:compiler_oneAPI
if not exist "C:\Program Files (x86)\Intel\oneAPI\setvars.bat" goto
 compiler_done
call "C:\Program Files (x86)\Intel\oneAPI\setvars.bat" intel64 vs2019 > NUL
set COMPILER=
for %%i in (ifort.exe) do set COMPILER=%%~$PATH:i
if "%COMPILER%"=="" goto compiler_done
if %RELEASE%==1 set MARC_MKL=C:\Program Files (x86)\Intel\oneAPI\mkl\latest
\lib\intel64
set INTELPATH=C:\Program Files (x86)\Intel\oneAPI\compiler\latest\windows
\compiler
if not "%COMPILER%"=="" goto compiler_done
```

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 form 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 3.2.2.1 above, too.

4 Troubleshooting

A list of known issues of this version of Simufact Forming 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 Forming.

4.1. Troubleshooting on Windows™

4.1.1. General hints

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

As stated in the Release Notes, Simufact Forming 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 sfForming.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 Forming 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.

4.1.1.2. 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 Forming navigate to **General** - **Message / dialog positions** and press Reset all. Try again afterwards.

- Close all GUIs of Simufact Forming and delete `view.ini` and `forming_windows.ini` from `C:\Users\<username>\AppData\Roaming\simufact\simufact.forming_2022.1` (`%appdata%`). Open the GUI again and try again afterwards. Regarding ini files compare Section 4.1.1.5 below.

4.1.1.3. 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 be 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.

4.1.1.4. 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.

4.1.1.5. INI files

INI files are created for Windows™ in

- `C:\Users\<username>\AppData\Local\Simufact` (`%localappdata%`),

- `C:\Users\<username>\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 Forming 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 Forming. Thus, if you consider re-installing Simufact Forming to fix some issues, consider deleting the INI-files, too.

4.1.1.6. 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.

4.1.1.7. 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 Forming, in particular of the solvers. If you experience such problems, deactivate your antivirus software.

4.1.1.8. 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.

4.1.2. Status "Waiting for License"

If a license can not be granted Simufact Forming 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.

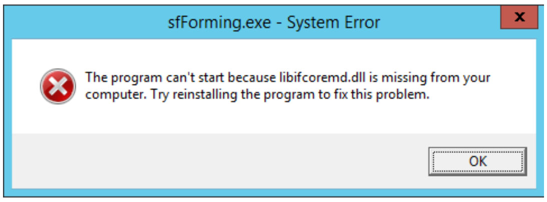
In a single user, single workstation environment the Simufact Monitor offers a simple but flexible batch queue for simulations, which is - in this environment - a better choice than using the status "Wait for license" as quasi batch mode.

4.1.3. 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 Forming. 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.


sfForming.exe - System Error

The program can't start because libifcoremd.dll is missing from your computer. Try reinstalling the program to fix this problem.

Figure 4.1. Error due to not found Fortran Redistributables

Figure 4.1 shows a typical error message occurring when Simufact Forming 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 redistributalbes 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 Forming, will *not* help. See Appendix C for the review and configuration of environment variables.

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

Some times 3rd-party applications install their DLLs in the system directories, but this DLLs conflict with DLLs Simufact Forming 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`.

4.1.4. 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™.

4.1.5. 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 Forming 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 4.1.1.2 above.

- Make sure that the target `"...\simufact\forming\2022.1\sfForming\bin\sfForming.exe"` of the shortcut to launch the program is correct.

- Open a command prompt, navigate to the installation directory and start `...\simufact\forming\2022.1\sfForming\bin\sfForming.exe` from within the command prompt. Watch for errors.

- Check in the Windows™ Taskmanager whether `sfForming.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.

4.1.6. 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 Forming.
- CAD import is done using a temporary directory in `%TEMP%\simufact.forming`. 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\<username>\AppData\Roaming\Simufact\simufact.forming_2022.1\simufact.forming.ini`.
- Execute the redistributable packages setup in the folders `...\3rd-party` and `...\3rd-party\Microsoft Visual C++ ...` of the Simufact Forming 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.

4.1.7. 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 Forming.
- Initial meshing is done using a temporary directory in `%TEMP%\simufact.forming`. 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\<username>\AppData\Roaming\Simufact\simufact.forming_2022.1\simufact.forming.ini`.

4.1.8. 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 Forming 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 Forming 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 Forming simulations stored on a network drive using "pseudo remote" to your own PC.

4.1.9. 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 fist 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.

4.1.10. 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.

4.1.11. Remote problems

If you change your license configuration, you need to stop and start again the Simufact Remote service ("Local server") afterwards (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.

4.1.11.1. Display of a remote server or a remote profile in the Remote Manger on the CLIENT

- Cannot access the remote server.
  - The remote server may be busy. Try again later.
- A remote server is not displayed.
  - Try clicking Update.

    The remote server might be busy and the time limit for the response time may be too short. Try again later. Check on the SERVER whether the Local server is running. Check your firewall.
- The list of programs on the remote server is empty.
  - The service of the remote computer could not read the file `simufact.softwares.ini` or the file contains wrong entries.

    For the location of this INI-file see Section 4.1.1.5 (Windows™) and Section 4.2.1 (Linux™).

    (Windows™ only) On the SERVER start Simufact Forming as Administrator, go to Extras / Options / Settings / General / Setup options and activate Registration for Simufact Remote. If this option is already activated, deactivate it and activate it again after closing Simufact Forming. Restart the Local server in the Simufact Remote Manager after this.

4.1.11.2. Job fails to start on the SERVER

If you struggle using Simufact Remote, try to run simulations locally on the SERVER first. To check the functionality of Simufact Forming on the SERVER for the user used on the SERVER to run simulations submitted by Simufact Remote, a step-by-step debugging should be done as follows:

a. Login the SERVER as the mentioned user and open Simufact Forming. If the GUI does not appear after launching the program by double-clicking the icon, go to Section 4.1.5.

b. Check under Extras / Options / Settings / Licenses if all required licenses are available and valid. If some of them fail, check the hints given in this installation guide, especially in Appendix A, if needed check the documentation on MSC Licensing.

c. Run an FE job without DDM (e.g. `...\simufact\forming\2022.1\sfForming\examples\cold_forming\upsetting\cylinder\cylinder.sfp`). If the job fails to run, analyse the reason, fix it and try again.

d. Run an FV job (e.g. `...\simufact\forming\2022.1\sfForming\examples\hot_forging\quickstart\quickstart-FV\Quickstart-FV.sfp`).

e. If you have licenses for parallel computing, run an FE job with DDM (e.g. `...\simufact\forming\2022.1\sfForming\examples\cold_forming\quickstart\quickstart-FE\Quickstart-FE.sfp`), and set the Number of domains in the Forming Control under Parallelization to 2 or more. If the job fails to run, analyse the reason, fix it and try again.

f. If necessary, run an FE job with a subroutine (e.g. `...\simufact\forming\2022.1\sfForming\examples\scientific\friction\user-friction.sfp`).

Once the simulations work locally on the SERVER you can start to use Simufact Remote to start simulations from a CLIENT on the SERVER. If these simulations fail, possible reasons are:

1. Check the rights of the working directory on the SERVER (by default the run folder will be created in `C:\Windows\Temp\` for Windows™). The user that has installed the program and runs the remote service should have full accessibility to this folder.

2. If the file synchroniser is used: check the synchronization process in `sync.log` in the `_Run_` folder on the local machine. You can find more information about the synchronization in `sync.ini` in the same folder.

3. If a mapped network drive is used: make sure that you have inserted correct local and remote paths. Make sure that it is accessible from the CLIENT and on the SEVER. On the SERVER it needs to be accessible by the user that runs the remote service.

4. Make sure if the contents of the `_Run_` directory on the CLIENT (right-clicking any started process and then **Open process folder** will open the local `_Run_` directory of this process) is completely or partially copied into the run directory (in the working directory) on the SERVER.

5. The input files for the analysis were copied to the SERVER but the analysis does not run:

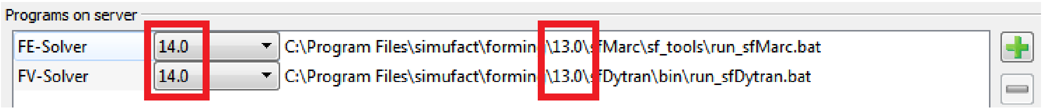
   - If you have entered a user and his password into the used remote profile: Please check if you have entered the correct password. Does your operating system support multiple user logins at the same time? Linux™ does!

   - If the simulation does not start on the SERVER or crashes later: Check the messages in the log- and out-files in the run directory on the SERVER and the CLIENT. Check for license failures. Make sure that the remote service is still running.

   - If the simulations start on the SERVER but gets stuck later: Make sure that the remote service is still running. Check the running processes on the SERVER whether `marc.exe` for FE jobs or `sfDytran.exe` for FV jobs is still running or not using the Windows Task Manager. Due to network instability, some times the job is still running on the SERVER but the files and the simulation status in the run directory on the SERVER are not copied to the CLIENT. As a workaround, it suffices to copy the files manually.

4.1.11.3. The Remote Manager assigns an old solver path for the new version

After the installation of a new version, the Remote Manager may assign old solver path for the new version:



Programs on server

FE-Solver	14.0	C:\Program Files\simufact\formin\13.0\sfMarc\sf_tools\run_sfMarc.bat
FV-Solver	14.0	C:\Program Files\simufact\formin\13.0\sfDytran\bin\run_sfDytran.bat

Figure 4.2. Wrong solver path for newly installed version

- On the CLIENT in the Remote Manager, delete the remote server. Close the Remote Manager and add the remote server again.
- On the CLIENT rename or delete the file `C:\Users\<USERNAME>\AppData\Roaming\Simufact\simufact.remote\remote.ini`. Consider the INI files in `C:\ProgramData\Simufact\`, too, and repeat the step above.
- On the SERVER start Simufact Forming as Administrator, go to Extras / Options / Settings / General / Setup options and activate Registration for Simufact Remote. If this option is already activated, deactivate it and activate it again after closing Simufact Forming. Restart the Local server in the Simufact Remote Manager after this. Repeat the steps above on the CLIENT.
- On the SERVER rename or delete all `simufact.software.ini` files in all locations indicated in Section 4.1.1.5. Now, repeat the step above on the SERVER and after this the steps above on the CLIENT. *Note*: You will have to register all versions of Simufact Forming you want to use for Simufact Remote separately.
- On the SERVER check and correct the content of the `simufact.software.ini` files. Typically delete all but the one in `ProgramData`.

4.1.11.4. Instable file transfer if the file file synchroniser is used

In case you are using the file synchroniser for the transfer of the files between CLIENT and SEVER and you experience slow or instable file transfer, you may try whether using a mapped network drive improves the behaviour. And for sure the other way round. In case the file synchroniser is used, you can try to stabilise the file transfer by adding or modifying the following settings in the `remote.ini` on the SERVER and the `simufact.forming.ini` in the section `[sfRemote]` on the CLIENT:

- `FileTransferTimeout=60000`
- `FileTransferChunkSize=1600000`

The `FileTransferTimeout` is the maximum permissible time in milliseconds to transfer a data block of `FileTransferChunkSize` in bytes. A smaller `FileTransferChunkSize` can stabilise the data transfer, but will add more overhead to the transfer. Remember to restart the Simufact Remote Sever after changing the settings.

4.1.11.5. Usage of Start Scripts

In case of unresolvable trouble with Simufact Remote, consider using Start Scripts as an alternative. See the Infosheet provided. Samples are provided in `...\simufact\forming\2022.1\sfForming\lib\scripts\samples`. The samples have to be adapted to your local environment. Start Scripts are generally more flexible than Simufact Remote, but require more configuration effort.

4.2. Troubleshooting on Linux™

Please check the troubleshooting hints for Windows™, too. Some typically problems are independent form the operating system. Especially regarding Section 4.1.2, the behaviour of the Remote Manager on the Client Section 4.1.11.1, Section 4.1.11.3, regarding the file transfer Section 4.1.11.4 and the usage of Start Scripts Section 4.1.11.5 apply for Linux™ SERVERs, too. For sure the location of the INI files mention is different, compare Section 4.2.1.

4.2.1. 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. Use the configuration utility `config.exe` (compare Section 1.2.1) to recreate the settings for the Simufact Remote service. Consider using the configuration utility `config.exe` to hard code the license settings into the solver start scripts.

4.2.2. The Simufact Remote server doesn't respond anymore

If the Simufact Remote server uses 100% of one CPU core and does not respond any more, please make sure that the daemon knows where to write the logging information. Check the file `~/.config/Simufact/simufact.remote/remote.ini` in the home directory of the user running the service and the global configuration file `/etc/xdg/Simufact/simufact.remote/remote.ini` for an entry `LOGFILE` in the section `[sfRemote]`. It has to point to a location for which the user running the service has writing permissions. If you don't want this information to be logged, either set the `SERVERLOGLEVEL` to “`0`” or set “`/dev/null`” as logfile.

4.2.3. The Simufact Remote server does not start

The Simufact Remote server may not start or work due to old PID files and sockets. Irregular exits of the server may lead to these files left over, potentially causing problems.

The start-stop script of the remote server (`/etc/init.d/sfserviced`) use the PID file `/var/run/sfserviced.pid` to keep track of the running server.

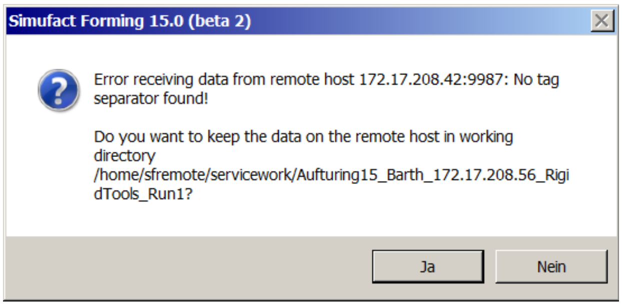
The remote server (`../simufact/system/bin/simufact.service`) executable use the socket `/var/tmp/simufactservice.<username>` for network communication.

If you experience problems with the server not starting, hanging during start, not responding or similar, ensure (using `ps`) that the relevant server process is not running. Then delete the PID file and / or the socket and try again.

Further more, for debugging you can start the Simufact Remote server in the foreground using `.../simufact/system/bin/simufact.service -e` directly. If the environment variable GLOG_logtostderr is set to `1`, all messages will be shown on stderr, thus in the terminal used to start the process if you do not redirect them.

4.2.4. The Simufact Remote server does accept simulations

If a simulation submitted to the Simufact Remote server using scheduling does not leads to the error message shown in Figure 4.3 on the Windows CLIENT, the scheduling environment needs to be configured in `schedule_job` (compare Section 1.2.5.4). Check the initialisation files of your login shell for the needed configuration or as your local administrator.


Simufact Forming 15.0 (beta 2)

Error receiving data from remote host 172.17.208.42:9987: No tag separator found!

Do you want to keep the data on the remote host in working directory /home/sfremote/servicework/Aufturing15_Barth_172.17.208.56_RigidTools_Run1?

Figure 4.3. Error due to not initialized scheduling environment

4.2.5. Jobs submitted by Simufact Remote fail to start on the SERVER

If you struggle using Simufact Remote, try to run simulations locally on the SERVER first. To check the functionality of the Simufact Forming solvers on the SERVER for the user used on the SERVER to run simulations submitted by Simufact Remote, a step-by-step debugging should be done as indicated below. The same tests should be done for the user used to run the Simufact Remote service on the SERVER.

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 `sf_testjob_fe*` from `.../simufact/forming/2022.1/sfMarc/tools/test` into the temporary directory, change into that directory.

- execute the solver start script like this:

  ```
  .../simufact/forming/2022.1/sfMarc/tools/run_marc -j sf_testjob_fe -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/forming/2022.1/sfMarc/tools/run_marc -j sf_testjob_fe -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/forming/2022.1/sfMarc/tools/run_marc -j sf_testjob_fe -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 you Fortran file without the extension. A small test user subroutine is provided in `.../simufact/`forming/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. Try to run an FV jobs manually, start with the minimum options and without parallelisation before increasing the complexity:

- copy the all files from `.../simufact/forming/2022.1/sfDytran/test` into the temporary directory, change into that directory.

- execute the solver start script like this:

  ```
  .../simufact/forming/2022.1/sfDytran/bin/sfDytran jid=sf_testjob_fv
  ```

  This will start the testjob in the background. Wait for it to finish after some minutes. Check the OUT file for errors like missing licenses. You can use a 2nd shell to monitor the testjob.

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

  ```
  .../simufact/forming/2022.1/sfDytran/bin/sfDytran jid=sf_testjob_fv
  dmp=yes ncpus=2
  ```

  to check whether the shared memory parallelisation is working. 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.

d. If you use a scheduling engine, repeat b. and c. with using scheduling. See Section 1.2.5.4 and use adapted versions of the scripts `schedule_marc.sh` and `schedule_sfdytran.sh` provided in `.../simufact/forming/2022.1/sfTools/sfScheduling/testscripts`.

If all tests to run simulations manually locally on the SERVER have been successful, you can proceed with testing Simufact Remote from a CLIENT. Again, increase the complexity step-by-step:

a. Check if the remote service on the SERVER is running or stopped. Use `ps`, do not relay on the output of `/etc/init.d/sfserverd status`.

b. Do **not** enter a user name and a password in the configuration of the Remote Profile in the Remote Manager on the CLIENT. In this case all simulations will run on the SERVER using the user that runs the Simufact Remote Service. This avoids a couple of possible errors, especially in the configuration of `sudo` and in the access rights to the files and directories used. Do not activate scheduling in the Remote Profile.

c. Now try to start a simple FE simulation without using parallelisation (e.g. `....\simufact\forming\2022.1\sfForming\examples\cold_forming\upsetting\cylinder\cylinder.sfp`). If it fails:

- Check the rights of the working directory on the SERVER (by default the run folder will be created in `/tmp` for Linux™). The user runs the service should have full accessibility to this folder.

- If the file synchroniser is used: check the synchronization process in `sync.log` in the `_Run_` folder on the local machine. You can find more information about the synchronization in `sync.ini` in the same folder.

- If a mapped network drive is used: make sure that you have inserted correct local and remote paths. Make sure that it is accessible from the CLIENT and on the SEVER. On the SERVER it needs to be accessible by the user that runs the remote service.

- Make sure if the contents of the run directory on the CLIENT (right-clicking any started process and then **Open process folder** will open the local `_Run_` directory of this process) is completely or partially copied into the run directory (in the working directory) on the SERVER.

- If the simulation does not start on the SERVER or crashes later: Check the messages in the log- and out-files in the run directory on the SERVER and the CLIENT. Check for license failures. Make sure that the remote service is still running using `ps`.

- If the simulations start on the SERVER but gets stuck later: Make sure that the remote service is still running. Check the running processes on the SERVER whether `marc` for FE jobs or `sfDytran.exe` for FV jobs is still running or not using `ps`. Due to network instability, some times the job is still running on the SERVER but the files and the simulation status in the run directory on the SERVER are not copied to the CLIENT. As a workaround, it suffices to copy the files manually.

d. Try to start a simple FV simulation without using parallelisation (e.g. `...\simufact\forming\2022.1\sfForming\examples\hot_forging\quickstart\quickstart-FV\Quickstart-FV.sfp`). If this fails, do the same checks as for FE outlined above.

e. If both works, and if you have licenses for parallel computing, you can proceed with trying a simulation using DDM (e.g. `...\simufact\forming\2022.1\sfForming\examples\cold_forming\quickstart\quickstart-FE\Quickstart-FE.sfp`), and set the Number of domains in the Forming Control under Parallelization to 2 or more. A typical problem is given in Section 4.2.6.

f. If necessary, run an FE job with a subroutine (e.g. `...\simufact\forming\2022.1\sfForming\examples\scientific\friction\user-friction.sfp`).

g. If this works you can proceed to enter a user name and password in the configuration of the Remote Profile in the Remote Manager on the CLIENT - if you wish to do so. In this case all simulations will run on the SERVER using the given user. The Simufact Remote Service internally uses `sudo` to call the solver as a different user. If the file synchroniser is used, the file transfer is still done using the user that runs the service. If a mapped network drive is used for the file transfer, the user used to map the network drive is used for the file transfer. In both cases, the user used to run the simulations has to be able to change into the temporary run directory on the SERVER (and to create it if the file synchroniser is used).

   Repeat the test given in c) to f) above in the same order. Typical sources for errors are:

   - For sure the user specified on the client needs a valid account on the SERVER and needs to specify correct login details for this account, which may be different from his account on the CLIENT. Try with different users.
   - The `sudo` configuration. Double check the hints given in Section 1.2.6.
   - The environment after `sudo` and - if used - after the scheduling engine. You can use `sudo -u user /bin/sh` to check on this. Consider the options `sudo`, the shell and your system / distribution offer for environment settings and variables. A PAM module may be involvable, if you use it. Consider hard coding all needed variables in the solver start scripts and in the scheduling scripts.
   - The access rights to files and directories. This may lead to situations where a simulation can be started from the CLIENT but no results or no simulation status can be read back from the SEVER. Or files remain on the server. Thus carefully monitor the local and the remote run directory. If the file synchroniser is used, the user that runs the service has to be able to read and delete all files written by the solver. If a mapped network drive is used, check the access to it manually from the CLIENT and from the SERVER. On the SERVER check for both, the user used to run the simulation and the user that runs the service. SE linux has special restriction and configurations for mapped network drives.

h. If you use a scheduling engine, you may now try to involve it repeating the steps above. Keep in mind to the check the configuration on all hosts of your cluster. The working directory of Simufact Remote must be shared on all hosts using the same path.

4.2.6. DDM jobs started by the Remote server hang

DDM jobs started by the Simufact Remote server 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/forming/2022.1/sfMarc/tools/include_linux64`

4.3. Log files

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

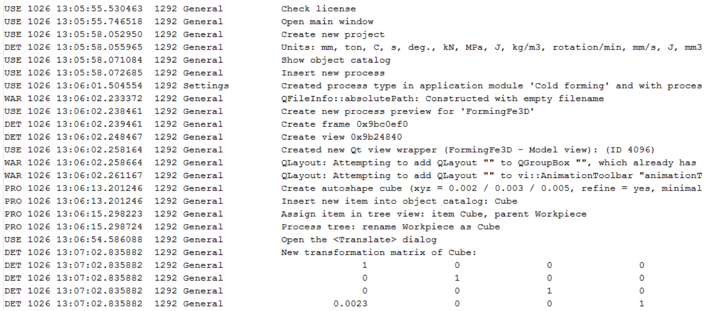
- **Solver**: The solver write extensive information in the `log`- and `out`-files in the `_Run_` directory of each simulation. These files contain logging information of the **meshers**, too. They are the first place to check about not starting or aborting simulations. They can be used to judge the simulation progress, too.
- **Simufact Forming 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_`).
- **Remote service**: When calculating on a remote computer, this keeps track of the simulation control and remote status queries.
- **File synchronizer**: When calculating on a remote computer with the "file synchronization" method, keeps track of the synchronized files.
- **Simufact Forming 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.

4.3.1. Simufact Forming 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 13:06:02.258664  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 vi::AnimationToolbar "animationT
PRO 1026 13:06:13.201246  1292 General         Create autoshape 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            1            0
DET 1026 13:07:02.835882  1292 General                0.0023            0            0            1
```

Figure 4.4. Simufact Forming GUI log file

Format:

- message type (USE = user, WAR = warning, PRO = progress, DET = details)

- date
- time (with microseconds)
- thread ID
- topic
- message

4.3.2. License client

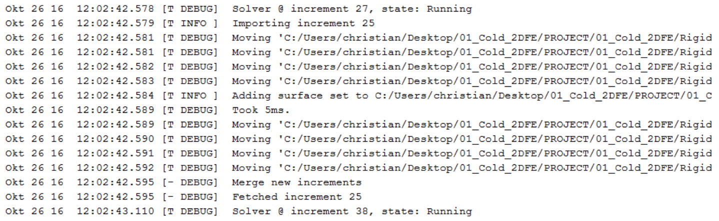
The messages of the license client get part of the messages in the log- and out-files of the solver and of the meshers. The GUI does not log this.

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

4.3.3. 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_`).



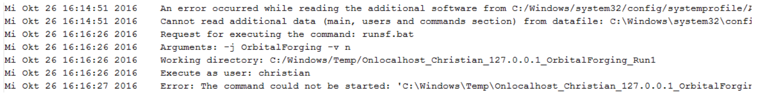
```
Okt 26 16  12:02:42.578 [T DEBUG]  Solver @ increment 27, state: Running
Okt 26 16  12:02:42.579 [T INFO ]  Importing increment 25
Okt 26 16  12:02:42.581 [T DEBUG]  Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16  12:02:42.581 [T DEBUG]  Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16  12:02:42.582 [T DEBUG]  Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16  12:02:42.583 [T DEBUG]  Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16  12:02:42.584 [T INFO ]  Adding surface set to C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_C
Okt 26 16  12:02:42.589 [T DEBUG]  Took 5ms.
Okt 26 16  12:02:42.589 [T DEBUG]  Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16  12:02:42.590 [T DEBUG]  Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16  12:02:42.591 [T DEBUG]  Moving 'C:/Users/christian/Desktop/01_Cold_2DFE/PROJECT/01_Cold_2DFE/Rigid
Okt 26 16  12:02:42.592 [T 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 [T DEBUG]  Solver @ increment 38, state: Running
```

Figure 4.5. Result manager log file

4.3.4. Remote service

Path: As specified in the Simufact Remote Manager resp. in `.../Simufact/simufact.remote/remote.ini`. Additionally log files are written to `%localappdata%/Simufact/log` resp. if running normally as user SYSTEM to `%WINDIR%\System32\Config\SystemProfile\AppData\Local\simufact\log`.

Contents: When calculating on a remote SERVER, this keeps track of the simulation control and remote status queries on the SERVER.



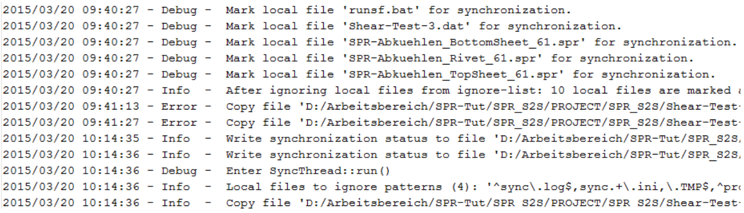
```
Mi Okt 26 16:14:51 2016    An error occurred while reading the additional software from C:/Windows/system32/config/systemprofile/A
Mi Okt 26 16:14:51 2016    Cannot read additional data (main, users and commands section) from datafile: C:\Windows\system32\confi
Mi Okt 26 16:16:26 2016    Request for executing the command: runsf.bat
Mi Okt 26 16:16:26 2016    Arguments: -j OrbitalForging -v n
Mi Okt 26 16:16:26 2016    Working directory: C:/Windows/Temp/Onlocalhost_Christian_127.0.0.1_OrbitalForging_Run1
Mi Okt 26 16:16:26 2016    Execute as user: christian
Mi Okt 26 16:16:27 2016    Error: The command could not be started: 'C:\Windows\Temp\Onlocalhost_Christian_127.0.0.1_OrbitalForgir
```

Figure 4.6. Remote service log file

4.3.5. File synchronizer

Path: `<process directory>/_Run_/sync.log`

Contents: When calculating on a remote computer with the "file synchronization" method, this keeps track of the synchronized files.



```
2015/03/20 09:40:27 - Debug -  Mark local file 'runsf.bat' for synchronization.
2015/03/20 09:40:27 - Debug -  Mark local file 'Shear-Test-3.dat' for synchronization.
2015/03/20 09:40:27 - Debug -  Mark local file 'SPR-Abkuehlen_BottomSheet_61.spr' for synchronization.
2015/03/20 09:40:27 - Debug -  Mark local file 'SPR-Abkuehlen_Rivet_61.spr' for synchronization.
2015/03/20 09:40:27 - Debug -  Mark local file 'SPR-Abkuehlen_TopSheet_61.spr' for synchronization.
2015/03/20 09:40:27 - Info  -  After ignoring local files from ignore-list: 10 local files are marked 
2015/03/20 09:41:13 - Error -  Copy file 'D:/Arbeitsbereich/SPR-Tut/SPR_S2S/PROJECT/SPR_S2S/Shear-Test-
2015/03/20 09:41:27 - Error -  Copy file 'D:/Arbeitsbereich/SPR-Tut/SPR_S2S/PROJECT/SPR_S2S/Shear-Test-
2015/03/20 10:14:35 - Info  -  Write synchronization status to file 'D:/Arbeitsbereich/SPR-Tut/SPR_S2S/
2015/03/20 10:14:36 - Info  -  Write synchronization status to file 'D:/Arbeitsbereich/SPR-Tut/SPR_S2S/
2015/03/20 10:14:36 - Debug -  Enter SyncThread::run()
2015/03/20 10:14:36 - Info  -  Local files to ignore patterns (4): '^sync\.log$,sync.+\.ini,\.TMP$,^pr
2015/03/20 10:14:36 - Info  -  Copy file 'D:/Arbeitsbereich/SPR-Tut/SPR S2S/PROJECT/SPR S2S/Shear-Test-
```

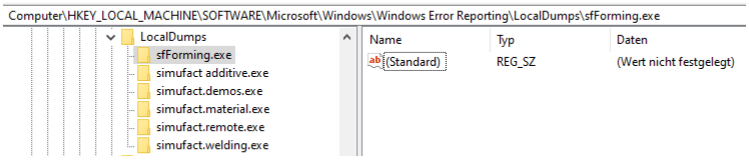
Figure 4.7. File synchronizer log file

4.3.6. Simufact Forming memory dumps

Path: `%LOCALAPPDATA%\CrashDumps` (the standard Windows™ location), may differ if old versions have been installed before. The simufact service, used by Simufact Remote only, writes its crash dumps into the directory `%WINDIR%\System32\Config\SystemProfile\AppData\Local\CrashDumps` when started normally as user SYSTEM.

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.



Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps\sfForming.exe

- LocalDumps
  - sfForming.exe
  - simufact additive.exe
  - simufact.demos.exe
  - simufact.material.exe
  - simufact.remote.exe
  - simufact.welding.exe

Name	Typ	Daten
(Standard)	REG_SZ	(Wert nicht festgelegt)

Figure 4.8. Window Registry keys needed for memory dumps

4.4. 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 Forming menu you can find an Infosheet that describes how to contact the Simufact support and what information needs to be provided to enable a 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 indented 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_FLIE` (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 variable 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 form 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 recheckout 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 4.1.2.
- 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 and 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 an other. 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.

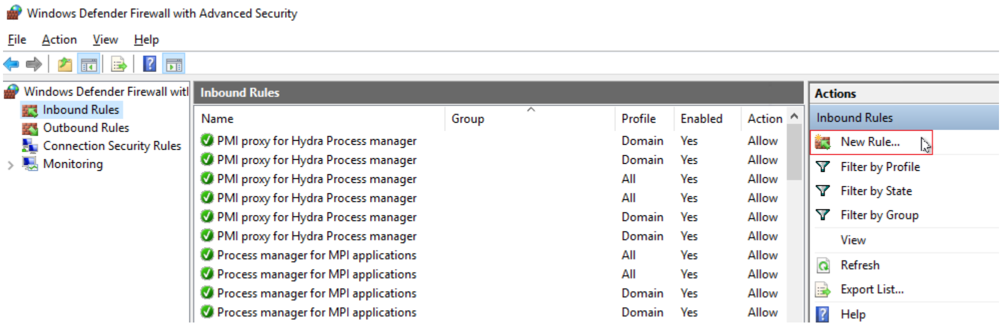
- Network licenses can be served by redundant license servers where one out of three may be offline. A special license file is needed for this. If needed, please ask your local Simufact sells representative about this. Please note: While new license queries will be very stable in this configuration, 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.

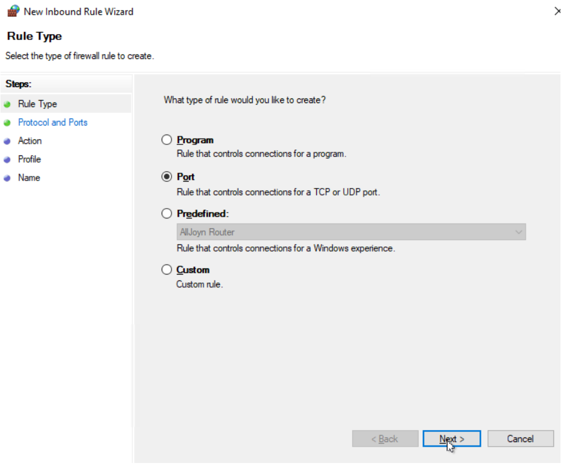


Inbound Rules

Name	Group	Profile	Enabled	Action
PMI proxy for Hydra Process manager		Domain	Yes	Allow
PMI proxy for Hydra Process manager		Domain	Yes	Allow
PMI proxy for Hydra Process manager		All	Yes	Allow
PMI proxy for Hydra Process manager		All	Yes	Allow
PMI proxy for Hydra Process manager		Domain	Yes	Allow
PMI proxy for Hydra Process manager		Domain	Yes	Allow
Process manager for MPI applications		All	Yes	Allow
Process manager for MPI applications		All	Yes	Allow
Process manager for MPI applications		Domain	Yes	Allow
Process manager for MPI applications		Domain	Yes	Allow

Figure B.1. New firewall rules for incoming messages

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



Rule Type

Select the type of firewall rule to create.

What type of rule would you like to create?

○ Program
Rule that controls connections for a program.

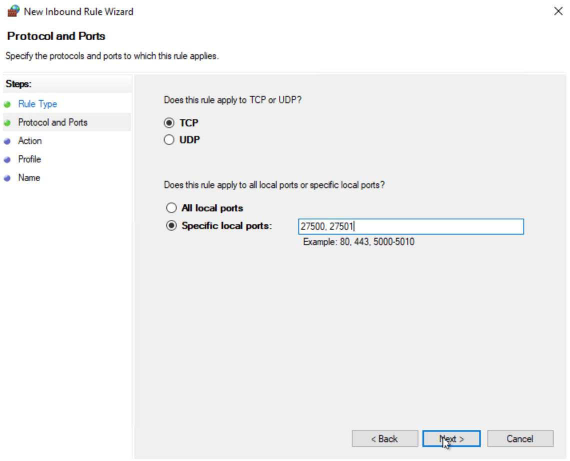
● Port
Rule that controls connections for a TCP or UDP port.

○ Predefined:
AllJoyn Router
Rule that controls connections for a Windows experience.

○ Custom
Custom rule.

Figure B.2. New firewall rule: rule type

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



New Inbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

Does this rule apply to TCP or UDP?

● TCP
○ UDP

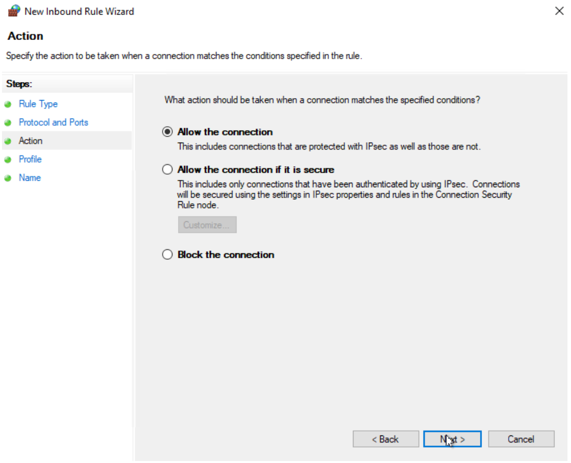
Does this rule apply to all local ports or specific local ports?

○ All local ports
● Specific local ports: 27500, 27501

Example: 80, 443, 5000-5010

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.



New Inbound Rule Wizard

Action

Specify the action to be taken when a connection matches the conditions specified in the rule.

What action should be taken when a connection matches the specified conditions?

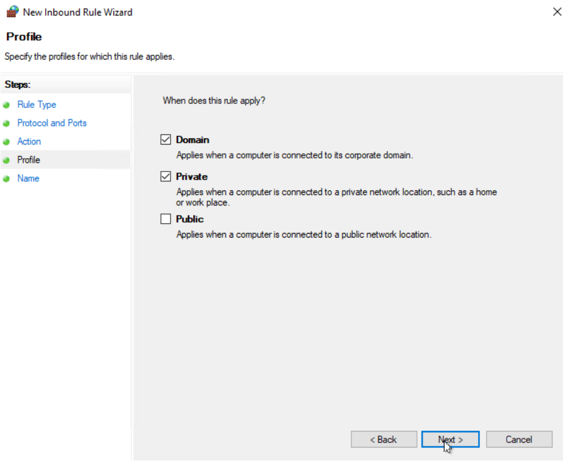
● Allow the connection
This includes connections that are protected with IPsec as well as those are not.

○ Allow the connection if it is secure
This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.

○ Block the connection

Figure B.4. New firewall rule: action

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



New Inbound Rule Wizard

Profile

Specify the profiles for which this rule applies.

When does this rule apply?

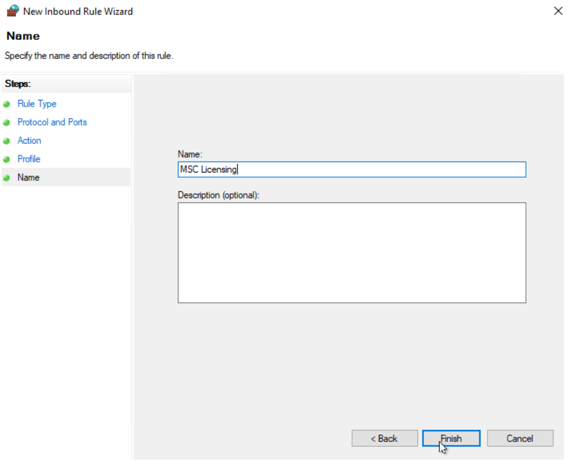
☑ **Domain**
Applies when a computer is connected to its corporate domain.

☑ **Private**
Applies when a computer is connected to a private network location, such as a home or work place.

□ **Public**
Applies when a computer is connected to a public network location.

Figure B.5. New firewall rule: profile

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



New Inbound Rule Wizard

Name

Specify the name and description of this rule.

Name: MSC Licensing

Description (optional):

Figure B.6. New firewall rule: name

7. 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 Forming 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 in 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:

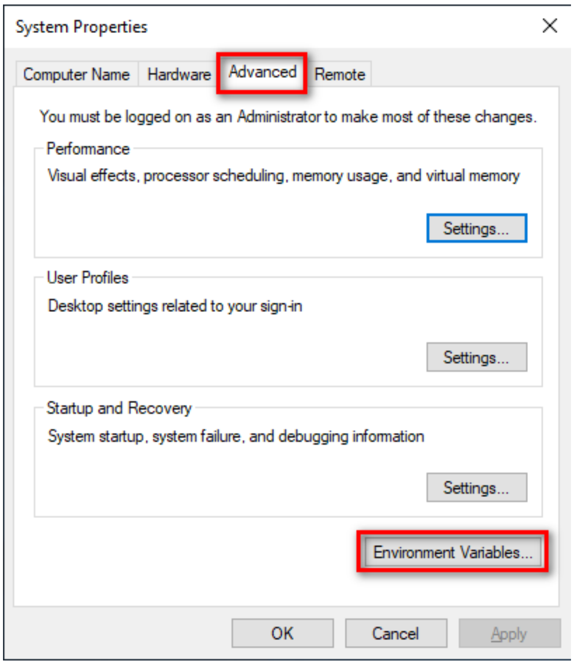
- SF_LANGUAGE
- MSC_LICENSE_FILE
- MSC_LICENSE_SF_WAITINGTIME

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.



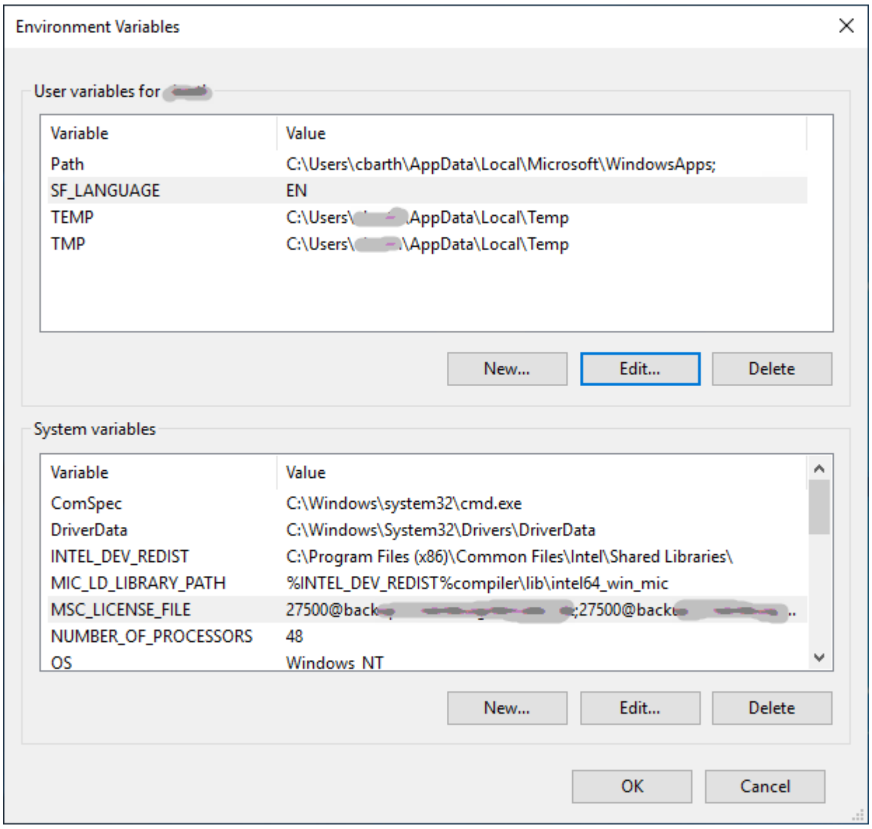
You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your sign-in

System startup, system failure, and debugging information

Figure C.1. System Properties



User variables for [redacted]

Variable	Value
Path	C:\Users\cbarth\AppData\Local\Microsoft\WindowsApps;
SF_LANGUAGE	EN
TEMP	C:\Users\[redacted]\AppData\Local\Temp
TMP	C:\Users\[redacted]\AppData\Local\Temp

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
INTEL_DEV_REDIST	C:\Program Files (x86)\Common Files\Intel\Shared Libraries\
MIC_LD_LIBRARY_PATH	%INTEL_DEV_REDIST%compiler\lib\intel64_win_mic
MSC_LICENSE_FILE	27500@back[redacted];27500@back[redacted]..
NUMBER_OF_PROCESSORS	48
OS	Windows_NT

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

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.