Digimat 2023.1

Installation and Operations Guide

HEXAGON

Americas
5161 California Ave. Suite 200
University Research Park
Irvine, CA 92617
Telephone: (714) 540-8900
Email: americas.contact@hexagon.com

Europe, Middle East, Africa
Am Moosfeld 13
81829 Munich, Germany
Telephone: (49) 89 431 98 70
Email: info.europe@hexagon.com

Japan
KANDA SQUARE 16F
2-2-1 Kanda Nishikicho, Chiyoda-ku
1-Chome, Shinjuku-Ku
Tokyo 101-0054, Japan
Telephone: (81)(3) 6275 0870
Email: MSCJ.Market@hexagon.com

Asia-Pacific
100 Beach Road
#16-05 Shaw Tower
Singapore 189702
Telephone: 65-6272-0082
Email: APAC.Contact@hexagon.com

Worldwide Web
www.mscsoftware.com, www.hexagon.com

Support
https://simcompanion.hexagon.com

Disclaimer

March 23, 2023

DIGI:V2023.1:Z:Z:Z:DC-IOG-PDF

Documentation Feedback

At Hexagon, we strive to produce the highest quality documentation and welcome your feedback. If you have comments or suggestions about our documentation, write to us.

Please include the following information with your feedback:

- Document name
- Release/Version number
- Chapter/Section name
- Topic title (for Online Help)
- Brief description of the content (for example, incomplete/incorrect information, grammatical errors, information that requires clarification or more details and so on).
- Your suggestions for correcting/improving documentation

You may also provide your feedback about Hexagon documentation by taking a short 5-minute survey.

Note:	The above mentioned e-mail address is only for providing documentation specific feedback. If you have any technical problems, issues, or queries, please contact Technical Support.

Contents

Digimat Installation and Operations Guide

3 Installing Digimat Examples Manual

4 CAE Interfaces

5 Supported Platforms

6 Windows Prerequisites

7 Known Limitations

Preface

- About This Guide
- Purpose of This Guide
- Contents of This Guide
- Typographical Conventions
- Technical Support
- Accessing Digimat Documentation
- Internet Resources

About This Guide

This Guide *Digimat Installation and Operations Guide* contains information about installing and running license manager and its components.

Purpose of This Guide

This guide explains the procedure for installing Digimat and its components. It also describes how to install the associated documentation. This purpose of this guide is to:

- Help you install Digimat on Windows and Linux platforms.
- Help you install the licensing server on Windows and Linux platforms.
- Identify and ensure that the installation is successful.
- Provide you with basic troubleshooting.
- Provide you information about files, directories, and their location in the installed folders.

Contents of This Guide

The principal categories of information are found under the following titles:

Chapter	Title	Description
Chapter 1	Installing Digimat Licensing	Information about Digimat licensing system and how to set-up is provided in this chapter.
Chapter 2	Installing Digimat	Digimat installation is demonstrated in this chapter.
Chapter 3	Installing Digimat Examples Manual	Structure of `settings.ini` file and environment variables are discussed in this chapter
Chapter 4	CAE Interfaces	Procedure of setting up Digimat with various third party products is described in this chapter.
Chapter 5	Supported Platforms	Information regarding supported software is provided in this chapter
Chapter 6	Windows Prerequisites	Installation prerequisites are described in this chapter
Chapter 7	Known Limitations	Various limitations involved in Digimat are explained.

The information in this manual is both descriptive and theoretical. You will find some techniques discussed in detail. You will also find specific instructions for operating the various options offered by Digimat.

Typographical Conventions

This section provides a brief overview of the typographical conventions used in the document to help the user better follow the Digimat documentation.

This section describes some syntax that will help you in understanding text in the various chapters and thus in facilitating your learning process. It contains stylistic conventions to denote user action, to emphasize particular aspects of Digimat to signal other differences within the text.

LM Sans 10	Body and general text
Courier New	▪ Represents command-line options of Digimat. ▪ Directory names and paths ▪ File names and Paths ▪ Linux terminal script `Example: lmreread -c<parent>/msc/MSC.Licensing/licenses/license.dat`
Bold Text	▪ Highlights ▪ Dialog box names ▪ Buttons ▪ Menus ▪ User inputs ▪ The commands/user inputs for all descriptions related to terminal commands. ▪ Default values Example: `[root@vm-tmrhel73 MSC]#` **./msc_licensing_helium_linux64.bin**
HelveticaNeue LT Pro Cn 57	▪ Hyperlinks ▪ Weblinks Example: Chapter 2: Installing Digimat
Italic Text	Represents references to books. Example: *Digimat MF User's Guide*

Technical Support

For technical support phone numbers and contact information, please visit:
https://simcompanion.hexagon.com/customers/s/article/support-contact-information-kb8019304

Support Center

https://simcompanion.hexagon.com

The Support Center provides technical articles, frequently asked questions, and documentation from a single location.

Accessing Digimat Documentation

This section describes how to access the Digimat documentation outside of Digimat. Digimat documentation is available through PDF files. The PDF files can be obtained from the following sources:

- Digimat documentation installer
- SimCompanion
- Combined documentation

The PDF documentation files are appropriate for viewing and printing with Adobe Acrobat Reader (version 10.1.4 or higher), which is available for most Windows and Linux systems. These files are identified by a .pdf suffix in their file names.

Downloading the PDF Documentation Files

You can download the PDF documentation from SimCompanion (https://simcompanion.hexagon.com).

Navigating the PDF Files

For the purpose of easier online document navigation, the PDF files contain hyperlinks in the table of contents and index. In addition, links to other guides, hyperlinks to all cross-references to chapters, sections, figures, tables, bibliography, and index entries have been applied.

To open the cross reference to other guides in a new window, you can make following changes to your Adobe Reader settings:

1. Click **Edit** -> **Preferences**.
2. Select **Documents.**
3. Un-check **Open cross-document links in same window** option.
4. Click **OK**.

Printing the PDF Files

Adobe Acrobat PDF files are provided for printing all or part of the manuals. You can select the paper size to which you are printing in Adobe Acrobat Reader by doing the following:

1. Click **File**.
2. Select the **Print**.... option. The **Print** dialog box is displayed.
3. Select **Page Setup**....
4. Choose the required paper size in the **Page Setup** menu.

The PDF files are recommended when printing long sections since the printout will have a higher quality.

If the page is too large to fit on your paper size, you can reduce it by doing the following:

1. Select the **File** -> **Print**.

2. Under **Page Scaling,** choose the **Shrink to Printable Area** option.

Internet Resources

Hexagon (www.hexagonmi.com/mscsoftware)

Hexagon corporate site with information on the latest events, products, and services for the CAD/CAE/CAM marketplace.

Hexagon Download Center (https://mscsoftware.subscribenet.com/)

1 Installing Digimat Licensing

- License Server Installation
- Client License Configuration

License Server Installation

Digimat licensing is based on the MSC license manager. It enables the following types of licensing:

- seat-based
- MSC One

Masterkey license system is not supported for Digimat products. It is recommended to use a separate lmgrd process for the installation, not a previously installed one.

For supported platforms, see section *Operating System Requirements* in *MSC Licensing User's Guide*. This guide is available on Hexagon Download Center (https://mscsoftware.subscribenet.com/).

Installation of the Licensing System

In order to set-up Digimat licensing, the executable `msc_licensing_helium_windows64_a.exe` (for Windows platforms) or `msc_licensing_helium_linux64_a.bin` available at Hexagon Download Center (https://mscsoftware.subscribenet.com/) needs to be run from the installation medium and the installation steps followed. This setup allows to

- install MSC FlexLM daemon, lmgrd license component and license management tools
- start lmgrd service

It is recommended to use the MSC Daemon only with `lmgrd` component provided by MSC license installation.

For a step-by-step standard installation of the MSC license server via the provided installer, please refer to the *MSC Licensing User's Guide*.

Starting MSC License Server under Windows

Under Windows, lmgrd component is started at the end of the installation procedure. lmgrd is installed as a Windows service name `MSC_Licensing_Helium`. Windows services are restarted automatically after a reboot.

Remark: it is required to be logged as administrator in order to install a Windows service.

Detailed procedure to define and start license server is described in *MSC Licensing User's Guide*.

Starting MSC License Server under Linux

Under Linux, lmgrd component is started at the end of the installation procedure but this starting process will be active only till the next reboot of the Linux machine. Each time the Linux machine is rebooted, the lmgrd component must be restarted manually.

Detailed procedure to define and start license server is described in *MSC Licensing User's Guide*.

Check License Server Installation

Ensuring that the license server is running can be achieved by looking at the log file of the license server. Path of log file can be found by using `lmtools.exe` utility in **Config services** toggle (see Figure 1-1. If the server did start successfully, you should see something similar to script below. If the server is running and you still have issue to start the products, check the firewall configuration of your license server, and open the right ports.

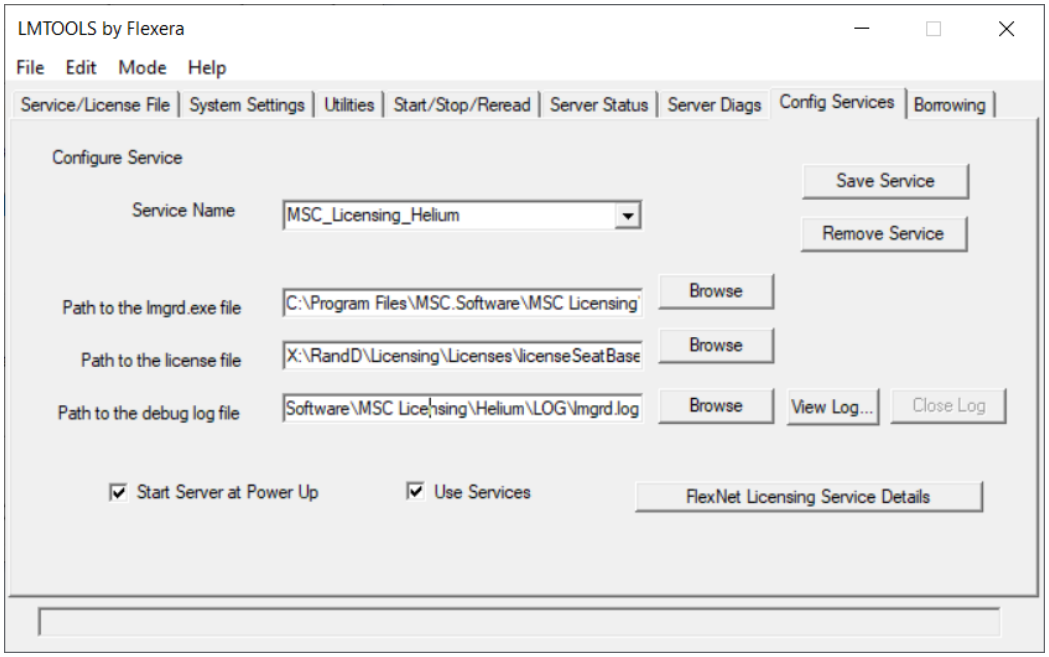


Figure 1-1 Get path to license log file.

```
9:42:53 (lmgrd) -----------------------------------------------
9:42:53 (lmgrd) Please Note:
9:42:53 (lmgrd)
9:42:53 (lmgrd) This log is intended for debug purposes only.
9:42:53 (lmgrd) In order to capture accurate license
9:42:53 (lmgrd) usage data into an organized repository,
9:42:53 (lmgrd) please enable report logging. Use Flexera's
9:42:53 (lmgrd) software license administration solution,
9:42:53 (lmgrd) FlexNet Manager, to readily gain visibility
9:42:53 (lmgrd) into license usage data and to create
9:42:53 (lmgrd) insightful reports on critical information like
9:42:53 (lmgrd) license availability and usage. FlexNet Manager
9:42:53 (lmgrd) can be fully automated to run these reports on
9:42:53 (lmgrd) schedule and can be used to track license
9:42:53 (lmgrd) servers and usage across a heterogeneous
9:42:53 (lmgrd) network of servers including Windows NT, Linux
9:42:53 (lmgrd) and UNIX.
9:42:53 (lmgrd)
9:42:53 (lmgrd) -----------------------------------------------
9:42:53 (lmgrd)
9:42:53 (lmgrd)
9:42:53 (lmgrd) Server's System Date and Time: Fri Oct 11 2019 09:42:53 Romance Daylight
Time
```

```
9:42:53 (lmgrd) pid 12356
9:42:53 (lmgrd) SLOG: Summary LOG statistics is enabled.
9:42:54 (lmgrd) Done rereading
9:42:54 (lmgrd) FlexNet Licensing (v11.16.3.0 build 246844 x64_n6) started on PPJ-PC2015
(IBM PC) (10/11/2019)
9:42:54 (lmgrd) Copyright (c) 1988-2019 Flexera. All Rights Reserved.
9:42:54 (lmgrd) World Wide Web: http://www.flexerasoftware.com
9:42:54 (lmgrd) License file(s): C:\MSC.Software\Digimat\licenseHelium.dat
9:42:54 (lmgrd) lmgrd tcp-port 27500
9:42:54 (lmgrd) (@lmgrd-SLOG@) ===============================================
9:42:54 (lmgrd) (@lmgrd-SLOG@) === LMGRD ===
9:42:54 (lmgrd) (@lmgrd-SLOG@) Start-Date: Fri Oct 11 2019 09:42:54 Romance Daylight Time
9:42:54 (lmgrd) (@lmgrd-SLOG@) PID: 12356
9:42:54 (lmgrd) (@lmgrd-SLOG@) LMGRD Version: v11.16.3.0 build 246844 x64_n6 ( build 246844
(ipv6))
9:42:54 (lmgrd) (@lmgrd-SLOG@)
9:42:54 (lmgrd) (@lmgrd-SLOG@) === Network Info ===
9:42:54 (lmgrd) (@lmgrd-SLOG@) Listening port: 27500
9:42:54 (lmgrd) (@lmgrd-SLOG@)
9:42:54 (lmgrd) (@lmgrd-SLOG@) === Startup Info ===
9:42:54 (lmgrd) (@lmgrd-SLOG@) Is LS run as a service: Yes
9:42:54 (lmgrd) (@lmgrd-SLOG@) Server Configuration: Single Server
9:42:54 (lmgrd) (@lmgrd-SLOG@) Command-line options used at LS startup: -c
C:\MSC.Software\Digimat\licenseHelium.dat
-l C:\MSC.Software\MSC Licensing\Helium\LOG\lmgrd.log -z -s local
9:42:54 (lmgrd) (@lmgrd-SLOG@) License file(s) used:
C:\MSC.Software\Digimat\licenseHelium.dat
9:42:54 (lmgrd) (@lmgrd-SLOG@) ===============================================
9:42:54 (lmgrd) SLOG: FNPLS-INTERNAL-VL1-4096
9:42:54 (lmgrd) Starting vendor daemons ...
9:42:54 (lmgrd) License server manager (lmgrd) startup failed:
9:42:54 (lmgrd) File not found, C:\Program Files\MSC.Software\MSC Licensing\Helium
9:42:54 (lmgrd) Started MSC (pid 14200)
9:42:54 (MSC) FlexNet Licensing version v11.16.3.0 build 246844 x64_n6
9:42:54 (MSC) Loading feature details 3
9:42:54 (MSC) Loading feature details 3
9:42:54 (MSC) SLOG: Summary LOG statistics is enabled.
9:42:54 (MSC) SLOG: FNPLS-INTERNAL-CKPT1
9:42:54 (MSC) SLOG: VM Status: 255
9:42:54 (MSC) SLOG: FNPLS-INTERNAL-CKPT5
9:42:54 (MSC) SLOG: TPM Status: 0
9:42:54 (MSC) SLOG: FNPLS-INTERNAL-CKPT6
9:42:54 (MSC) Server started on PPJ-PC2015 for: EX_DIGIMAT_MF
9:42:54 (MSC) EX_DIGIMAT_FE_MODELER EX_DIGIMAT_FE_SOLVER EX_DIGIMAT_VA_GUI
9:42:54 (MSC) EX_DIGIMAT_VA_SOLVER EX_DIGIMAT_MX EX_DIGIMAT_MX+
9:42:54 (MSC) EX_DIGIMAT_MAP EX_DIGIMAT_RP_GUI EX_DIGIMAT_CAE_STRUCTURAL
9:42:54 (MSC) EX_DIGIMAT_CAE_MOLDING EX_DIGIMAT_RP_MOLDEX3D EX_DIGIMAT_HC
9:42:54 (MSC) EX_DIGIMAT_AM_SLS_GUI EX_DIGIMAT_AM_STANDARD_SOLVER EX_DIGIMAT_AM_FFF_GUI
9:42:54 (MSC) EX_DIGIMAT_AM_GUI EX_DIGIMAT_AM_ADVANCED_SOLVER EX_DIGIMAT_CAE_FATIGUE
9:42:54 (MSC) EX_DIGIMAT_CAE_DRAPING EX_DIGIMAT_CAE_CASTING EX_DIGIMAT_CAE_CT
9:42:54 (MSC) EX_DIGIMAT_CAE_ADDITIVE EX_DIGIMAT_CAE_TOKEN EX_DIGIMAT_FE_SOLVER_TOKEN
9:42:54 (MSC) EX_DIGIMAT_CAE_LINEAR
9:42:54 (MSC) EXTERNAL FILTERS are OFF
9:42:54 (lmgrd) MSC using TCP-port 19980
9:42:54 (MSC) Initialization
9:42:54 (MSC) (@MSC-INFO@) === Date ===
9:42:54 (MSC) (@MSC-INFO@) 2019-10-11
9:42:54 (MSC) (@MSC-INFO@)
9:42:54 (MSC) (@MSC-INFO@) === Component Versions ===
9:42:54 (MSC) (@MSC-INFO@) MSC created executables:
9:42:54 (MSC) (@MSC-INFO@) 1) MSC.exe vendor daemon (msc.exe):
9:42:54 (MSC) (@MSC-INFO@) Flexera internal version = 11.16.3.0 x64_n6
9:42:54 (MSC) (@MSC-INFO@) MSC internal version = Helium (Build 659671)
9:42:54 (MSC) (@MSC-INFO@) 2) summarizer: Helium (Build 659671) x64_n6
9:42:54 (MSC) (@MSC-INFO@) 3) uploader: Helium (Build 659671) x64_n6
9:42:54 (MSC) (@MSC-INFO@) Flexera created executables:
9:42:54 (MSC) (@MSC-INFO@) 1) lmgrd.exe: 11.16.3.0 x64_n6
9:42:54 (MSC) (@MSC-INFO@) 2) lmtools.exe: 11.16.3.0 x64_n6
9:42:54 (MSC) (@MSC-INFO@) 3) lmutil.exe: 11.16.3.0 x64_n6
```

```
9:42:54 (MSC) (@MSC-INFO@)
9:42:54 (MSC) (@MSC-INFO@) === Usage Reporting Tool Status ===
9:42:54 (MSC) (@MSC-INFO@) Usage Reporting Activated: yes
9:42:54 (MSC) (@MSC-INFO@) Summarizer Activated: yes
9:42:54 (MSC) (@MSC-INFO@) Uploader Activated: no
9:42:54 (MSC) (@MSC-INFO@) Usage Reporting Tool Log File: C:\MSC.Software\MSC
Licensing\Helium\LOG/mscusage.mpl
9:42:54 (MSC) (@MSC-INFO@)
9:42:54 (MSC) (@MSC-INFO@) === License File Information === (Actual details pending)
9:42:54 (MSC) (@MSC-INFO@) MSC License File Reference: 6XJD
9:42:54 (MSC) (@MSC-INFO@) 6XJD: Maint Date=2019-02-21; End Date=2019-12-31:
9:42:54 (MSC) (@MSC-INFO@)
9:42:54 (MSC) (@MSC-INFO@) === Token Pool Information === (Actual details pending)
9:42:54 (MSC) (@MSC-INFO@) MSC One Tokens: Base = 1000
9:42:54 (MSC) (@MSC-INFO@)
9:42:54 (MSC) SLOG: Statistics Log Frequency is 240 minute(s).
9:42:54 (MSC) SLOG: TS update poll interval is 600 seconds.
9:42:54 (MSC) SLOG: Activation borrow reclaim percentage is 0.
9:42:54 (MSC) (@MSC-SLOG@) ================================================
9:42:54 (MSC) (@MSC-SLOG@) === Vendor Daemon ===
9:42:54 (MSC) (@MSC-SLOG@) Vendor daemon: MSC
9:42:54 (MSC) (@MSC-SLOG@) Start-Date: Fri Oct 11 2019 09:42:54 Romance Daylight Time
9:42:54 (MSC) (@MSC-SLOG@) PID: 14200
9:42:54 (MSC) (@MSC-SLOG@) VD Version: v11.16.3.0 build 246844 x64_n6 ( build 246844
(ipv6))
9:42:54 (MSC) (@MSC-SLOG@)
9:42:54 (MSC) (@MSC-SLOG@) === Startup/Restart Info ===
9:42:54 (MSC) (@MSC-SLOG@) Options file used: None
9:42:54 (MSC) (@MSC-SLOG@) Is vendor daemon a CVD: No
9:42:54 (MSC) (@MSC-SLOG@) Is FlexNet Licensing Service installed and compatible: No
9:42:54 (MSC) (@MSC-SLOG@) FlexNet Licensing Service Version: -NA-
9:42:54 (MSC) (@MSC-SLOG@) Is TS accessed: No
9:42:54 (MSC) (@MSC-SLOG@) TS access time: -NA-
9:42:54 (MSC) (@MSC-SLOG@) Number of VD restarts since LS startup: 0
9:42:54 (MSC) (@MSC-SLOG@)
9:42:54 (MSC) (@MSC-SLOG@) === Network Info ===
9:42:54 (MSC) (@MSC-SLOG@) Listening port: 19980
9:42:54 (MSC) (@MSC-SLOG@) Daemon select timeout (in seconds): 1
9:42:54 (MSC) (@MSC-SLOG@)
9:42:54 (MSC) (@MSC-SLOG@) === Host Info ===
9:42:54 (MSC) (@MSC-SLOG@) Host used in license file: PPJ-PC2015
9:42:54 (MSC) (@MSC-SLOG@) HostID node-locked in license file: 9890969f897a
9:42:54 (MSC) (@MSC-SLOG@) HostID of the License Server: 9890969f897a 0a002700000f
9:42:54 (MSC) (@MSC-SLOG@) Running on Hypervisor: None (Physical)
9:42:54 (MSC) (@MSC-SLOG@) ================================================
9:42:54 (MSC) Loading feature details 3
9:42:54 (MSC) Usage records are being written to C:\MSC.Software\MSC
Licensing\Helium\LOG/mscusage_2019-10-11.ddu
```

Client License Configuration

Configuration of the Licensing System

During Digimat installation, a valid license file path is requested (see Figure 2-8). Given value defines `MSC_LICENSE_FILE` global environment variable. Most classical values are:

```
port@host
```

where

- `host` refers to the IP address or the host name of the computer on which runs the license server;
- `port` is the port to be used to establish the connection between the application and the license server (default port is 27500).

Multiple license servers can be defined with different @host references separated by ':' under Linux and ';' under Windows, for example:

```
MSC_LICENSE_FILE = 27500@WorkStation1:27500@192.168.1.10
```

If no `MSC_LICENSE_FILE` environment variable is defined, it is not possible to run any Digimat products.

MSC One Licensing System

Digimat also supports MSC One licensing system. MSC One licensing is such that it allows most Hexagon products to use a shared pool of license tokens.

All Digimat capabilities are working in the same way as with the traditional licensing system explained in the previous section, except that:

- The fiber orientation estimator embedded in Digimat-RP is not available.
- Crystal plasticity functionalities in Digimat-FE is not available.

Installation of MSC One license system is identical to ones of seat-based license systems (please refer to section License Server Installation).

Licensing Location Check

The licensing location defined by `MSC_LICENSE_FILE` can be directly checked by using the **License** option of the Digimat platform (see Figure 1-2). Clicking on **Query status** will give a status of license server defined in `MSC_LICENSE_FILE` (number of available/used licenses...).

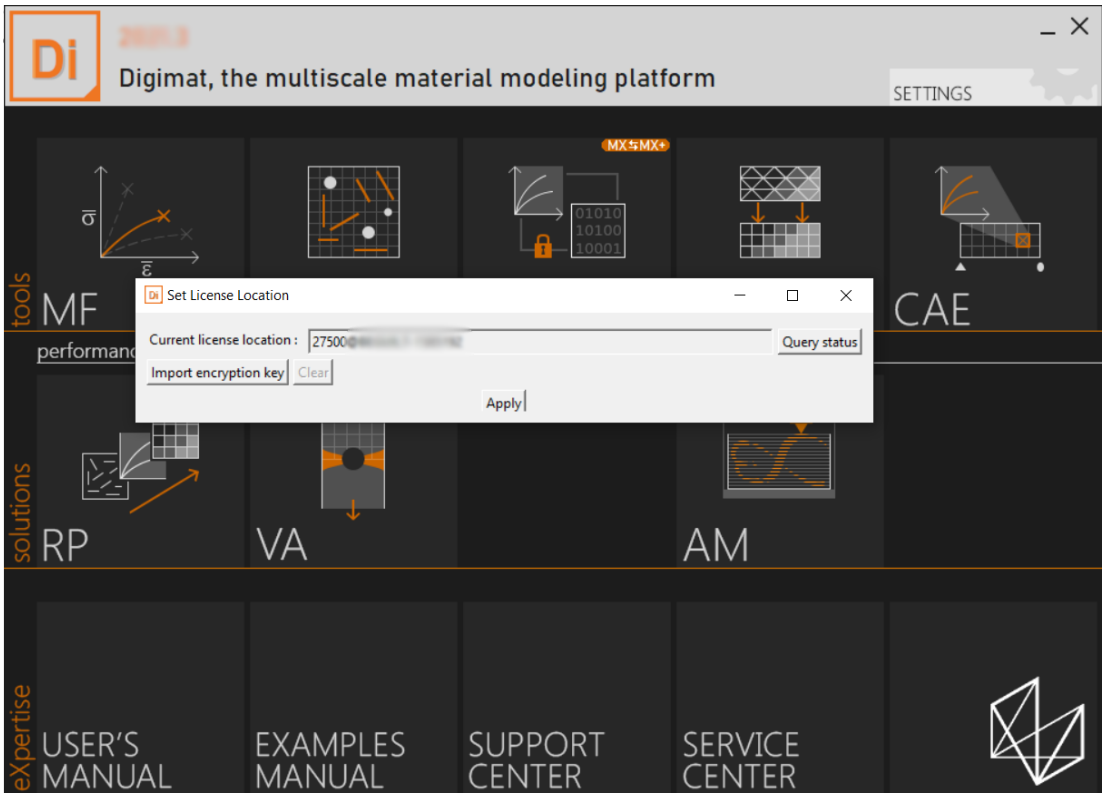


Figure 1-2 Checking the licensing location via the Digimat GUI.

2 Installing Digimat

- Introduction
- Local Installation of Digimat on a Windows Machine
- Install Digimat on a Network Windows Machine
- Command Line Digimat Installation on a Windows Machine
- Installation of Digimat-MX Database
- Installation of Digimat on a Linux machine
- Digimat Settings

Introduction

The following sections are intended to give a quick and general overview over the single steps of the Digimat installation. A general overview over the installation procedure is followed by a typical example of a local Digimat installation on a Windows computer. For detailed procedures please refer to the dedicated chapters.

It is highly recommended to follow the described procedures as given in the overview step by step to receive a stable installation of Digimat software!

Files for testing Digimat 2023.1 installation are available in Digimat documentation.

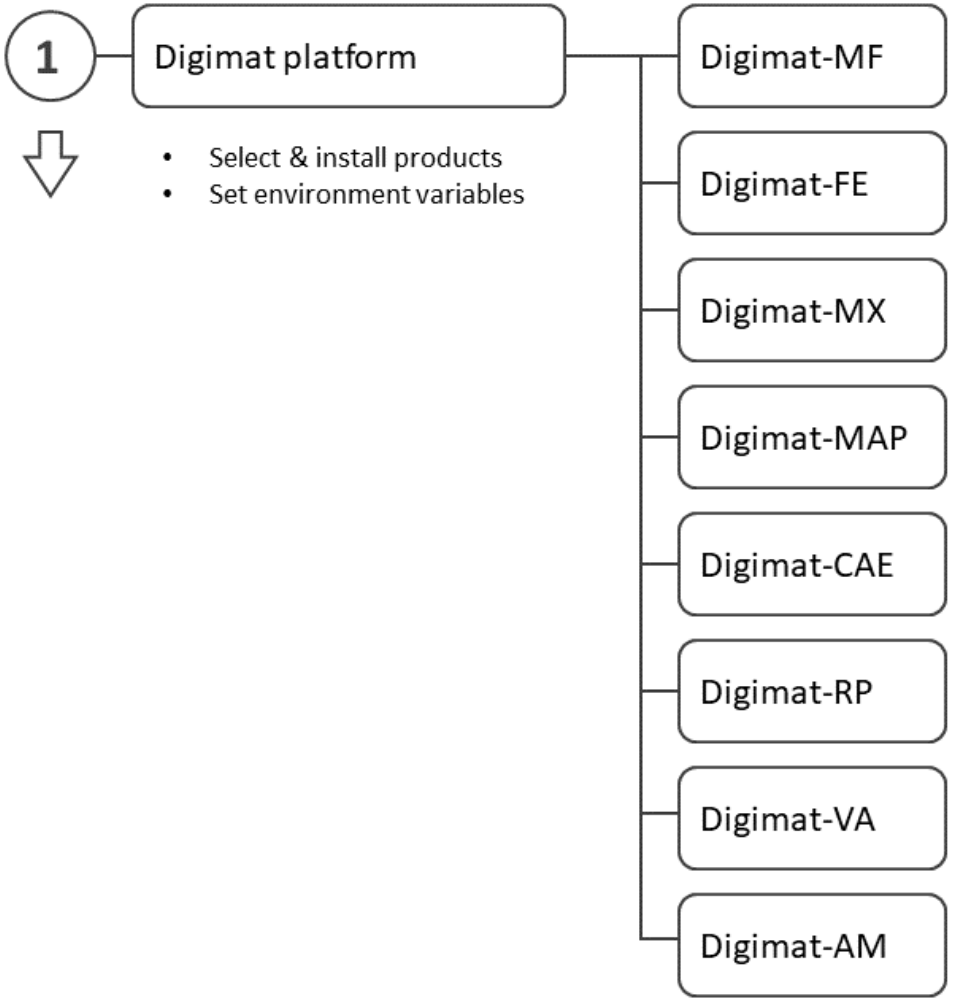


Figure 2-1 General overview over steps required to receive a stable installation of Digimat software.

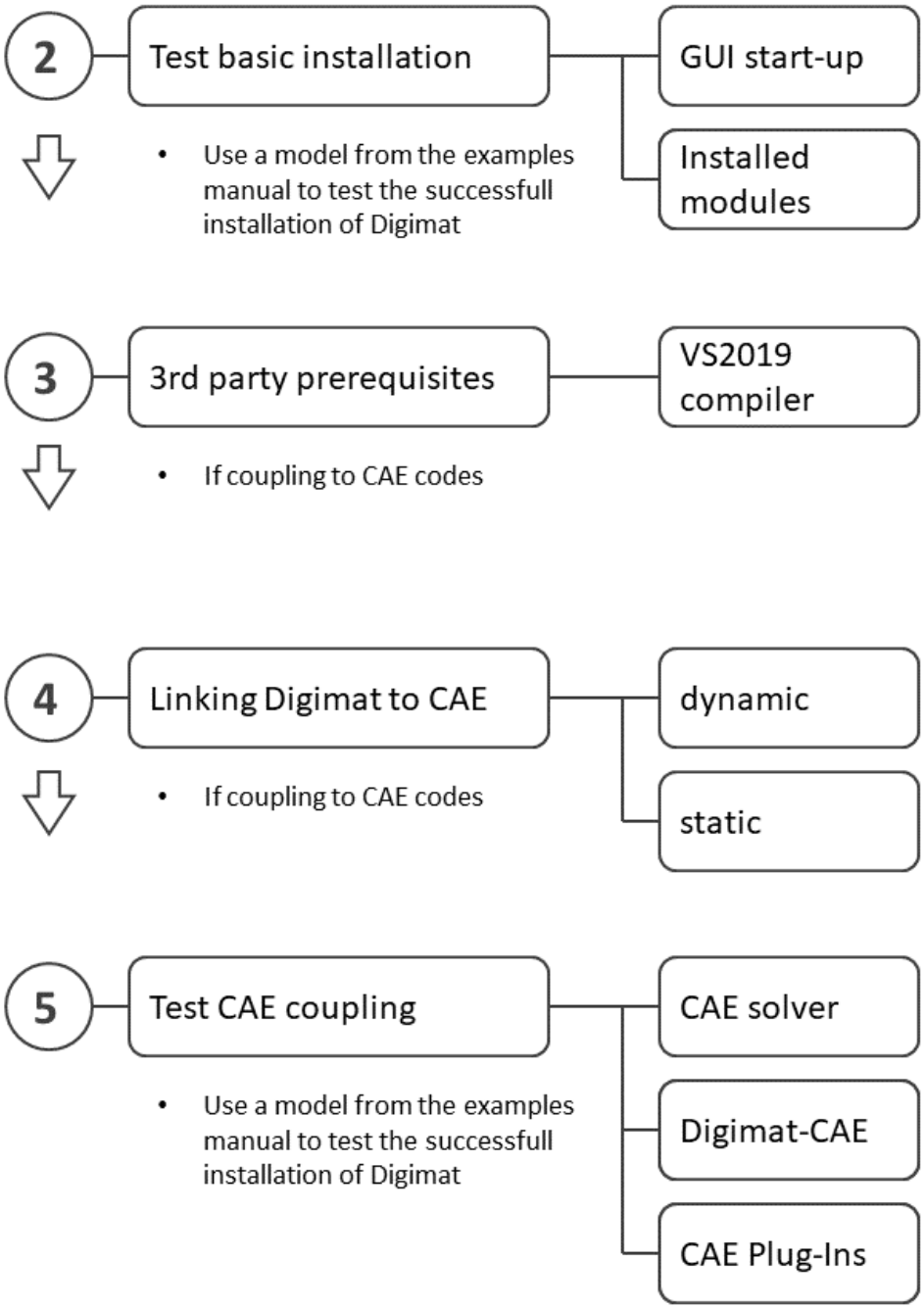


Figure 2-2 General overview over steps required to receive a stable installation of Digimat software.

Local Installation of Digimat on a Windows Machine

This section demonstrates the most straightforward way to create a **FULL standard installation of the Digimat software on a Windows machine**.

Step-by-step: Digimat software

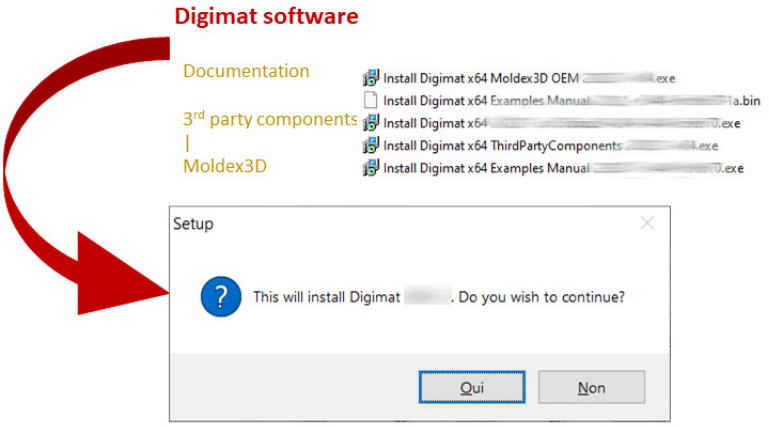


Figure 2-3 Archive from Hexagon Download Center (https://mscsoftware.subscribenet.com/) contains 3 or 4 executable files according to download of standard installation or installation including Digimat- RP/Moldex3D: main installer, third-Party installer, examples manual installer, Digimat-RP/Moldex3D installer (optional).

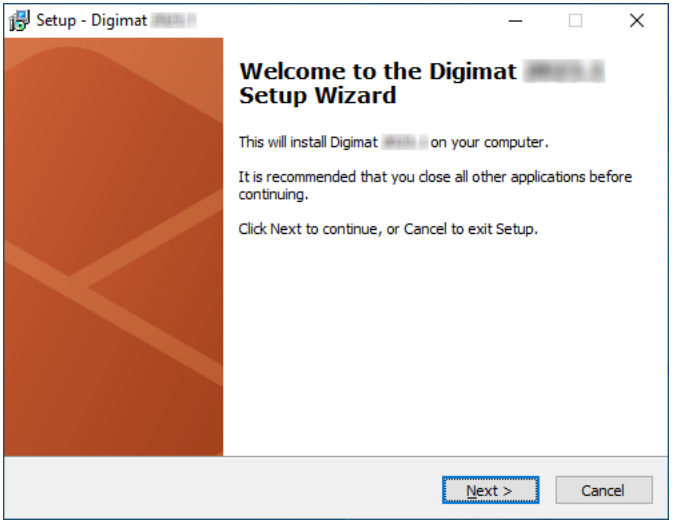


Figure 2-4 Opening prompt of the Digimat installer. Follow the given instructions step-by-step.

Digimat

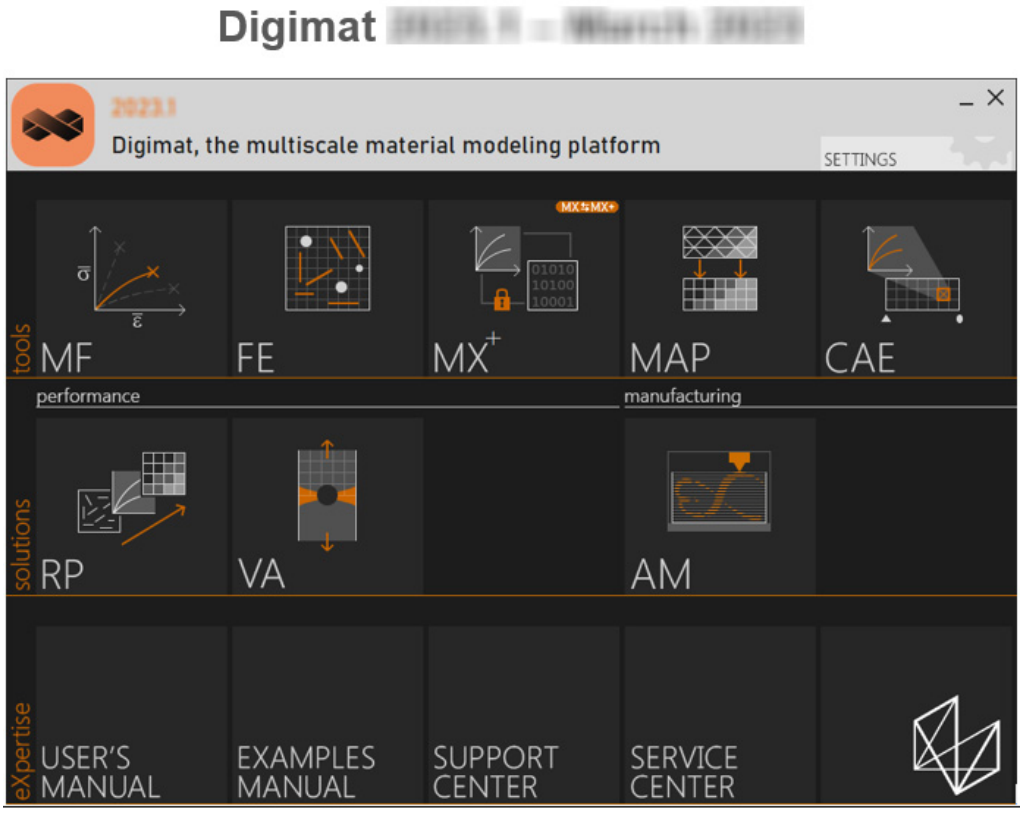


Figure 2-5 Upon execution of the Digimat installer, the release notes will be shown in a separate PDF viewer.

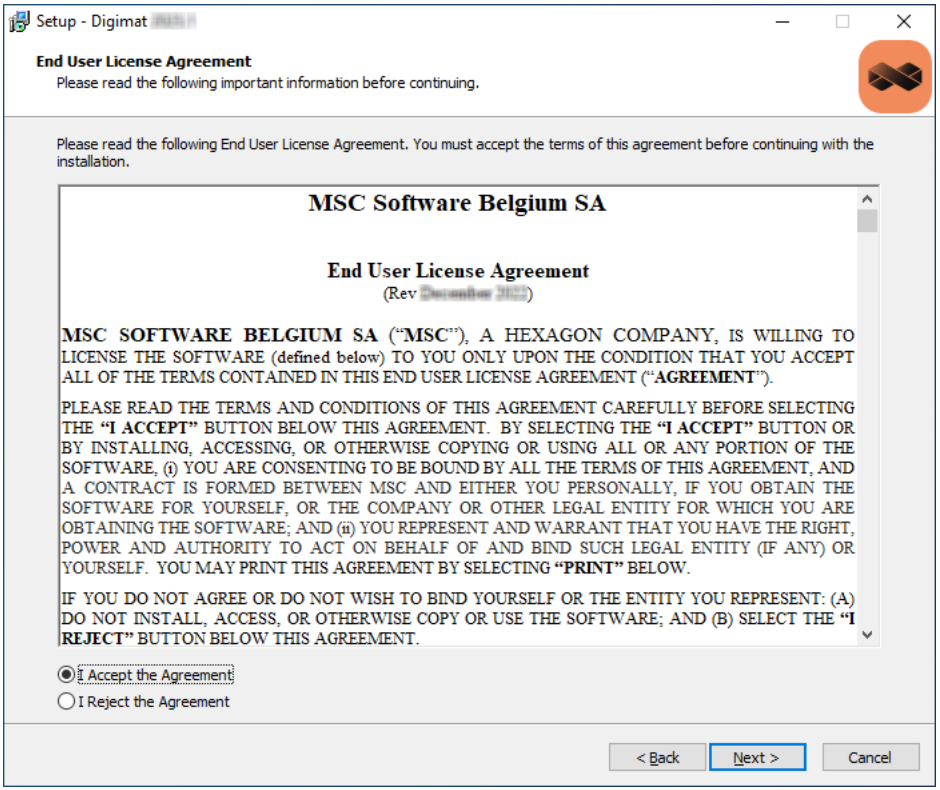


Figure 2-6 Please read carefully the software license agreement. It must be agreed to before being able to continue with the installation procedure.

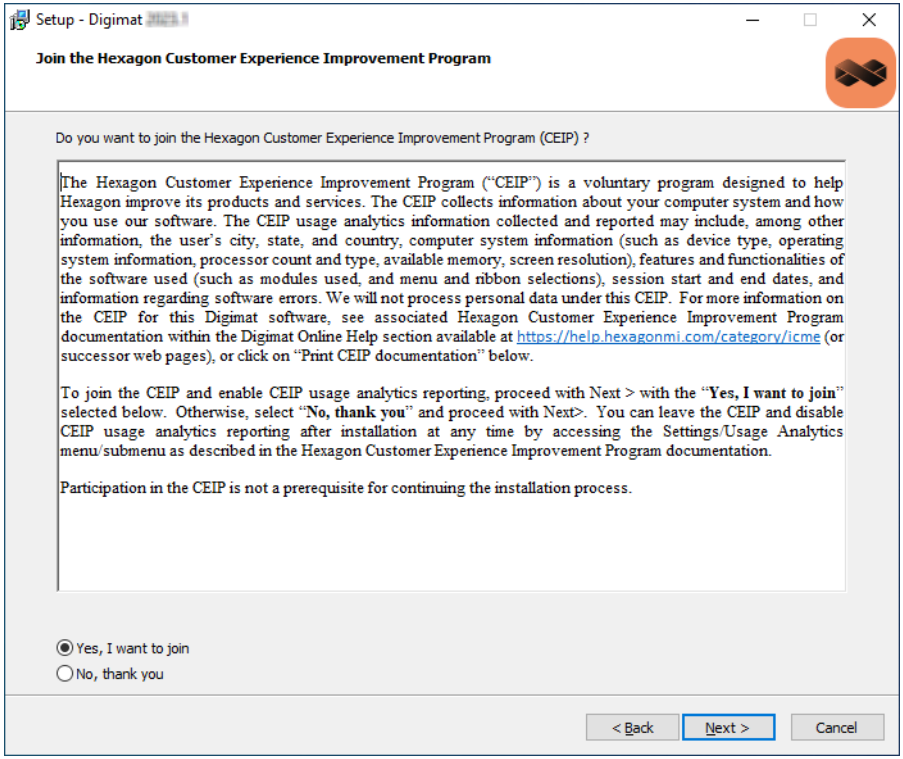


Figure 2-7 Please read the description of the Hexagon Customer Experience Improvement Program. Click "Yes, I want to join" if you agree to join the program.

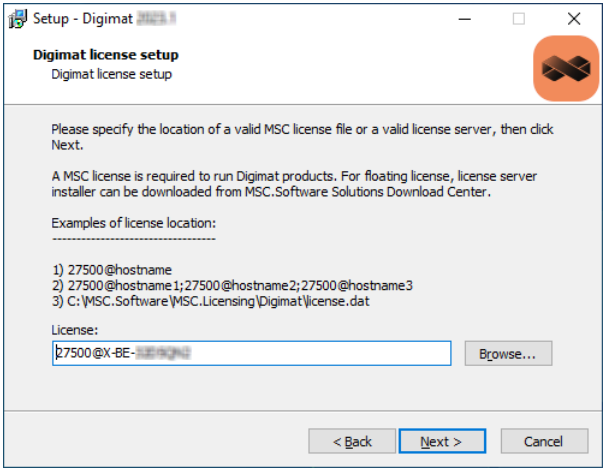


Figure 2-8 The IP address for communication with the license service has to be given, preceded by the @. If needed, user can also specify explicitly the port used by the license server, e.g., 27500@hostname. Port specification should only be used if it is explicitly specified in the license server.

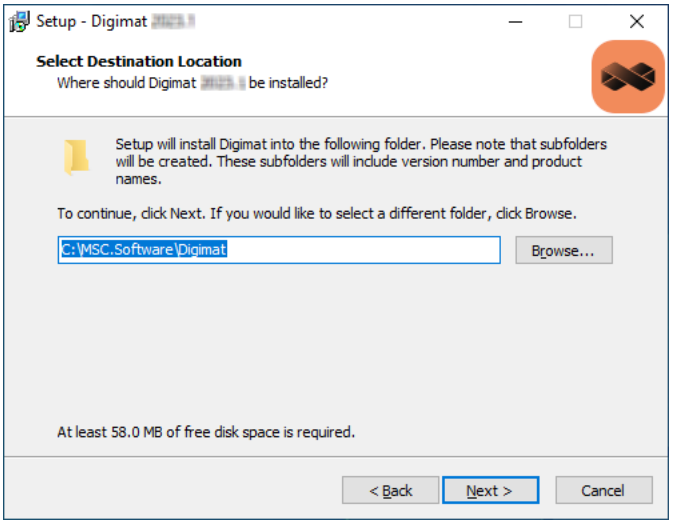


Figure 2-9 The destination folder for the Digimat installation has to be given. A 2023.1 directory will be automatically created.

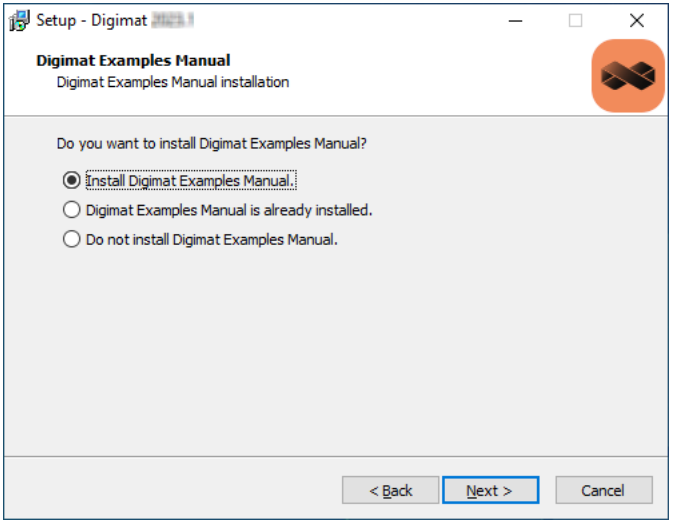


Figure 2-10 Choice of installing Digimat Examples Manual. If user doesn't choose Digimat Examples Manual, it is still possible to install Digimat Examples Manual step after complete Digimat installation.

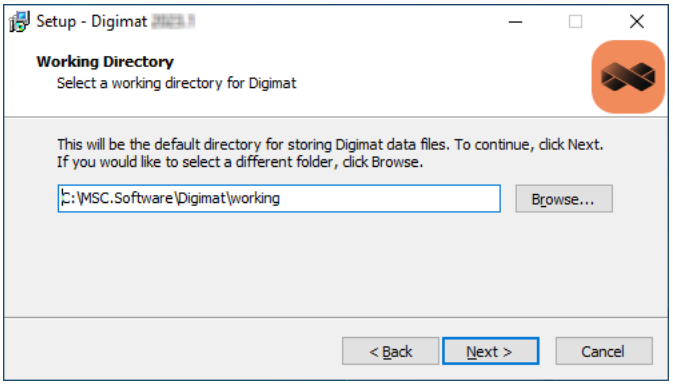


Figure 2-11 The destination folder for the Digimat working directory has to be given. This directory can be located anywhere on the computer and shared also between different versions of Digimat.

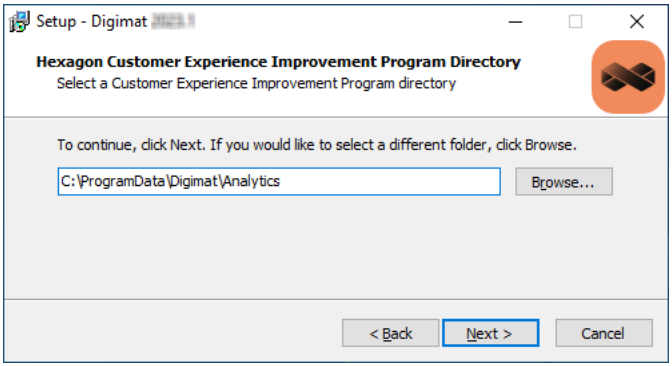


Figure 2-12 The preferences folder for the Hexagon Customer Experience Improvement Program directory has to be given. This directory can be located anywhere on the computer and can be shared between different versions of Digimat.

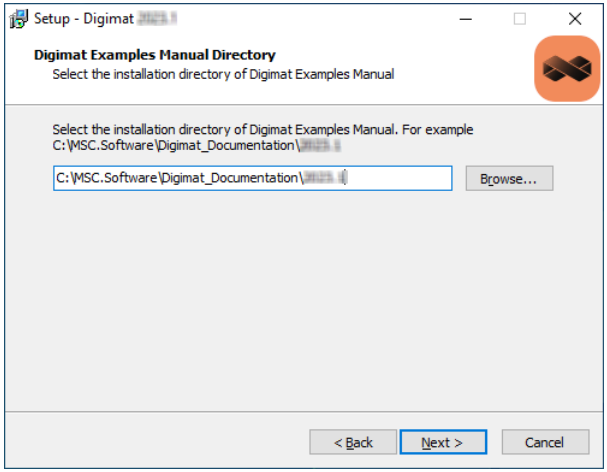


Figure 2-13 If the option Digimat documentation is already installed is selected, user has to point to the Digimat documentation directory. So the settings of Digimat will be automatically updated to point to this documentation. Note that user cannot point to a Digimat documentation prior to Digimat 2016.0.

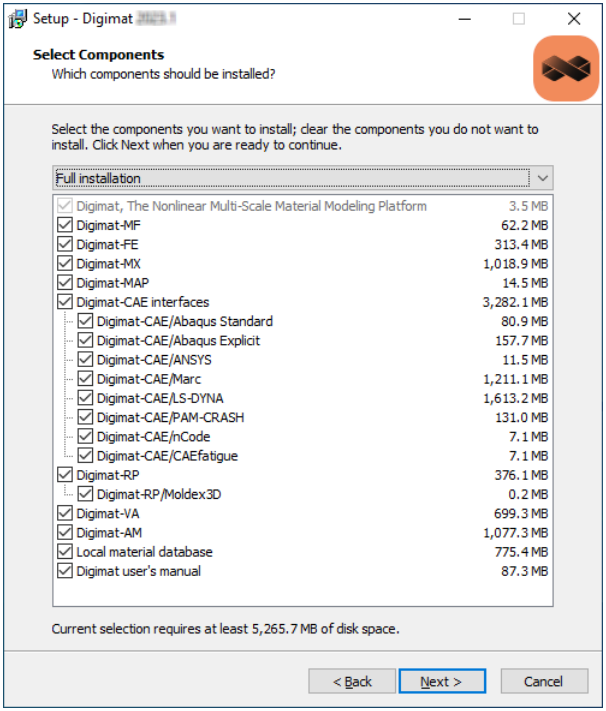


Figure 2-14 The required Digimat modules can be chosen individually to save disk space for the installation. In the default case as used here all modules will be installed. If Moldex3D installer has been downloaded, it will be installed by default, unless the sub-component Moldex3D integrated into Digimat-RP is unchecked. When intending to use Digimat-MX Remote database, Local database component must be unchecked.

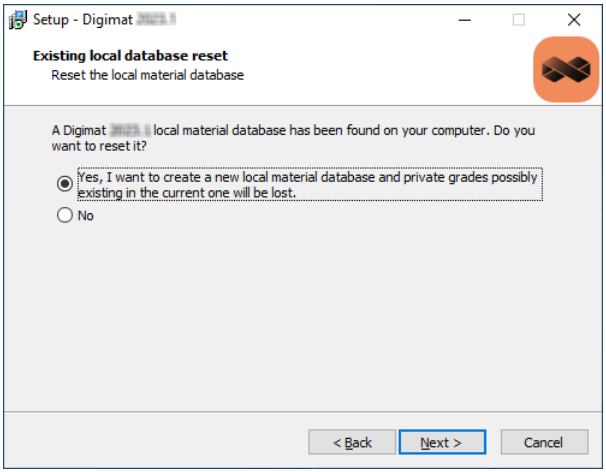


Figure 2-15 The local material database can be reset if a Beta version of Digimat 2023.1 has been previously installed (Database Installation).

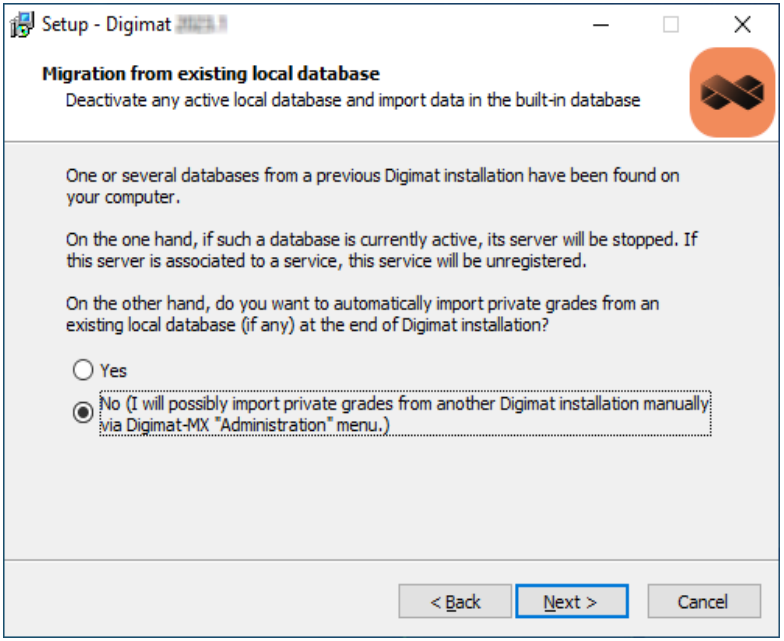


Figure 2-16 To be fully operational, the new built-in local database may require the migration of an existing local database at the end of Digimat installation (see Database Installation and Figure 2-28).

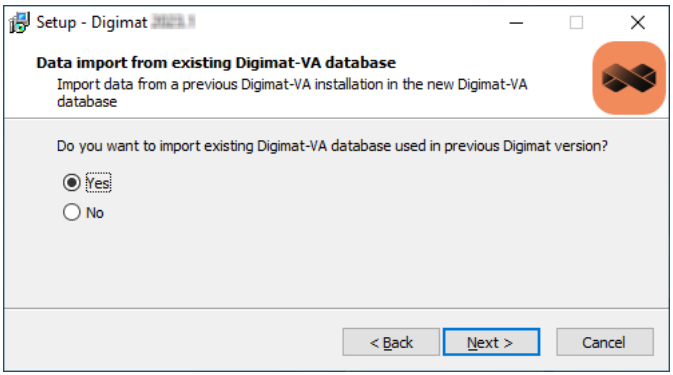


Figure 2-17 If Digimat-VA component is selected and if a previous version of Digimat-VA database is found, this database can be imported in database 2023.1.

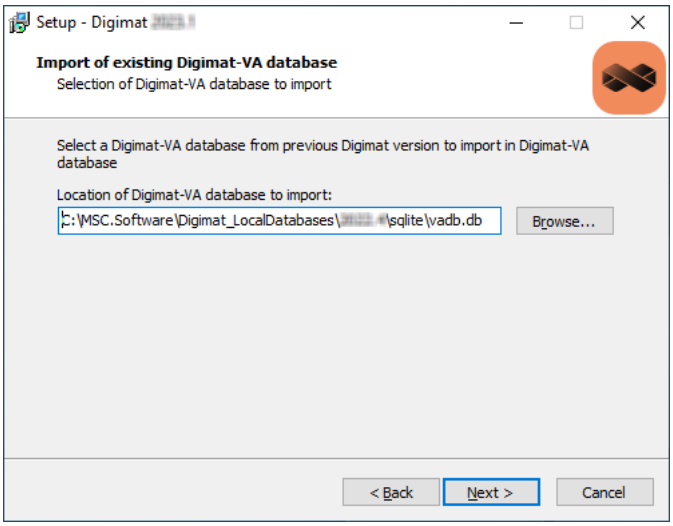


Figure 2-18 If user wants to import a previous version of Digimat-VA database, the path to this Digimat-VA database must be given.

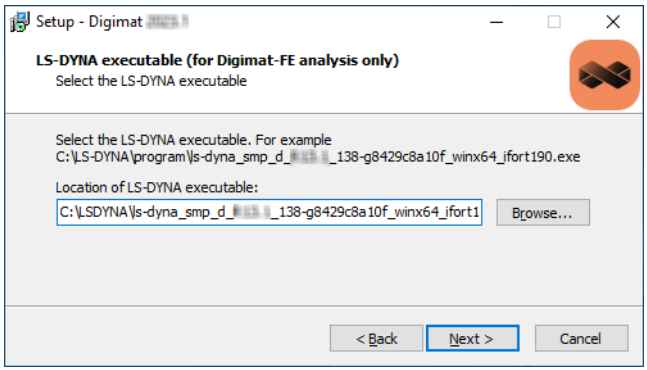


Figure 2-19 If Digimat-FE component is selected, path of an existing LS-DYNA executable is requested to allow running Digimat-FE computation using LS-DYNA implicit. If not using LS-DYNA FEA solver the step can be skipped and the input field left blank. If needed this path can be specified in a later step via Digimat settings as explained in DIGIMAT_Settings.ini File.

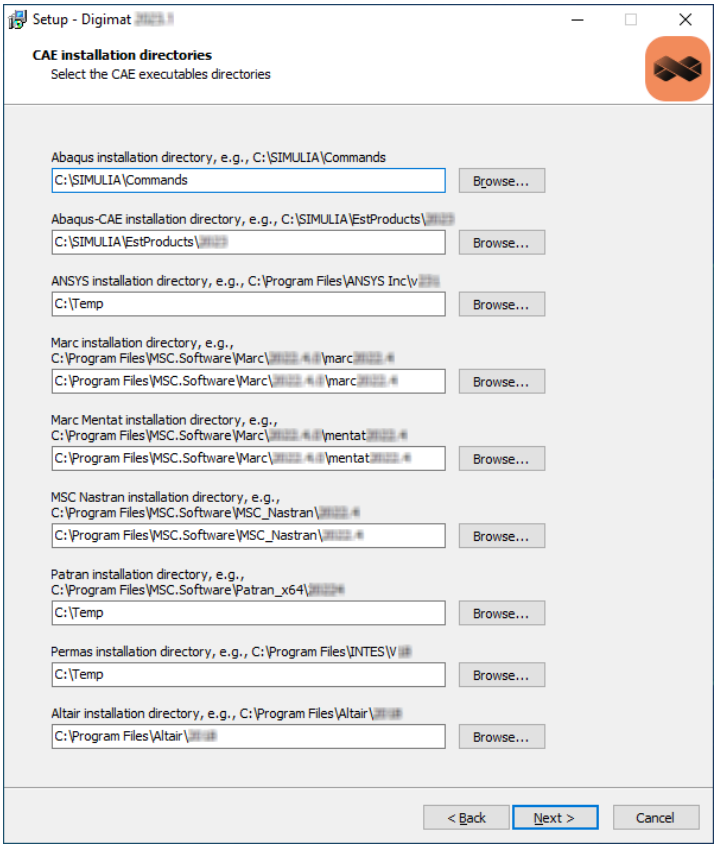


Figure 2-20 If Digimat-CAE, Digimat-RP or Digimat-FE components are selected, paths to existing CAE installations are requested. If not using this FEA solver the step can be skipped and the input field left blank. If needed this path can be specified in a later step via Digimat settings as explained in DIGIMAT_Settings.ini File.

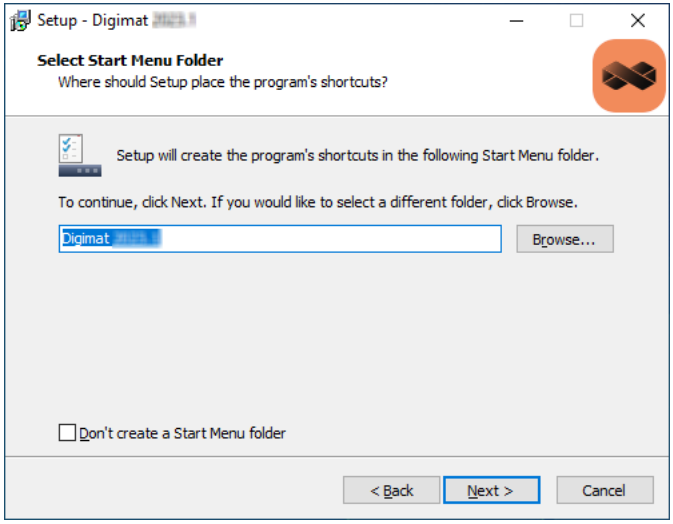


Figure 2-21 An individual name for the Digimat shortcut can be specified.

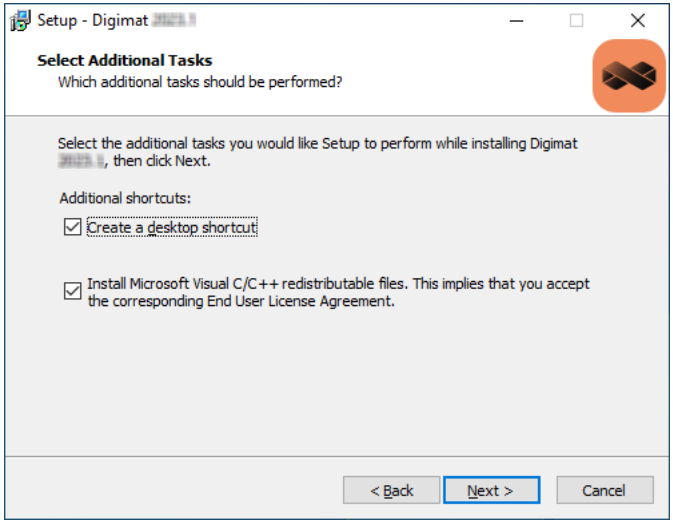


Figure 2-22 A desktop icon can be created. User can also choose to install Visual 2010, 2012, 2015 and 2017 C/C++ redistributable files together with Digimat. These files are required to run Digimat computations. In this case, Visual C/C++ redistributable files will be installed.

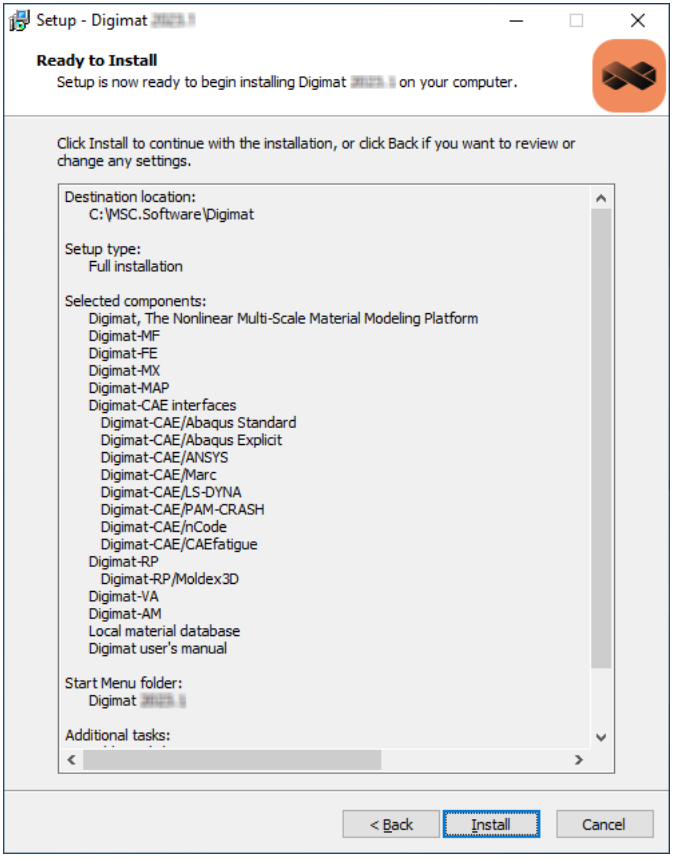


Figure 2-23 A summary of the installation details is given and can be checked. Proceeding with the Install button will start the installation of Digimat.

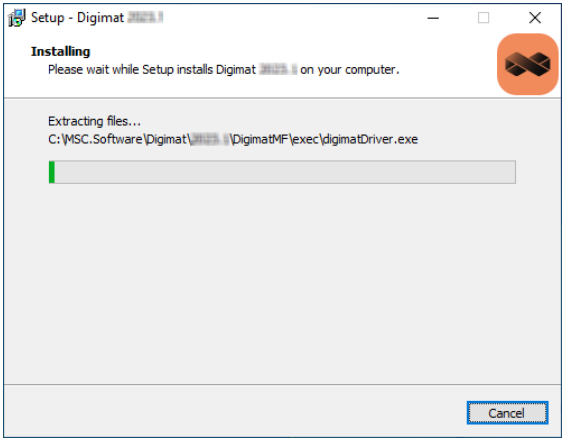


Figure 2-24 The installation of the Digimat core software is executed.

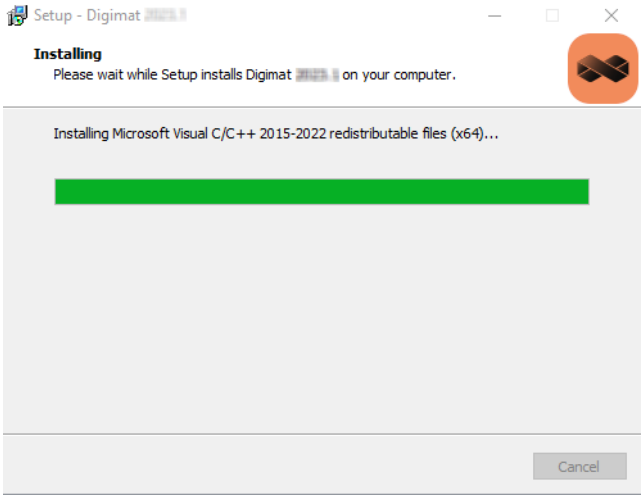


Figure 2-25 The installation of the Microsoft Visual C++ 2015-2022 redistributables is executed.

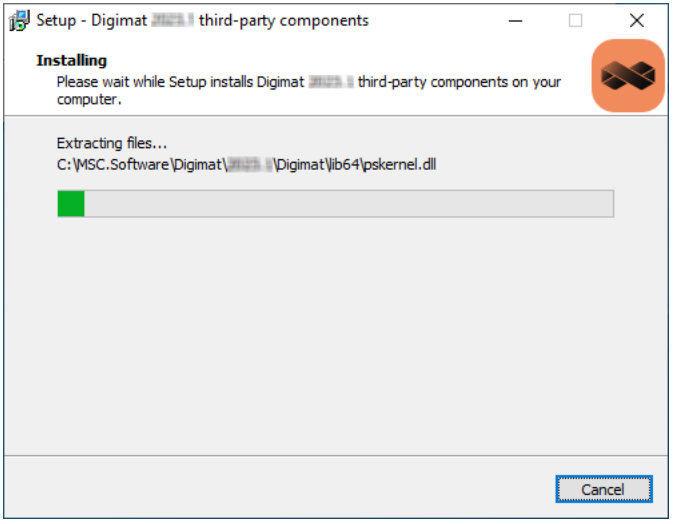


Figure 2-26 Third party products are installed. Third party components must be installed in order to be able to run the Digimat software.

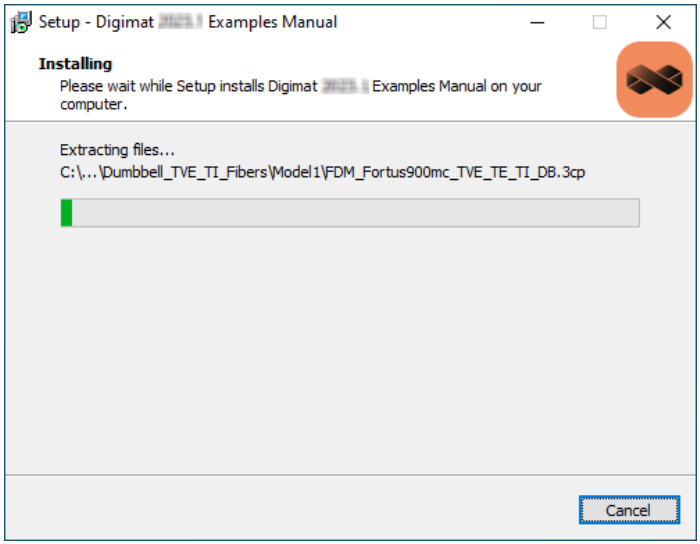


Figure 2-27 As a last step, the Digimat Examples Manual is installed (if chosen).

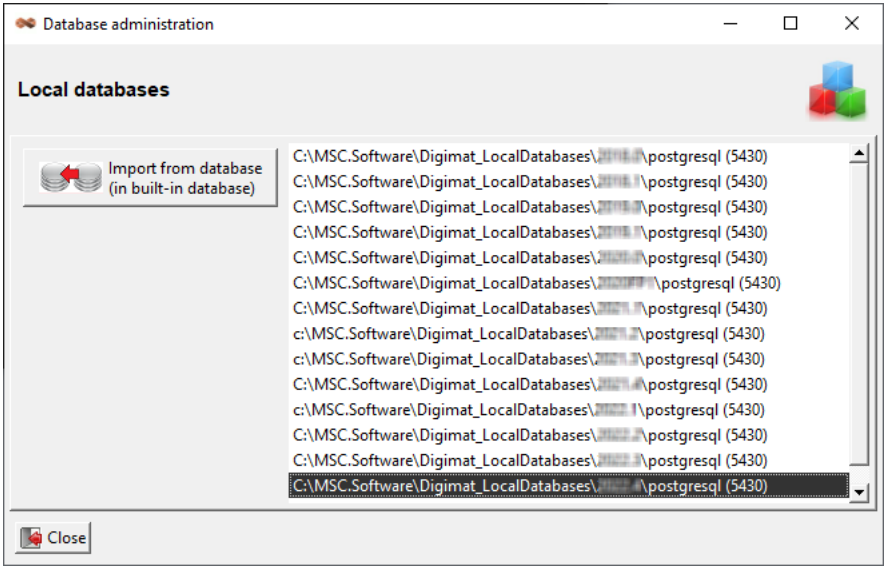


Figure 2-28 If the migration of a previous local database in the new local database is required, the directory of this database must be selected prior to the actual database import (Refer Figure 2-16 and Local Databases Administration *in Digimat-MX User's Guide*).

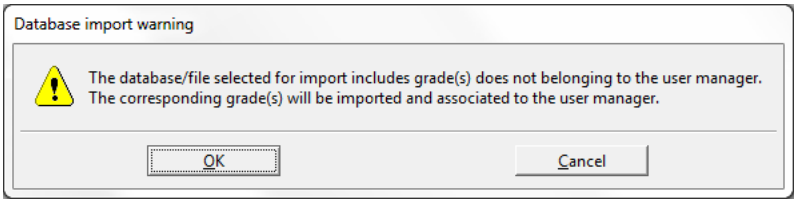


Figure 2-29 During database import from previous Digimat version, data belonging to different users are addressed specifically.

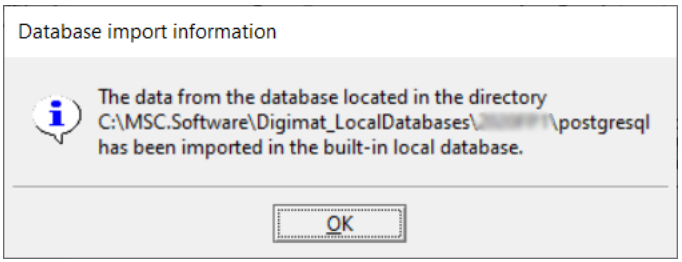


Figure 2-30 Successful database import from previous Digimat version.

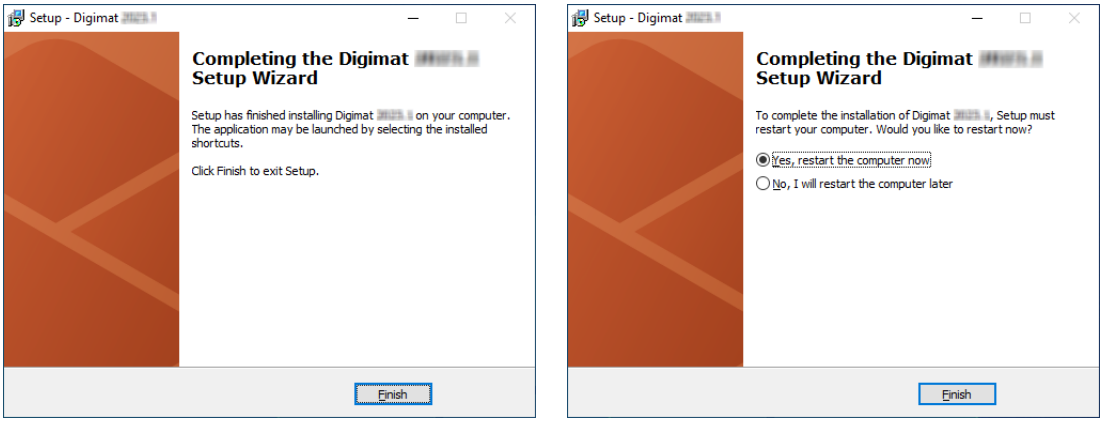


Figure 2-31 Upon finalization of the installation, if requested, the user can choose to reboot the computer immediately.

To ensure safely a fully functional installation of Digimat, the reboot of the machine is mandatory!

Encryption key management

The encryption keys needed to decrypt the material files are not managed via the license file. The keys are handled by the Digimat platform and written in the Digimat configuration file.

Here is the procedure to be performed prior to being able to decrypt any material files:

- The encryption keys are delivered by Hexagon (digimat.support@hexagon.com) within files named `key_number.priv`. If not already received please contact your Digimat support. You will have at least two encryption keys to handle, one to decrypt material files that were encrypted for you only, and one to decrypt material files that can be decrypted by everyone.
- Go to the Digimat platform to have access to the license management functionalities (see Figure 2-32). There you will have access to a button named **Import encryption key**. Click on the button, select your `.priv` file, click **OK** and then click the **Apply** button. Perform this procedure as many time you have received different encryption key files.

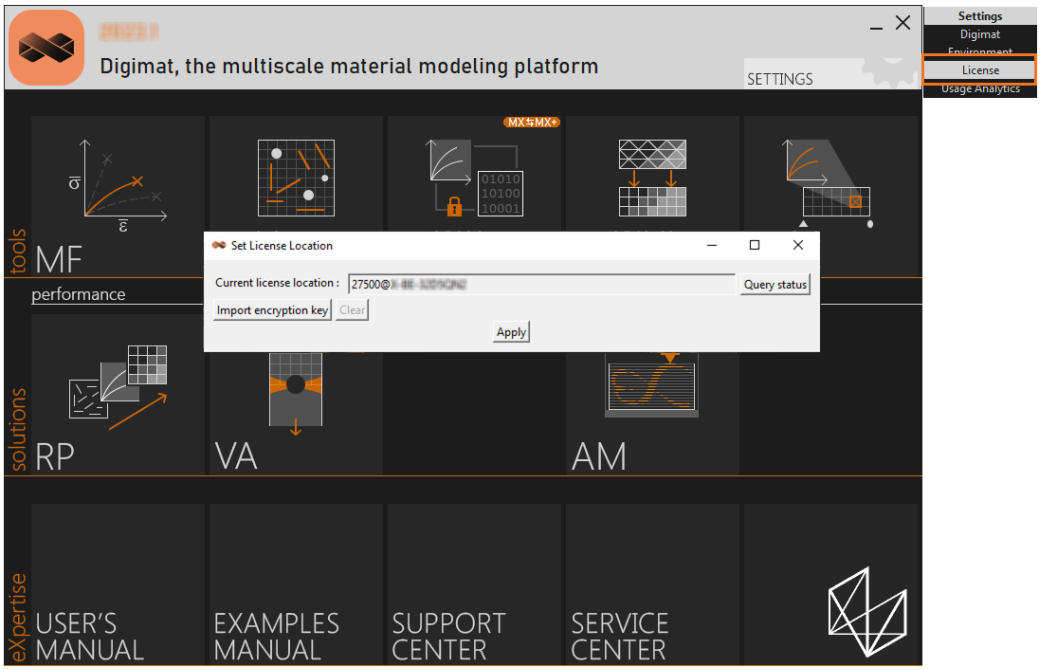


Figure 2-32 Import private encryption key using Digimat platform.

Installation of Intel-MPI 2019 Hydra Service for Fiber Orientation Estimation

When installing Digimat-RP/Moldex3D for fiber orientation estimation, Intel-MPI 2019 hydra service is automatically installed to allow parallel computation for fiber orientation estimation. The executables associated to the service are located in directory "`C:\Program Files\Intel MPI 2019`. It appears in Windows task manager with the name "`impi_hydra`" (see Figure 2-33 and Figure 2-34), corresponding to the executable named "`hydra_service.exe`". Only one instance of this service with "`impi_hydra`" name can run on a computer.

So, if another instance of the service is running with `impi_hydra` name when installing Digimat, this instance will be replaced by the one installed by Digimat-RP/Moldex3D.

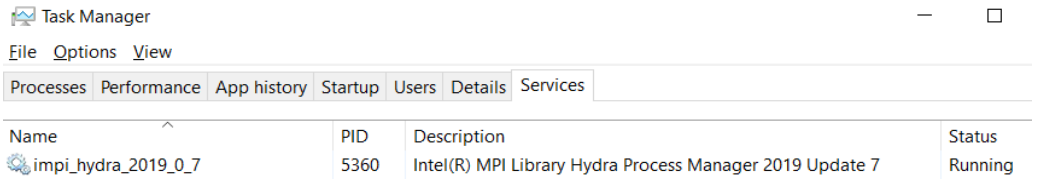


Figure 2-33 Intel-MPI 2019 hydra service in task manager.

Remark: Intel-MPI 2019 hydra service is not uninstalled when uninstalling Digimat.

To remove an existing Intel-MPI hydra service:

- Open a Command prompt as an administrator.
- Find the directory of Intel-MPI hydra service to be removed. This directory can be found by clicking on properties of the existing service (e.g., `C:\Program Files\Intel MPI 2019`), see Figure 2-34 and Figure 2-35. The name of the executable associated to the service (`hydra_service.exe`) can also be found in the properties of the service. Type in the command prompt:
  - cd `C:\Program Files\Intel MPI 2019`
  - `hydra_service.exe -stop`
  - `hydra_service.exe -remove`

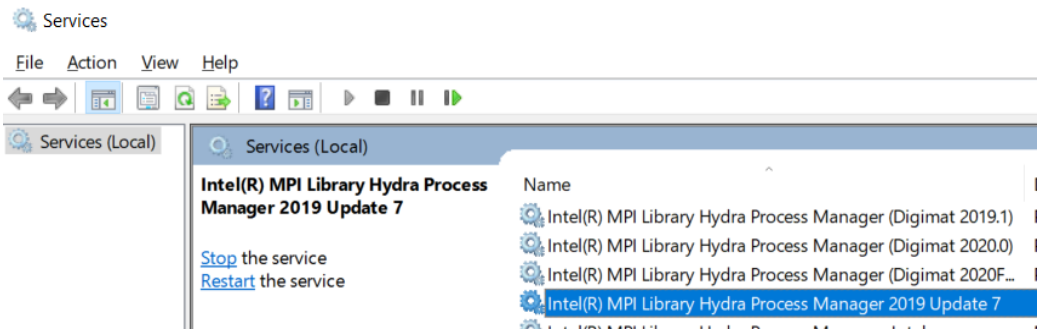


Figure 2-34 Intel-MPI hydra service.

To re-install a new Intel-MPI hydra service:

- Open a Command prompt as an administrator.
- Go to directory of Intel-MPI hydra service to be installed.
- Type `hydra_service.exe -install`.

In case of conflicts with another Intel-MPI hydra service, please contact digimat.support@hexagon.com.

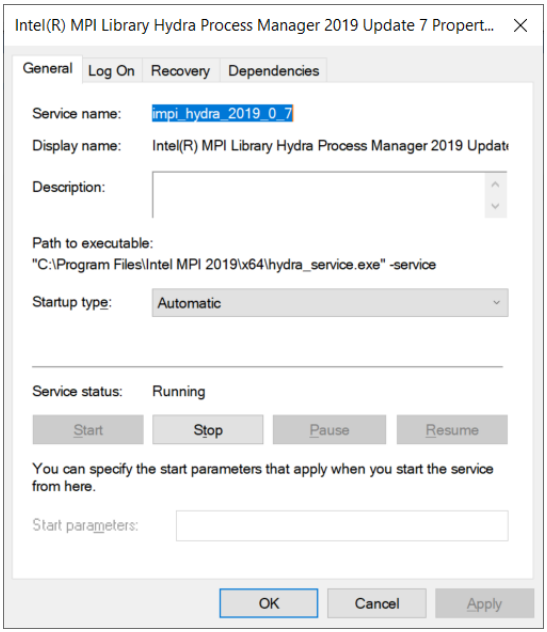


Figure 2-35 Intel-MPI hydra service installation directory.

Install Digimat on a Network Windows Machine

This section explains extra operations to be able to use Digimat with network installation.

First operation consists in installing Digimat on network machine following procedure described in section Local Installation of Digimat on a Windows Machine.

Digimat Configuration on Server Machine

After having installed Digimat on server machine, it is needed to adapt path defined in `DIGIMAT_Settings.ini` file, so that they point now to shared location. For example, suppose that Digimat is installed on server machine in directory:

```
C:\MSC.Software\Digimat\2023.1
```

and suppose that shared location on client machine is:

```
\\AppShare\MSC.Software\Digimat\2023.1
```

Then, `DIGIMAT_Settings.ini` file is located in directory:

```
C:\MSC.Software\Digimat\2023.1\Digimat\exec
```

This file must be edited in 4 steps:

- Path to Digimat executables must be changed to point to shared path. For example, key

  ```
  DIGIMATMF_Directory=C:\MSC.Software\Digimat\2023.1\DigimatMF\exec
  ```

  must be changed into:

  ```
  DIGIMATMF_Directory=\\AppShare\MSC.Software\Digimat\2023.1\DigimatMF\exec
  ```

- Path to working directory must point to a local directory, e.g., `C:\temp`: key

  ```
  Working_Directory=C:\MSC.Software\Digimat\working
  ```

  must be changed into:

  ```
  C:\temp
  ```

- Finally, path to CAE codes must also be adapted if needed.
- If Customer Experience Improvement Program has been set during local installation, it is needed to change setting `DIGIMAT_Analytics_Directory` to make it point to a location visible by all client machines

In same directory as `DIGIMAT_Settings.ini` file, a Python script (`installDigimatNetwork.py`) is provided. Running this python script will automatically adapt path to Digimat executables and to Digimat working directories. This file is used in the following way:

- Open the script in a text editor
- Adapt the path for initial and new Digimat installation path, for new Digimat working directory and for new Customer Experience Improvement Program directory:

  ```
  instdir = 'C:\\MSC.Software\\Digimat'
  new_instdir = '\\\\AppShare\\MSC.Software\\Digimat'
  ```

```
new_workdir = 'C:\\temp'
new_ciepdir = '\\\\AppShare\\ProgramData\\Digimat\\Analytics'
```

- Run Python script like:

  ```
  python.exe installDigimatNetwork.py
  ```

- `DIGIMAT_Settings.ini` file is now adapted with new path to Digimat executables. Backup of initial `DIGIMAT_Settings.ini` file (with _old suffix) is also created.

Refer DIGIMAT_Settings.ini File for full explanation about the contents of it.

Digimat Configuration on Client Machine

In order to make Digimat fully functional on client machine, three environment variables must be defined on each client machine:

- `MSC_LICENSE_FILE` pointing to license server.
- `DIGIMAT_BIN_20231` pointing to `DIGIMAT_Settings.ini` file of shared installation, for example:

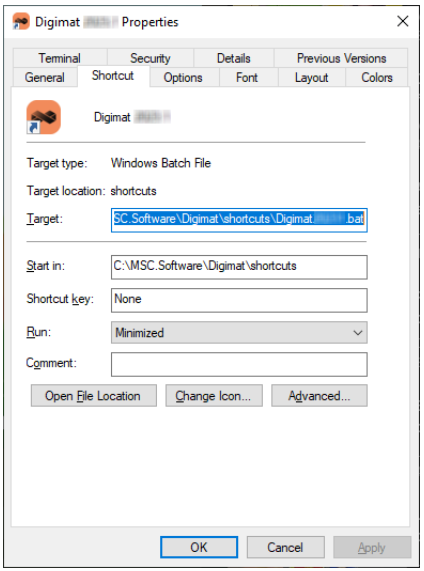
  ```
  \\AppShare\MSC.Software\Digimat\2023.1\Digimat\exec
  ```

- `DIGIMAT_FONT_CACHE` pointing to a local directory accessible in writing mode by the user, typically

  ```
  C:\temp\fonts
  ```

  This directory will contain the files related to the font cache specific to the client machine, which will be automatically created the first time they are needed. This creation operation may take up to several minutes but will happen only once, provided that the font cache directory is not modified or deleted. Next usage will then be smoother.

- It is also required to manually update the target path of the Digimat executable being used in the client machine to start Digimat.
  Initial path: `C:\MSC.Software\Digimat\shortcuts\Digimat20231.bat`. Needed path on client: `\\AppShare\MSC.Software\Digimat\shortcuts\Digimat20231.bat` (see figure).



Reverse Engineering on Client Machine

The reserve engineering can be very slow in case of network installation where the Digimat executables are called through the local network. To solve this problem, all the libraries and executables are copied on the local computer, on which Digimat will be run, in a folder located in Digimat-MX working directory.

To activate this particular installation, it is necessary to modify the Digimat setting `DIGIMATMX_Local_Installation` to true. This can be done through Digimat platform setting (see Figure 2-36). Apply this setting modification will create, at the first execution of Digimat-MX, a new folder named bin in the Digimat-MX working directory and copy all the needed files.

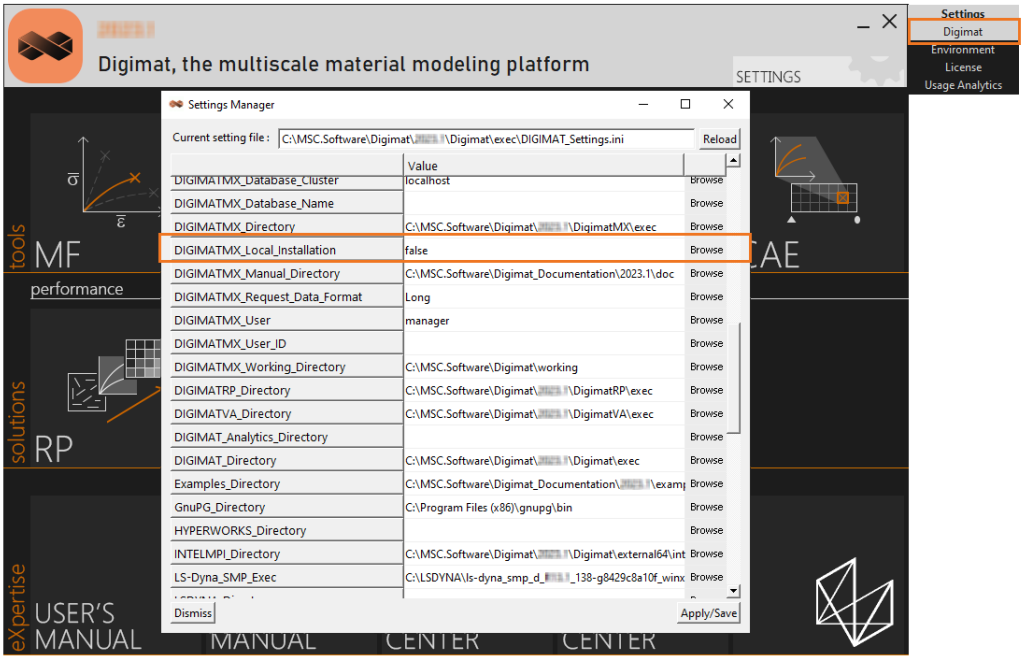


Figure 2-36 Digimat-MX local installation setting.

Command Line Digimat Installation on a Windows Machine

It is possible to install Digimat using command line operation. In a command line, as administrator mode, execute following command in directory where Digimat installation files are located:

```
"Install Digimat x64 2023.1-XXXXXXXXXX-YYYY-ZZZZ-WWWWWWWW" /SP- /NOCANCEL /SILENT /TYPE="full" /NORESTART /LOGFILE="C:\TEMP\instll.log" /GROUP="Digimat 2023.1" /DIR="C:\MSC.Software\Digimat"
```

Following parameters can be adapted:

- `/GROUP` argument defines the **Start Menu** entry.
- `/DIR` argument defines the installation directory. A subdirectory **2023.1** will be created.
- `/LOGFILE` argument defines path and name of Digimat installation log file.
- If Moldex3D installer is present, it is mandatory to restart computer. To that aim, remove `/NORESTART` argument.

Installation of Digimat-MX Database

This section explains the steps to install Digimat-MX remote database.

Database Installation

Material data can be easily accessed from databases in several Digimat modules, Digimat-MX in particular. Hence such a database must be created to enable material data access in 2 different ways:

- Local database: The database is physically located on the disk of the (only) computer used to connect to it. Such configuration is relevant when a single user needs material data access.
- Remote database: The database is physically located on a computer server to which several Digimat-MX installations installed on distant computers can connect. Such configuration is relevant when several users need to share material data.

Local Database Installation

A local database is exclusively created during the Digimat installation process in a directory besides Digimat installation directory e.g.,

`C:\MSC.Software\Digimat_LocalDatabases\2023.1\postgresql`.

As the new – so-called built-in – local database is likely intended to become the default one, together with the new Digimat installation, its creation deactivates – but does not modify – any database created with an earlier Digimat version on which a server is running. Such an active database may exist especially with Digimat 2016.0 and earlier versions, which enabled advanced local databases administration similar to remote databases administration. Hence, stopping any running server and unregistering any active service prior to Digimat installation would prevent from any inconvenience, e.g., abrupt disconnection.

The built-in local database creation involves 3 possible actions:

- Select the component **Local material database** (see Figure 2-14; selected by default).
- Request to reset a previous database of the same Digimat version if such a database exists (see Figure 2-15; not requested by default). In such a case, the existing database is deleted before creating the new one (see Local Databases Administration in *Digimat-MX User's Guide*).
- Request to migrate a previous database of an earlier Digimat version if such a database exists (see Figure 2-16; not requested by default). In such a case, the private grades of the existing database are imported at the end of Digimat installation via Digimat-MX local databases administration window (see Local Databases Administration in *Digimat-MX User's Guide* and Figure 2-28).

Remote Database Installation

A remote database can be created after Digimat installation, via Digimat-MX remote database administration (see sections Remote Database Administration and Database Creation in *Digimat-MX User's Guide*). Such a database is usefully associated to a Windows service. In addition, its creation requires to stop the postgresql server of all remote databases currently running.

The creation of a remote database may even constitute the only purpose of Digimat installation, i.e., on a computer server. In such a case, any server (resp. service) running on a remote database of an earlier Digimat version already existing on the computer server must be stopped (resp. unregistered) with the corresponding Digimat-MX version prior to the uninstallation of the earlier version and the installation of the new version.

Install successively Digimat and the remote database as follows:

- Install Digimat and select only the Digimat-MX component (see Figure 2-14). In particular, do not select the component **Local material database**.
- Open Digimat-MX and do not connect to any database.
- Open the remote databases administration window via the menu **Administration** / **Remote databases/ Databases**.
- Select **New / Create**. (Define a service if appropriate.)
- Choose the new database and select **Server / Start** or **Service->Start**.
- Close Digimat-MX.

To continue working with a remote database of an earlier Digimat version already existing on the computer server, upgrade it as follows (see Database Upgrade in *Digimat-MX User's Guide*).

- Open Digimat-MX and do not connect to any database.
- Open the remote databases administration window via the menu **Administration / Remote databases / Databases**.
- Choose the existing database to upgrade and select **Server / Upgrade**.
- Choose the upgraded database and select **Server / Start** or **Service / Create** and **Service->Start**.
- Close Digimat-MX.

Installation of Digimat on a Linux machine

This section demonstrates the most straightforward way to create an installation of the Digimat software on a Linux machine.

It supposes that a Digimat license server has already been installed (see Release Guide License Server Installation).

- Step 1: Unzip installer in a temporary directory (see Figure 2-37):

  ```
  unzip Digimat2023.1-rNNNNN-MMMM-OOOO-Linux64bit.zip
  ```

- Step 2: Execute Digimat installation script: `./DigimatInstall` (see Figure 2-38)
- Step 3: If accept license agreement, type 1 (see Figure 2-39)
- Step 4: If accept the general conditions, type 1 (see Figure 2-40)
- Step 5: Select Digimat installation directory (see Figure 2-41)
- Step 6: Digimat installation in progress (see Figure 2-42)
- Step 7: Introduce Digimat license adress (see Figure 2-43)
- Step 8: End of Digimat installation (see Figure 2-44)

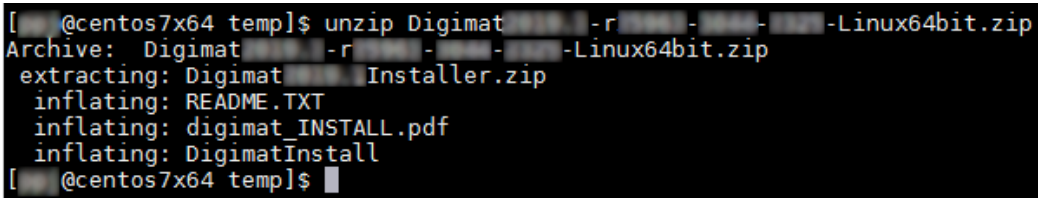


Figure 2-37 Unzip installation file.

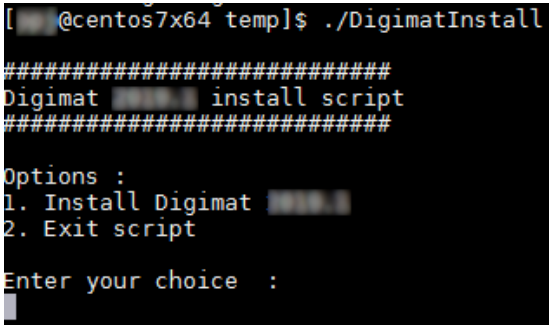


Figure 2-38 Execute installation script.

```
Please read the following license agreement.
You must accept the terms of this agreement before continuing with the installation.

Options :
1. Continue
2. Exit

Enter your choice  :
1
```

Figure 2-39 License agreement.

```
14.5 If any provision of this Agreement is invalid, the parties agree that suc
ment.  The parties further agree to substitute a valid provision for the inval
e invalid provision.
14.6 Ambiguities, inconsistencies, or conflicts in this Agreement, will not be
esolved by applying the most reasonable interpretation under the circumstances
acting.  The section headings in this Agreement are for convenience only and w
14.7 MSC will not be liable for any loss, damage or penalty resulting from del
r causes beyond MSC's reasonable control.
14.8 All notices will be in writing.  Notices permitted or required under this
d or registered mail, return receipt requested, or by confirmed facsimile tran
 shall be directed to the attention of the Legal Department.
14.9 Customer acknowledges and agrees that any and all consulting services per
ase and use of the Software licenses. Customer further agrees that payment for
d with the commencement, completion or delivery of consulting services.
14.10 The English language version of this Agreement is legally binding in cas

--------------------------------------------------------------------------------


Do you accept the agreement ?
Options :
1. Yes (Continue)
2. No (Exit)

1
```

Figure 2-40 General conditions.

```
Digimat [redacted] installation requires [redacted] of disk space

Write permissions are required in target installation directory

Enter target installation directory (example : /msc). A directory /Digimat/[redacted] will
 be automatically created :
```

Figure 2-41 Select Digimat installation directory.

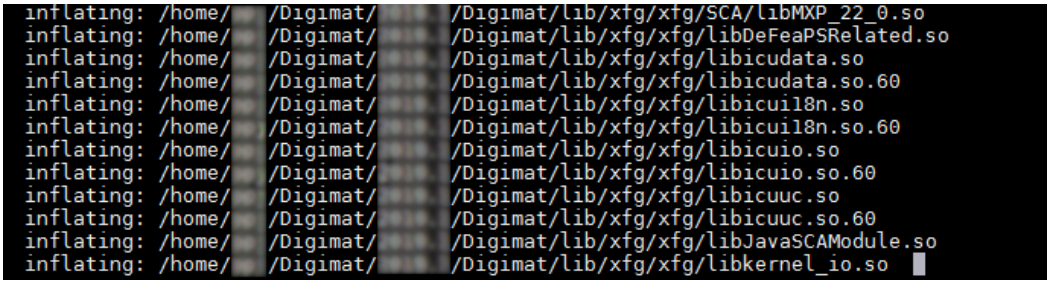


Figure 2-42 Digimat installation progress.

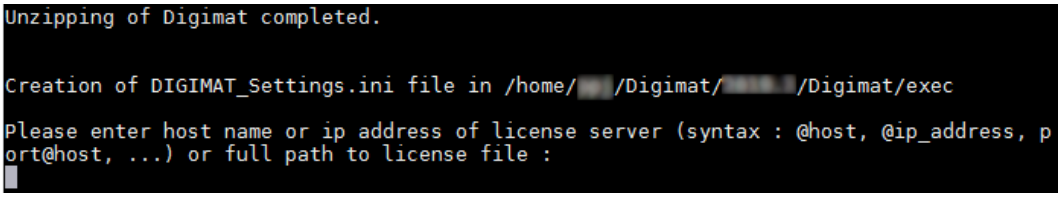


Figure 2-43 License server address.

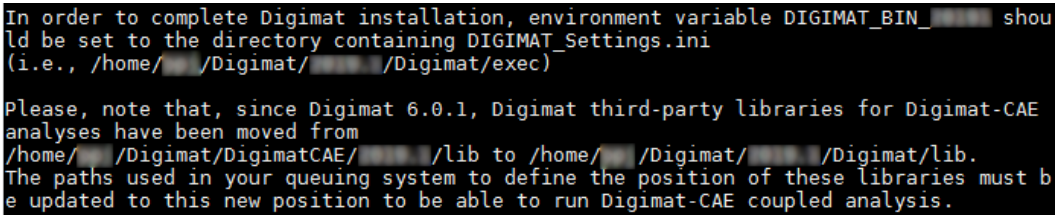


Figure 2-44 End of Digimat installation.

Digimat Settings

MSC_LICENSE_FILE Environment Variable

Under Windows platform, MSC_LICENSE_FILE environment variable is defined at Digimat installation as global environment variable. This environment variable is used to define path to the license server(s). If an initial value exists, Digimat installation will propose this value.

If MSC_LICENSE_FILE is not defined as environment variable, no Digimat product can run.

The licensing location defined by MSC_LICENSE_FILE can be directly checked by using the **License** option of the Digimat platform (see Figure 2-45). Clicking on **Query status** will give a status of license server defined in MSC_LICENSE_FILE (number of available/used licenses...).

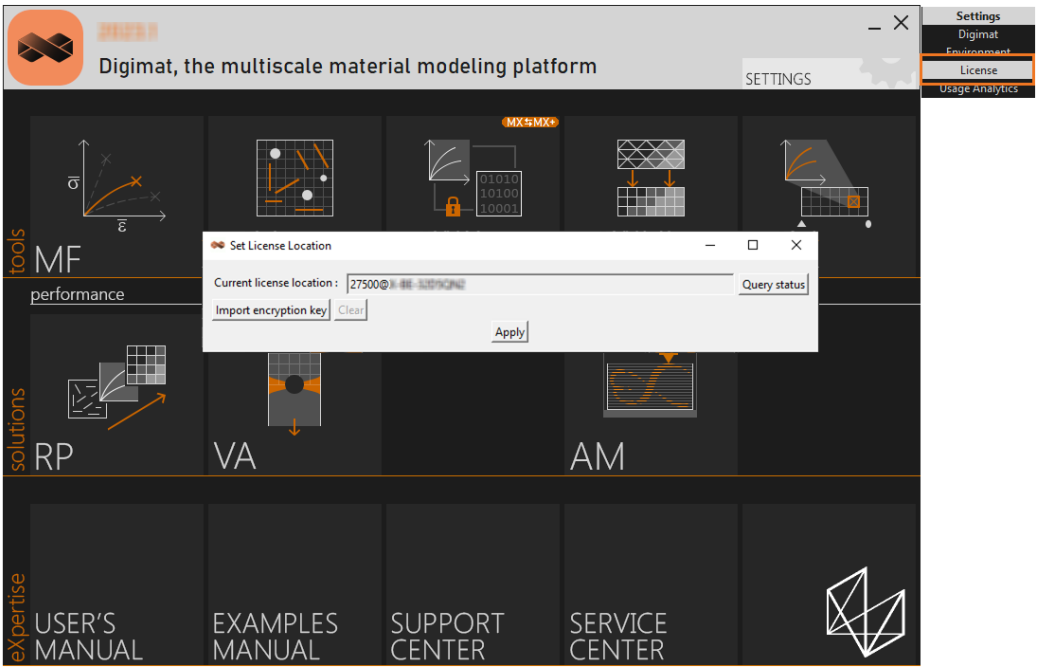


Figure 2-45 Checking the licensing location via the Digimat platform.

DIGIMAT_BIN_20231 Environment Variable

Under Windows platform, DIGIMAT_BIN_20231 environment variable is defined at Digimat installation as a global environment variable. This environment variable is used to define directory of DIGIMAT_Settings.ini file (see DIGIMAT_Settings.ini File). After installation, DIGIMAT_BIN_20231 points initially to directory:

```
DIGIMAT_INSTALL_DIR\Digimat\exec
```

with DIGIMAT_INSTALL_DIR is the Digimat 2023.1 installation directory, e.g.,

```
C:\MSC.Software\Digimat\2023.1
```

If this value is not defined, it will be defined by Digimat product launching scripts (Digimat platform, Digimat-MF GUI...). Global environment variable always override the one defined in Digimat product launching scripts.

Value of `DIGIMAT_BIN_20231` can be checked from the setting of Digimat platform and then changed to point to another `DIGIMAT_Settings.ini` file (see Figure 2-46). This change can be done only in administrator mode.

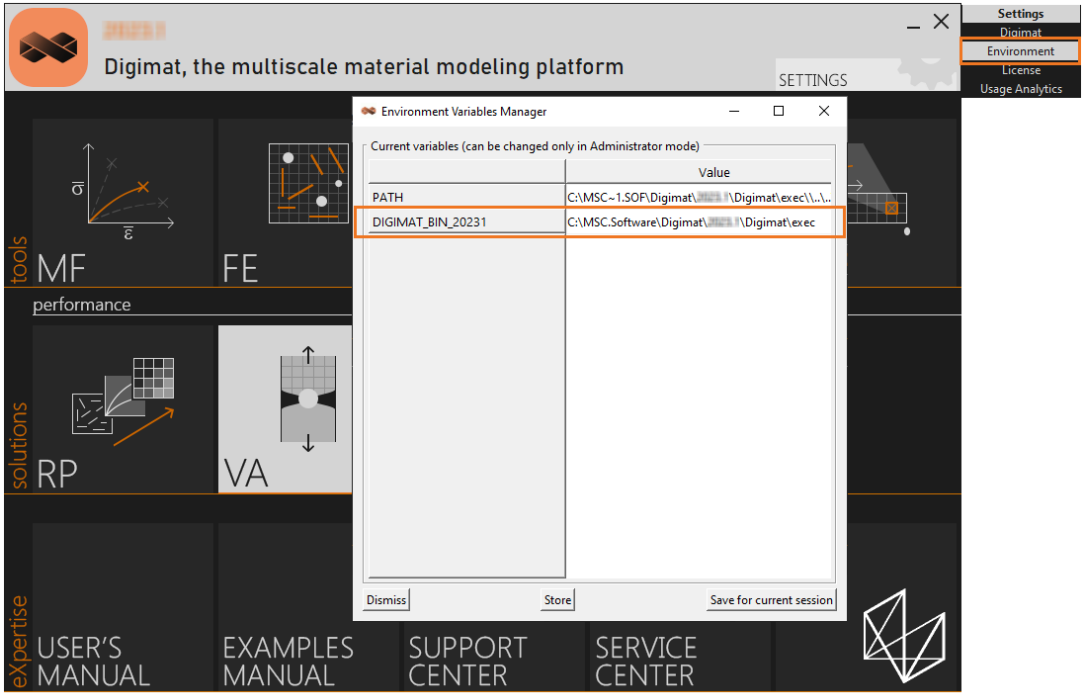


Figure 2-46 Definition of Digimat environment variables.

Remarks:

1. When using Digimat-CAE plugins (Abaqus, ANSYS and Marc Mentat), it is mandatory to define the global `DIGIMAT_BIN_20231` environment variable as pointing to a valid directory containing `DIGIMAT_Settings.ini` file, e.g.,

   ```
   C:\MSC.Software\Digimat\2023.1\Digimat\exec
   ```

2. Under Linux platform, the `DIGIMAT_BIN_20231` environment variable must always be set manually to point on a valid `DIGIMAT_Settings.ini` file.

During Digimat installation, path to the shortcuts folder, containing the file that launches the Digimat platform is added to the system environment variable PATH.

Additional Environment Variables for Network Installations of Digimat

See section Install Digimat on a Network Windows Machine to get all information relative to network installation settings.

DIGIMAT_Settings.ini File

Digimat settings are set by the `DIGIMAT_Settings.ini` file which contains a list of key-values. This file is read by Digimat each time the platform is launched to let them become the current settings. The entire Digimat installation can be parameterized by these settings.

Digimat looks in the following directories for the `DIGIMAT_Settings.ini` file:

1. in the current working directory, i.e., the directory in which the computation is run;
2. in the directory pointed to by the environment variable `DIGIMAT_BIN_20231`.

Note that Digimat will use the first `DIGIMAT_Settings.ini` file that it finds! Since it first looks in the current working directory, it allows using a local settings file, and if none is defined, it is not problematic as long as it can find the global settings file defined in the folder where the `DIGIMAT_BIN_20231` variable is pointing to.

The `DIGIMAT_Settings.ini` file can be modified from the platform settings menu (see Figure 2-47) or directly via a text editor.

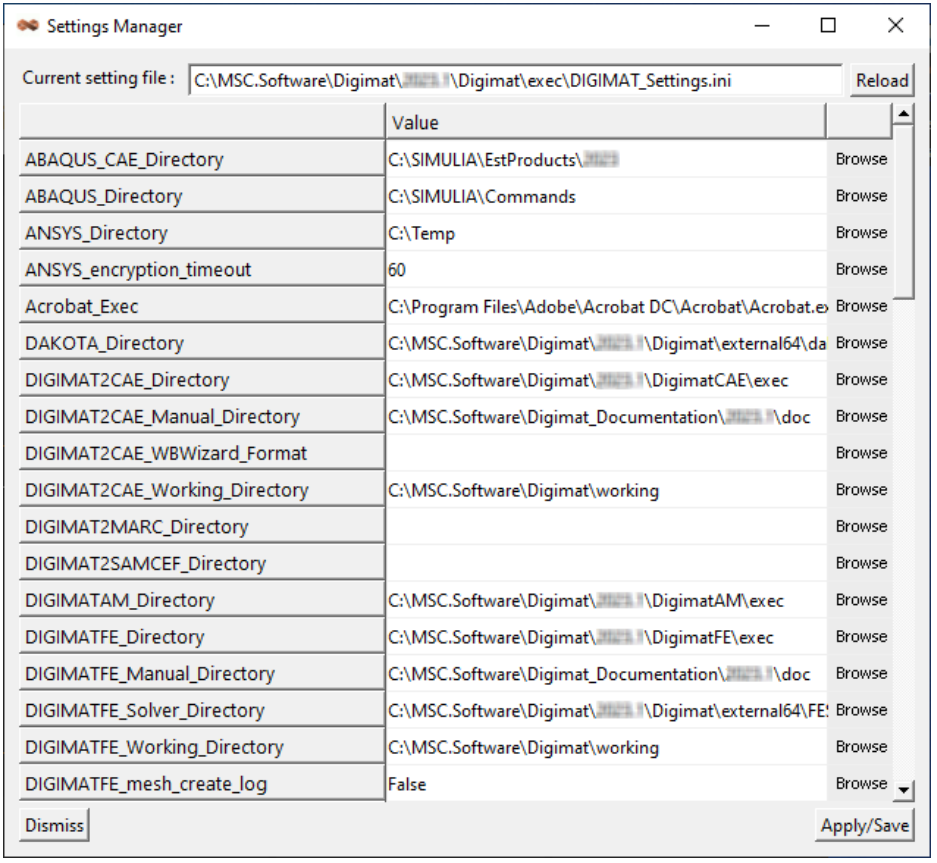


Figure 2-47 Definition of Digimat settings.

Structure of the DIGIMAT_Settings.ini File

The `DIGIMAT_Settings.ini` file is made of sections delimited by a line '[`SectionKeyWord`]', each section containing a list of lines '`key = value`'.

For the current version, this file is made of a unique section which is identified by the [`Default`] tag as a header to the file content. The list of keys that can be used is the following:

- `ABAQUS_CAE_Directory`: path to the Abaqus CAE directory.
- `ABAQUS_Directory`: path to the Abaqus root directory.
- `ANSYS_Directory`: path to the ANSYS root directory.
- `ANSYS_encryption_timeout`: relates to the encryption of material properties when using Macro solution in Digimat-RP for ANSYS. Number of seconds between the beginning of the ANSYS-service startup process, and its forced termination by Digimat. This entry is generated during Digimat installation, with a default value of 60. It should be defined according to the ANSYS licence timeout settings.

- `Acrobat_Exec`: path to Adobe Reader executable, including its name.
- `DAKOTA_Directory`: path to Dakota binaries.
- `DIGIMAT2CAE_Directory`: path to the Digimat-CAE binaries.
- `DIGIMAT2CAE_Manual_Directory`: path to the Digimat-CAE documentation. It is used by Digimat when opening the documentation (general or context help page) from the GUI.
- `DIGIMAT2CAE_WBWizard_Format`: Deprecated setting
- `DIGIMAT2CAE_Working_Directory`: path to the working directory used by Digimat-CAE.
- `DIGIMAT2MARC_Directory`: path to the directory containing Digimat-CAE/Marc executable (see Digimat-CAE/Marc).
- `DIGIMAT2SAMCEF_Directory`: path to the directory containing Digimat-CAE/Samcef executable (see Digimat-CAE/Samcef).
- `DIGIMATAM_Directory`: path to the Digimat-AM binaries.
- `DIGIMATFE_Directory`: path to the Digimat-FE binaries.
- `DIGIMATFE_Manual_Directory`: path to the Digimat-FE documentation. It is used by Digimat when opening the documentation (general or context help page) from the GUI.
- `DIGIMATFE_Solver_Directory`: path to Digimat-FE solver directory.
- `DIGIMATFE_Working_Directory`: path to the working directory used by Digimat-FE.
- `DIGIMATFE_mesh_create_log`: In Digimat-FE analysis, write a log file containing CPU information for mesh generation. Must be True of False (default: False).
- `DIGIMATMF_Directory`: path to the Digimat-MF binaries.
- `DIGIMATMF_Manual_Directory`: path to the Digimat-MF documentation. It is used by Digimat when opening the documentation (general or context help page) from the GUI.
- `DIGIMATMF_Output_Precision`: Number of significant number for Digimat-MF outputs
- `DIGIMATMF_Working_Directory`: path to the working directory used by Digimat-MF.
- `DIGIMATMX_Bin_Directory`: path to PostgreSQL binaries.
- `DIGIMATMX_Crypt`
- `DIGIMATMX_Crypt_ForAll`: Decryption key allowing to use all public encrypted materials
- `DIGIMATMX_Crypt_ForDomo`
- `DIGIMATMX_Database_Cluster`: list of the locations (localhost and server IP addresses) of servers on which a successful connection has been made. The first listed value is the location of the server on which the last successful connection occurred and it is the default value used by Digimat-MX. If no successful connection ever occurred, localhost is the only value.
- `DIGIMATMX_DataBase_Name`: Digimat-MX default database to be opened. The default value is mxdb.
- `DIGIMATMX_Directory`: path to the Digimat-MX binaries.
- `DIGIMATMX_Local_Installation`: true of false (see Reverse Engineering on Client Machine).
- `DIGIMATMX_Manual_Directory`: path to the Digimat-MX documentation. It is used by Digimat when opening the documentation (general or context help page) from the GUI.

- DIGIMATMX_Request_Data_Format: length of the content of the email which is automatically created when requesting data to a material supplier from Digimat-MX. The two possible values are Long (which is the default) and Short (which is usually required when using Lotus Notes as e-mail client).
- DIGIMATMX_User: Digimat-MX user name.
- DIGIMATMX_User_ID
- DIGIMATMX_Working_Directory: path to the working directory used by Digimat-MX.
- DIGIMATRP_Directory: path to the Digimat-RP binaries.
- DIGIMATVA_Directory: path to the Digimat-VA binaries.
- DIGIMAT_Directory: path to the working directory used by the Digimat platform.
- DIGIMAT_Analytics_Directory: path to the directory used to store Usage Analytics preferences.
- Examples_Directory: path to the Digimat examples directory used by Digimat when opening the documentation (general or context help page) from the GUI.
- GnuPG_Directory: path to GnuPG directory. Needed to encrypt material properties when using Macro solution in Digimat-RP for LS-DYNA.
- HYPERWORKS_Directory: path to the HyperWorks root directory.
- INTELMPI_Directory: Intel-MPI used for Digimat-FE FFT computations.
- LS-Dyna_SMP_Exec: path to LS-DYNA executable used by Digimat-FE.
- LS-DYNA_Directory: path to the directory containing Digimat-CAE/LS-DYNA executable (see Digimat-CAE/LS-DYNA).
- LSPREPOST_Directory: path to the LS-PrePost root directory.
- LSTC_PGPKEY: Full path and name of key file needed to encrypt material properties when using Macro solution in Digimat-RP for LS-DYNA. Default location value is Digimat working directory and default name is lstc_pgpkey.asc. This entry is generated during Digimat installation. The way to generate this key is explained in Generation of Encryption Key File for Macro Solution.
- LocalDatabase_Directory: path to the built-in local database.
- LocalDatabase_SQLite_Directory: path to the Digimat-VA database.
- log_output: path to the location where Digimat will output its log messages. This is one of the keys a user could most probably be brought to change. The different choices are:
  - Default: the Digimat messages will be output to the default location which means, for example:
    - in the .log file of the analysis/job if using Digimat-MF or a Digimat-CAE interface;
    - the dos screen when using the interface to PAM-CRASH, ...
  - Any valid path to a file, including its name. The log messages will be output to the indicated file.
- MAP_Directory: path to the Digimat-MAP binaries.
- MAP_Manual_Directory: path to the Digimat-MAP documentation. It is

- `MAP_Working_Directory`: path to the working directory used by Digimat-MAP.
- `MARC_Directory`: path to the Marc root directory.
- `MENTAT_Directory`: path to the Marc Mentat root directory.
- `MSCNASTRAN_Directory`: path to the MSC Nastran root directory.
- `Manual_Directory`: path to the Digimat documentation.
- `Number_of_processors`: Number of processors of computer where Digimat is installed.
- `OPTISTRUCT_Directory`: path to the OptiStruct root directory.
- `PAMCRASH_Directory`: path to the PAM-CRASH root directory.
- `PATRAN_Directory`: path to the Patran root directory.
- `PCMPI_Directory`: path to Platform-MPI directory.
- `PERMAS_Directory`: path to the PERMAS root directory.
- `POSTGRESQL_PORT`: Port for access to remote database.
- `RADIOSS_Directory`: Deprecated.
- `SAMCEF_Directory`: path to the Samcef root directory.
- `Specific_features`: list of opened specific features (see chapter Specific Features in *Getting Started Guide*).
- `VISUALVIEWER_Directory`: path to the Visual-Viewer root directory.
- `WISETEX_Directory`: path to Wisetex binaries.
- `Working_Directory`: path to the working directory used by Digimat.

All these parameters can be set from the Digimat platform settings manager (see Figure 2-47).

An example of a valid `DIGIMAT_Settings.ini` file is:

```
[Default]
ABAQUS_CAE_Directory=C:\SIMULIA\EstProducts\2023
ABAQUS_Directory=C:\SIMULIA\Commands
ANSYS_Directory=C:\Program Files\ANSYS Inc\v231
ANSYS_encryption_timeout=60
Acrobat_Exec=C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader\AcroRd32.exe
DAKOTA_Directory=C:\MSC.Software\Digimat\2023.1\Digimat\external64\dakota
DIGIMAT2CAE_Directory=C:\MSC.Software\Digimat\2023.1\DigimatCAE\exec
DIGIMAT2CAE_Manual_Directory=C:\MSC.Software\Digimat_Documentation\2023.1\doc
DIGIMAT2CAE_WBWizard_Format=
DIGIMAT2CAE_Working_Directory=C:\MSC.Software\Digimat\working
DIGIMAT2MARC_Directory=C:\MSC.Software\Digimat\2023.1\DigimatCAE\exec\digi2Marc\2022.4\INTELMPI
DIGIMATAM_Directory=C:\MSC.Software\Digimat\2023.1\DigimatAM\exec
DIGIMATFE_Directory=C:\MSC.Software\Digimat\2023.1\DigimatFE\exec
DIGIMATFE_Manual_Directory=C:\MSC.Software\Digimat_Documentation\2023.1\doc
DIGIMATFE_Solver_Directory=C:\MSC.Software\Digimat\2023.1\Digimat\external64\FESolver
DIGIMATFE_Working_Directory=C:\MSC.Software\Digimat\working
DIGIMATFE_mesh_create_log=False
DIGIMATMF_Directory=C:\MSC.Software\Digimat\2023.1\DigimatMF\exec
DIGIMATMF_Manual_Directory=C:\MSC.Software\Digimat_Documentation\2023.1\doc
DIGIMATMF_Output_Precision=
DIGIMATMF_Working_Directory=C:\MSC.Software\Digimat\working
DIGIMATMX_Bin_Directory=C:\MSC.Software\Digimat\2023.1\DigimatMX\bin
DIGIMATMX_Crypt=
DIGIMATMX_Crypt_ForAll=3082037D020100300D06092A864886F70D01010105000482...
DIGIMATMX_Crypt_ForDomo=
DIGIMATMX_Database_Cluster=localhost
DIGIMATMX_Database_Name=
DIGIMATMX_Directory=C:\MSC.Software\Digimat\2023.1\DigimatMX\exec
```

```
DIGIMATMX_Local_Installation=false
DIGIMATMX_Manual_Directory=C:\MSC.Software\Digimat_Documentation\2023.1\doc
DIGIMATMX_Request_Data_Format=Long
DIGIMATMX_User=manager
DIGIMATMX_User_ID=
DIGIMATMX_Working_Directory=C:\MSC.Software\Digimat\working
DIGIMATRP_Directory=C:\MSC.Software\Digimat\2023.1\DigimatRP\exec
DIGIMATVA_Directory=C:\MSC.Software\Digimat\2023.1\DigimatVA\exec
DIGIMAT_Directory=C:\MSC.Software\Digimat\2023.1\Digimat\exec
Examples_Directory=C:\MSC.Software\Digimat_Documentation\2023.1\examples
GnuPG_Directory=C:\Program Files (x86)\gnupg\bin
HYPERWORKS_Directory=C:\Program Files\Altair\2018
INTELMPI_Directory=C:\MSC.Software\Digimat\2023.1\Digimat\external64\intelmpi
LS-Dyna_SMP_Exec=C:\LS-Dyna\ls-dyna_smp_d_R13.1_138-g8429c8a10f_winx64_ifort190.exe
LS-DYNA_Directory=C:\MSC.Software\Digimat\2023.1\DigimatCAE\exec\digi2Dyna\R13.1.1\Double\PCMPI
LSPREPOST_Directory=C:\Program Files\LSTC\LS-PrePost 4.8
LSTC_PGPKEY=C:\MSC.Software\Digimat\working\lstc_pgpkey.asc
LocalDatabase_Directory=C:\MSC.Software\Digimat_LocalDatabases\2023.1\postgresql
LocalDatabase_SQLite_Directory=C:\MSC.Software\Digimat_LocalDatabases\2023.1\sqlite
Log_output=Default
MAP_Directory=C:\MSC.Software\Digimat\2023.1\DigimatMAP\exec
MAP_Manual_Directory=C:\MSC.Software\Digimat_Documentation\2023.1\doc
MAP_Working_Directory=C:\MSC.Software\Digimat\working
MARC_Directory=C:\Program Files\MSC.Software\Marc\2022.4.0\marc2022.4
MENTAT_Directory=C:\Program Files\MSC.Software\Marc\2022.4.0\mentat2022.4
MSCNASTRAN_Directory=C:\Program Files\MSC.Software\MSC_Nastran\20224
Manual_Directory=C:\MSC.Software\Digimat_Documentation\2023.1\doc
Number_of_processors=8
OPTISTRUCT_Directory=C:\Program Files\Altair\2018
PAMCRASH_Directory=C:\Program Files\ESI Group\Virtual-Performance\2022.0
PATRAN_Directory=C:\Program Files\MSC.Software\Patran_x64\20224
PCMPI_Directory=
PERMAS_Directory=C:\Program Files\INTES\V17
POSTGRESQL_PORT=
RADIOSS_Directory=C:\Program Files\Altair\2018
SAMCEF_Directory=C:\Samtech\Samcef\V17.2_x64_i8
Specific_features=Default
VISUALVIEWER_Directory=C:\Program Files\ESI Group\Visual-Environment\18.5
WISETEX_Directory=C:\MSC.Software\Digimat\2023.1\Digimat\external32\wisetex
Working_Directory=C:\MSC.Software\Digimat\working
```

Remarks:

- The listing order of the keywords does not matter.
- Trailing or leading white spaces, tabulations, or quotations marks (') are removed from the values.
- The initial keyword [Default] must be defined.
- A path including directories with long names can be written in short notation, e.g.,

  `C:\Progra1\Digimat.`

Usage Analytics Settings

The **Hexagon Customer Experience Improvement Program** (**"CEIP"**) is a voluntary program designed to help Hexagon improve its products and services. The CEIP collects information about your computer system and how you use our software. This can be enabled or disabled in the **Usage Analytics** option of the Digimat platform (see Figure 2-48). Enabling or disabling this option takes effect immediately for the current session. Other currently opened Digimat products will need to be restarted for the update to be taken into account. More information about the CEIP can be found at Hexagon CEIP Documentation.

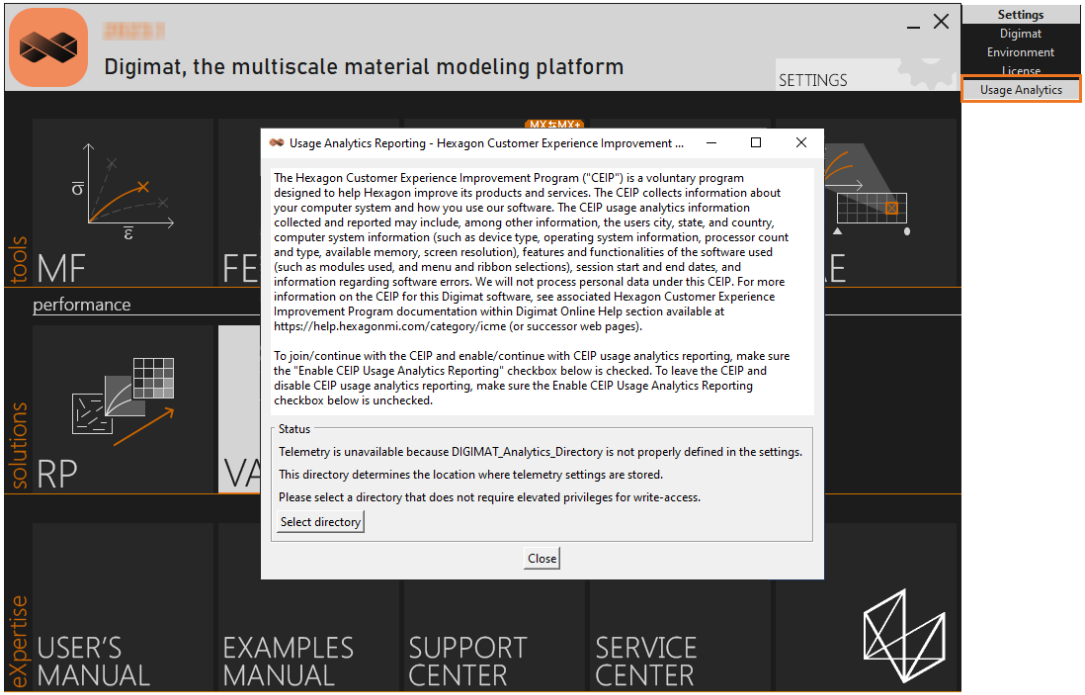


Figure 2-48 Usage Analytics settings.

Remarks:

- This preference is stored at a location specified by the `DIGIMAT_Analytics_Directory` key in the DIGIMAT_Settings.ini File.
- Write-access to the location specified by the `DIGIMAT_Analytics_Directory` is required to store this preference.

3 Installing Digimat Examples Manual

- Digimat Examples Manual

Digimat Examples Manual

It is possible to install Digimat Examples Manual before, after or automatically during Digimat software installation. It is also possible to use existing Digimat Examples Manual when installing Digimat software (see Figure 2-13). When installing Digimat Examples Manual separately from main Digimat software, the following step by step procedure must be followed.

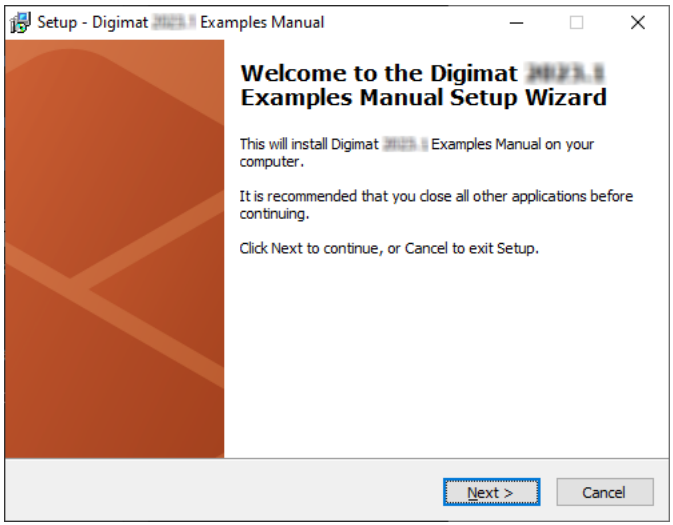


Figure 3-1 Opening prompt of the Digimat Examples Manual installer. Follow the given instructions step-by-step.

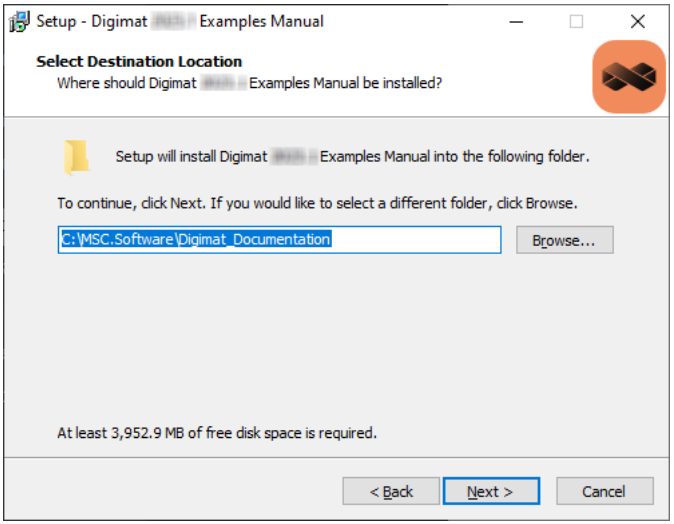


Figure 3-2 The destination folder for the Digimat Examples Manual installation has to be given. A 2023.1 directory will be automatically created.

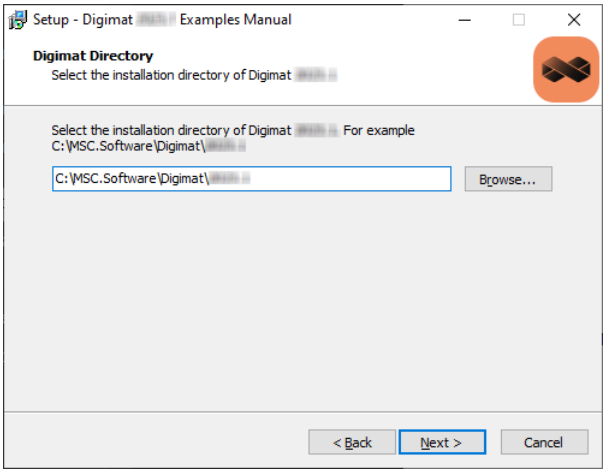


Figure 3-3 User has to point to an existing Digimat directory. So the settings of Digimat will be automatically updated to point to this Examples Manual directory. This field can remain blank if Digimat will be installed in a second step. In that case, when installing Digimat software, user has to select this Digimat Examples Manual installation directory (see Figure 2-13). Note that user cannot point to a Digimat prior to Digimat 2016.0.

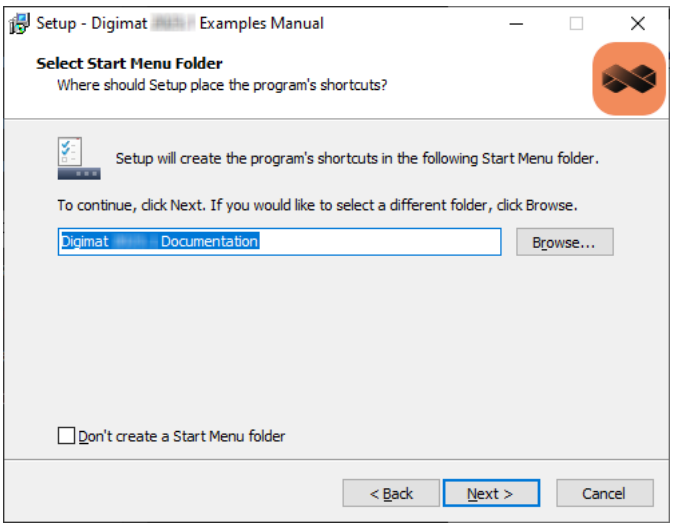


Figure 3-4 An individual name for the Digimat documentation shortcut can be specified.

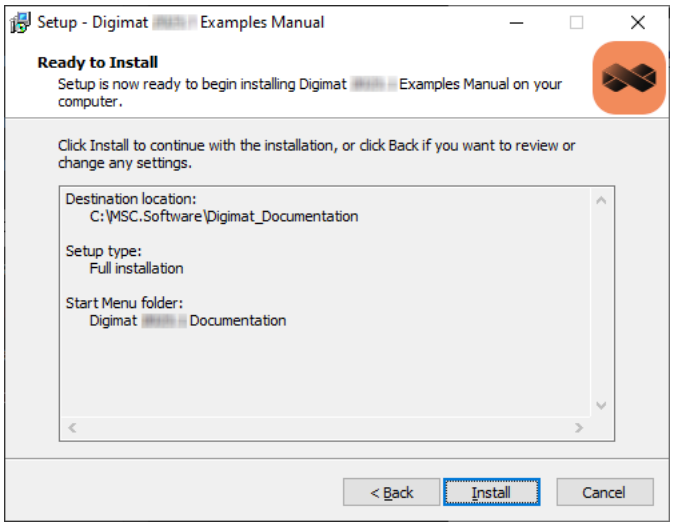


Figure 3-5 A summary of the installation details is given and can be checked. Proceeding with the Install button will start the installation of Digimat Examples Manual.

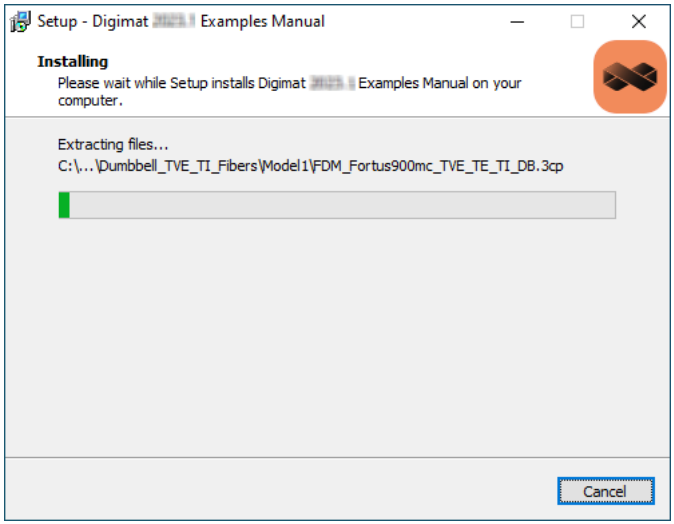


Figure 3-6 The installation of the Digimat Examples Manual is executed.

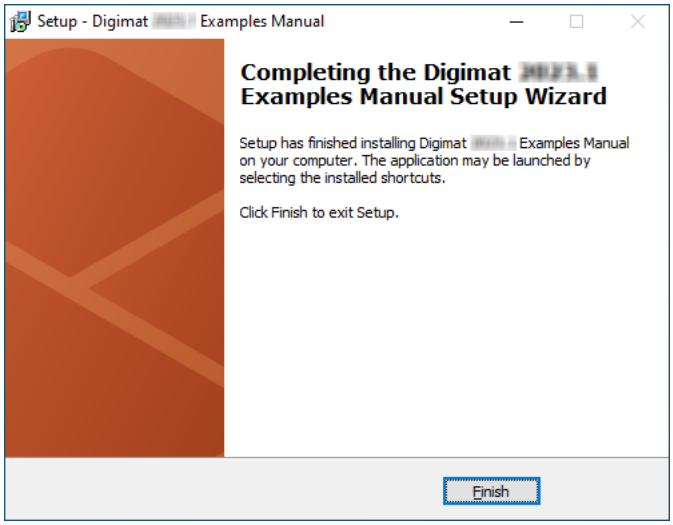


Figure 3-7 This concludes the installation.

4 CAE Interfaces

- Digimat-CAE/Generalities
- Digimat-CAE/Abaqus
- Digimat-CAE/ANSYS
- Digimat-CAE/Marc
- Digimat-CAE/MSC Nastran SOL400
- Digimat-CAE/Samcef
- Digimat-CAE/LS-DYNA
- Digimat-CAE/PAM-CRASH
- Digimat-CAE/MSC Nastran SOL1XX
- Digimat-CAE/OptiStruct
- Digimat-CAE/PERMAS
- Digimat-CAE/nCode DesignLife

Digimat-CAE/Generalities

This section describes common settings to be able to run Digimat-CAE simulations. They apply to all supported interfaces.

In case of parallel computation, one structural feature seat is first used, and for the other n-1 processors used for the parallel computation, n-1 Digimat-CAE parallel tokens are checked out. For example, a Digimat-CAE/Marc parallel run on 4 processors consumes 1 seat of a structural feature as well as 3 Digimat-CAE parallel tokens.

When performing parallel computations using distributed storage devices, Digimat requires definition of `DIGIMAT2CAE_Working_Directory` setting in DIGIMAT_Settings.ini File. This setting has to indicate a global, unique location which can be accessed by all the processes. **If this setting is not defined, it can result in an overconsumption of license features! This location must be specified on a NFS file system. Other file system, like BeeGFS, are not supported and may lead also to an overconsumption of license features.**

Linking with CAE Software

- Linking Digimat with the external CAE software has to be carefully set up and tested.
- Please note that depending on the solver type and platform special linker software might be required.
- The detailed linking procedure for each CAE solver is described in detail in this chapter
    - Please refer to the individual section of the required CAE code below.

Digimat-CAE/Abaqus

The objective of this section is to explain how to install the interface between Digimat and Abaqus. For more information concerning supported Abaqus releases and platforms, please refer to the below section Supported Versions.

Supported Versions

Digimat 2023.1 supports following Abaqus versions:

- Abaqus 2021 GA
- Abaqus 2022 GA
- Abaqus 2023 GA

Abaqus HotFix version are not officially supported. Please contact digimat.support@hexagon.com in case of issue with HotFix. All Abaqus 201X-EFy versions or similar are not supported.

Digimat 2023.1 supports following platforms for Abaqus interface:

- Windows 10 (64-bit)
- Windows Server 2019 (64-bit)
- Linux Red Hat 7.9 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux Red Hat 8.4 using GLIBC ≥ 2.28 and GLIBCXX ≥ 3.4.25
- Linux SUSE 12 SP1 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP2 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP4 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 15 SP1 using GLIBC ≥ 2.26 and GLIBCXX ≥ 3.4.28

Digimat 2023.1 supports following parallelization methods:

- For Windows platforms
  - Shared Memory Parallelization (SMP)
  - Distributed Memory Parallelization (DMP) using MS-MPI (default in Abaqus)
- For Linux platforms
  - Shared Memory Parallelization (SMP)
  - Distributed Memory Parallelization (DMP) using Platform-MPI (default in Abaqus Standard)
  - Mixed SMP/DMP computation using Platform-MPI (default in Abaqus Standard)

Digimat 2023.1 supports following Abaqus solutions:

- Abaqus Standard
- Abaqus Explicit single precision
- Abaqus Explicit double precision

Installation Procedure

Digimat-CAE/Abaqus is provided as a set of dynamic libraries. These libraries contain Digimat capabilities, allowing the use of Digimat materials in Abaqus analyses. This link is performed on both Linux and Windows operating systems.

Dynamic libraries are version dependent. For example, this means that libraries provided for Abaqus 2023 cannot be used with previous Abaqus versions.

Dynamic libraries are located in the directory

```
DIGIMAT_DIR\DigimatCAE\exec\digi2aba
```

of the Digimat installation. `DIGIMAT_DIR` is Digimat installation directory, e.g.,

- `C:\MSC.Software\Digimat\2023.1` (Windows)
- `/opt/software/Digimat/2023.1` (Linux)

This directory contains subfolders, each subfolder corresponding to a given supported Abaqus versions. Each of these folders contains three dynamic libraries:

- One single precision library for Abaqus/Explicit:
  - `explicitU.dll` (Windows)
  - `libexplicitU.so` (Linux)
- One double precision library for Abaqus/Explicit:
  - `explicitU-D.dll` (Windows)
  - `libexplicitU-D.so` (Linux)
- One double precision library for Abaqus/Standard:
  - `standardU.dll` (Windows)
  - `libstandardU.so` (Linux)

`digi2aba` directory also contains dynamic libraries for Digimat kernel (those libraries are common for all Abaqus versions):

- One single precision library for Abaqus/Explicit:
  - `digi2abaExp.dll` (Windows)
  - `libdigi2abaExp.so` (Linux)
- One double precision library for Abaqus/Explicit:
  - `digi2abaExpDouble.dll` (Windows)
  - `libdigi2abaExpDouble.so` (Linux)
- One double precision library for Abaqus/Standard:
  - `digi2abaStd.dll` (Windows)
  - `libdigi2abaStd.so` (Linux)

Finally, third-party components

- `boost_chrono-mt-x64.dll`
- `boost_filesystem-mt-x64.dll`

- boost_iostreams-mt-x64.dll
- boost_regex-mt-x64.dll
- boost_system-mt-x64.dll
- boost_thread-mt-x64.dll
- boost_zlib-mt-x64.dll
- digimatMathTools.dll
- digimatPocoFoundation.dll
- lapi.dll
- VMAP.dll

are located in digi2aba directory under Windows and in Digimat/lib directory under Linux. If the location of the dynamic libraries is changed, usub_lib_dir variable defined in abaqus_v6.env file must be changed accordingly (see below).

Abaqus Environment File abaqus_v6.env

To make communication between Abaqus and Digimat, an environment file abaqus_v6.env is provided with Digimat installation for each supported Abaqus version. This file depends on:

- targeted Abaqus version.
- parallelization that will be used:
  - SMP
  - DMP

For example, if Abaqus 2023 with DMP parallelization is targeted, it is needed to use abaqus_v6.env files located in directory:

```
DIGIMAT_DIR/DigimatCAE/exec/digi2aba/2023/DMP
```

where DIGIMAT_DIR is Digimat installation directory, e.g.,

```
C:\MSC.Software\Digimat\2023.1.
```

Abaqus environment file abaqus_v6.env for Windows platforms

If DIGIMAT_DIR is Digimat installation directory, e.g., C:\MSC.Software\Digimat\2023.1, and if Abaqus targeted version is 2023, abaqus_v6.env environment file provided in Digimat installation contains following information:

- Path to Digimat-CAE/Abaqus libraries and its dependencies:

```
usub_lib_dir=r"DIGIMAT_DIR\DigimatCAE\exec\digi2Aba\2023"
usub_lib_dir+="DIGIMAT_INST_DIR\DigimatCAE\exec\digi2Aba"
```

- Path to plugin directory:

```
plugin_central_dir=r"DIGIMAT_DIR\DigimatCAE\exec\digi2Aba\abaqus_plugins"
```

- Abaqus version:

```
os.environ["ABAQUS_VERSION"] = "2023"
```

- Parallelization method information:

  For SMP computations:

  ```
  mp_mode=THREADS
  ```

  For DMP computations:

  ```
  cpus = globals().get("cpus", locals().get("cpus", 1))
  standard_parallel=ALL
  os.environ["ABA_MPI_SKIP_BUNCH_NODES"] = "1"
  mp_host_list = [[socket.gethostname(), 1]]*cpus
  mp_mode=MPI
  ```

  Environment variable `ABA_MPI_SKIP_BUNCH_NODES` is needed to force pure DMP computations.

More information about the Abaqus environment file are provided in Abaqus documentation (Environment file settings section).

Abaqus environment file abaqus_v6.env for Linux platforms

If `DIGIMAT_DIR` is Digimat installation directory, e.g., `/opt/msc/Digimat/2023.1` and if Abaqus targeted version is 2023, `abaqus_v6.env` environment file provided in Digimat installation contains following information:

- Path to Digimat-CAE/Abaqus libraries and its dependencies:

  ```
  usub_lib_dir='DIGIMAT_DIR/DigimatCAE/exec/digi2aba'
  usub_lib_dir+='DIGIMAT_DIR/DigimatCAE/exec/digi2aba/2023'
  usub_lib_dir+='DIGIMAT_DIR/Digimat/lib'
  ```

- Abaqus version:

  ```
  os.environ['ABAQUS_VERSION'] = '2023'
  ```

- Environment variables to passed to Abaqus computations:

  ```
  os.environ['DIGIMAT_BIN_20231']='DIGIMAT_DIR/Digimat/exec'
  os.environ['MSC_LICENSE_FILE']='27500@hostname'
  os.environ['FLEXLM_TIMEOUT']='5000000'
  mp_environment_export =
  tuple(list(mp_environment_export)+['DIGIMAT_BIN_20231']+
  ['MSC_LICENSE_FILE']+['FLEXLM_TIMEOUT'])
  ```

  Definition of `FLEXLM_TIMEOUT` environment variable can be needed when using geographically distant license server to avoid failing of license checkout if answer of network is too slow. A meaningful value for `FLEXLM_TIMEOUT` is 5000000.

- Parallelization method information:
  - For SMP computations:

    ```
    standard_parallel=ALL
    ```

```
mp_mode=THREADS
cpus=globals().get('cpus',locals().get('cpus',1))
mp_host_list=[]
```

- For DMP computations:

```
os.environ['ABA_MPI_SKIP_BUNCH_NODES'] = '1'
standard_parallel=ALL
mp_mode=MPI
cpus=globals().get('cpus',locals().get('cpus',1))
mp_host_list=[]
```

Environment variable `ABA_MPI_SKIP_BUNCH_NODES` is needed to force pure DMP computations.

Environment variable `mp_host_list=[]` must be completed with list of hosts where Digimat-CAE/Abaqus computation will run, e.g., to run on 2 nodes named `node1` and `node2` and 4 CPUs on each node:

```
mp_host_list = [['node1', 4], ['node2', 4]]
```

- For mixed SMP/DMP computations:

```
standard_parallel=ALL
mp_mode=MPI
cpus=globals().get('cpus',locals().get('cpus',1))
mp_host_list=[]
```

Environment variable `mp_host_list=[]` must be completed with list of hosts where Digimat-CAE/Abaqus computation will run, e.g., to run on 2 nodes named `node1` and `node2` and 4 CPUs on each node:

```
mp_host_list = [['node1', 4], ['node2', 4]]
```

More information about the Abaqus environment file are provided in Abaqus documentation (Environment file settings section).

Running Coupled Digimat-CAE/Abaqus Analysis

Windows platforms

To run Digimat to Abaqus coupled analysis using command line, it is needed to copy the `abaqus_v6.env` of targeted Abaqus version and parallelization method (SMP or DMP) from Digimat installation directory to working directory. Once it is done, following script example can be used:

```
set FLEXLM_TIMEOUT=5000000
set MSC_LICENSE_FILE=27500@localhost
C:\Simulia\Commands\abaqus.bat job=test.inp cpus=N
```

First line is needed when using geographically distant license server to avoid failing of license checkout if answer of network is too slow. A meaningful value for `FLEXLM_TIMEOUT` is 5000000. Definition of

`MSC_LICENSE_FILE` is needed if it is not defined as a global environment variable. N is the number of threads (SMP) or domains (DMP).

When running Digimat-CAE/Abaqus coupled simulations using Digimat-RP, it is not needed to take care on copying `abaqus_v6.env` file. User has just to select Abaqus location in Digimat-RP settings (see Figure 4-1) and to select SMP or DMP computation in Digimat-RP GUI when running simulation (see Figure 4-2).

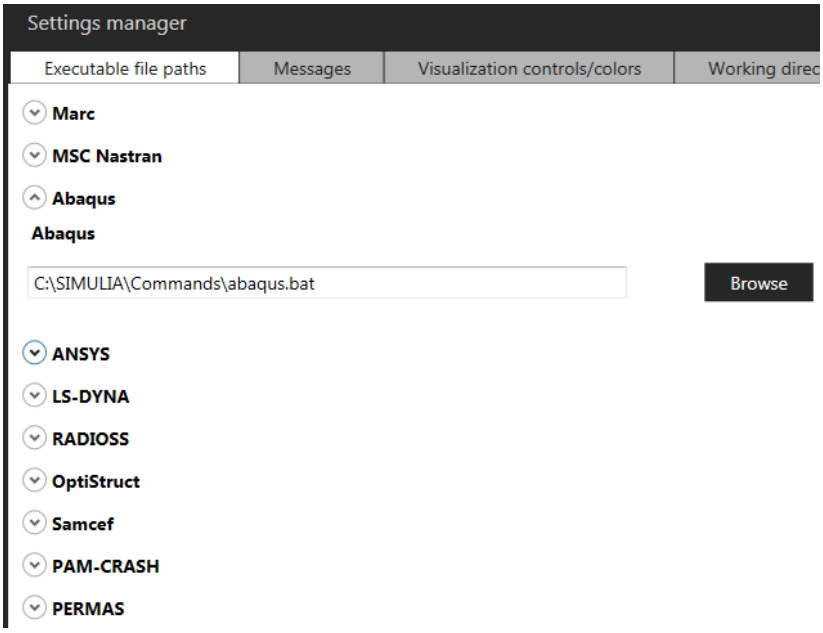


Figure 4-1 Setting Abaqus location in Digimat-RP.

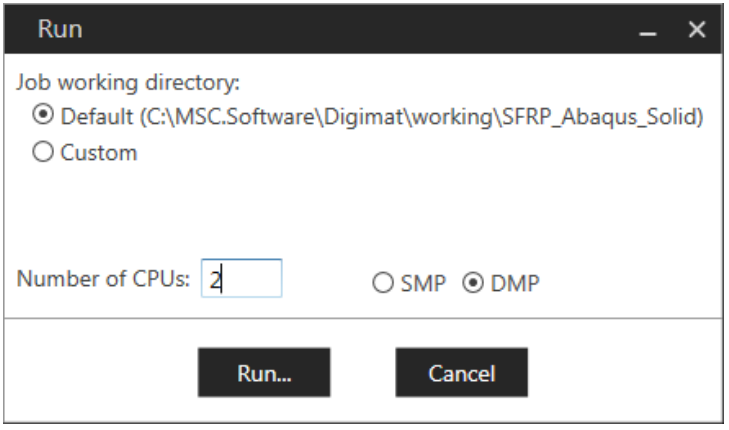


Figure 4-2 Choice of parallelization method in Digimat-RP.

Linux Platforms

To run Digimat to Abaqus coupled analysis, it is needed to copy the `abaqus_v6.env` of targeted Abaqus version and parallelization method (SMP or DMP) from Digimat installation directory to working directory. `abaqus_v6.env` file can also be generated on the fly by launching script, based on what is provided in Digimat installation.

As soon as `abaqus_v6.env` file exists in working directory, coupled Digimat-CAE/Abaqus simulation can run using command:

```
/opt/DassaultSystemes/SIMULIA/Commands/abaqus job=test.inp cpus=N
```

where N is the targeted number of threads (SMP) or domains (DMP).

Digimat-to-Abaqus Plug-in Installation

The path to the Abaqus installation must be specified during the installation of Digimat. The Digimat plugins are automatically installed for this version of Abaqus, e.g., if Abaqus targeted version is 2023, the required script files are copied to the folder:

```
C:\SIMULIA\EstProducts\2023\abaqus_plugins
```

To use the plugins in another version of Abaqus, you can either:

- Copy the `abaqus_plugins` folder to the appropriate Abaqus installation folder
- Add (or edit) the following line in the default `abaqus_v6.env` file:

  ```
  plugin_central_dir=r"DIGIMAT_DIR\DigimatCAE\exec\digi2Aba\abaqus_plugins\"
  ```

  where `DIGIMAT_DIR` is the Digimat installation directory, e.g., `C:\MSC.Software\Digimat\2023.1`.

Remarks:

- If you did both operations (e.g., copy the `abaqus_plugins` folder and edit the `abaqus_v6.env` file), Abaqus will warn you at startup that duplicate scripts files are found, and will use the ones from the `abaqus_plugins` folder. So if you want to use a different version of the plugin with Abaqus, you shall modify those `abaqus_plugins` files.
- The Abaqus default working directory (typically `C:\Temp` may contain residual files from previous installations; e.g., an `abaqus_plugins` folder, a `DIGIMAT_Settings.ini` file, and a `abaqus_v6.env` file. These files must be removed manually for the plugin to behave correctly.
- To be able to use plugin, it is mandatory to define global `DIGIMAT_BIN_20231` environment variable pointing to the Digimat directory containing `DIGIMAT_Settings.ini` file, e.g.,

  ```
  C:\MSC.Software\Digimat\2023.1\Digimat\exec
  ```

- The plugin version and the Digimat version are independent, e.g., you can use a version of the plugin with a different Digimat version:
  - The plugin version is given directly in the plugin script files;
  - The Digimat version used by the plugin (e.g. to generate the interface file, call Digimat-MX, etc.) is determined through the `DIGIMAT_BIN_20231` environment variable;

- The Digimat version used to run Digimat/Abaqus coupled analyses is given in the `abaqus_v6.env` file by the `usub_lib_dir` line.

Digimat-CAE/ANSYS

The objective of this section is to explain how to install the interface between Digimat and ANSYS. For more information concerning supported ANSYS releases and platforms, please refer to the below section Supported Versions.

Supported Versions

Digimat 2023.1 supports the following releases of ANSYS software:

- ANSYS 2021R2 (aka 21.2)
- ANSYS 2022R2 (aka 22.2)
- ANSYS 2023R1 (aka 23.1)

The ACT Digimat plug-in for ANSYS Workbench is supported for the following versions of ANSYS software:

- ANSYS 2021R2 (Windows platform) (aka 21.2)
- ANSYS 2022R2 (Windows platform) (aka 22.2)
- ANSYS 2023R1 (Windows platform) (aka 23.1)

Digimat 2023.1 supports the following platforms for ANSYS interface:

- Windows 10 (64-bit)
- Windows Server 2019 (64-bit)
- Linux Red Hat 7.9 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux Red Hat 8.4 using GLIBC ≥ 2.28 and GLIBCXX ≥ 3.4.25
- Linux SUSE 12 SP1 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP2 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP4 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 15 SP1 using GLIBC ≥ 2.26 and GLIBCXX ≥ 3.4.28

Digimat 2023.1 supports following parallelization methods:

- For Windows platforms
  - Shared Memory Parallelization (SMP)
  - Distributed Memory Parallelization (DMP) using Intel-MPI (default in ANSYS)
  - Distributed Memory Parallelization (DMP) using MS-MPI
- For Linux platforms
  - Shared Memory Parallelization (SMP)
  - Distributed Memory Parallelization (DMP) using Intel-MPI (default in ANSYS)

Digimat-CAE/ANSYS for Windows

Digimat-CAE/ANSYS is provided as a set of dynamic libraries. These libraries contain Digimat capabilities, allowing the use of Digimat materials in ANSYS analyses. This link is performed on both Linux and Windows operating systems.

Dynamic libraries are version dependent. For example, it means that libraries provided for ANSYS 2023R1 cannot be used with previous ANSYS versions.

Dynamic libraries are located in the directory

```
DIGIMAT_DIR\DigimatCAE\exec\digi2ansys
```

of Digimat installation. `DIGIMAT_DIR` is main Digimat installation directory, e.g.,

```
C:\MSC.Software\Digimat\2023.1
```

This directory contains subfolders, each subfolder corresponding to a given supported ANSYS versions. Each of these folders contains subfolders corresponding to the different supported parallelization versions:

- `SMP` folder for Shared Memory Parallelization.
- `DMP_INTELMPI` folder for Distributed Memory parallelization using Intel-MPI.
- `DMP_MSMPI` folder for Distributed Memory parallelization using Microsoft-MPI.

Each of these folders contains three dynamic libraries needed by ANSYS:

- `UserMatLib.dll`
- `USolBegLib.dll`
- `USolFinLib.dll`

`digi2ansys` directory also contains dynamic library `digi2ansys.dll` containing Digimat kernel. This library is common for all ANSYS versions. Finally, third-party components

- `boost_chrono-mt-x64.dll`
- `boost_filesystem-mt-x64.dll`
- `boost_iostreams-mt-x64.dll`
- `boost_regex-mt-x64.dll`
- `boost_system-mt-x64.dll`
- `boost_thread-mt-x64.dll`
- `boost_zlib-mt-x64.dll`
- `digimatMathTools.dll`
- `digimatPocoFoundation.dll`
- `lapi.dll`
- `VMAP.dll`

are located in `digi2ansys` directory.

In order to use Digimat-CAE/ANSYS in command line or via Digimat ACT plugin (see section Digimat-CAE/ANSYS Workbench ACT Plug-in for Windows), following manual operations are needed:

- The PATH environment variable must be updated by prepending it the path to Digimat to ANSYS directory, e.g.,

  ```
  C:\MSC.Software\Digimat\2023.1\DigimatCAE\exec\digi2Ansys
  ```

  The procedure to edit the PATH environment variable is detailed below.

  Remarks:

  - It is recommended to prepend the `digi2Ansys` directory to the PATH, not to append it.
  - The PATH should contain only once the `digi2Ansys` directory.

- The environment variable `ANS_USER_PATH` must be defined and must point to the directory containing the required dynamic libraries. `ANS_USER_PATH` value depends on the parallelization method that will be used for the ANSYS solver:

  - **SMP**: Point the environment variable to the sub-folder called `SMP`
  - **DMP - Intel-MPI**: Point the environment variable to the sub-folder called `DMP_INTELMPI`
  - **DMP - MS-MPI**: Point the environment variable to the sub-folder called `DMP_MSMPI`

  Remarks:

  - When using ANSYS Workbench, the default MPI library is Intel-MPI.
  - If the `ANS_USER_PATH` variable is incorrectly defined, the ANSYS computation may still run, but yield incorrect results (null stress everywhere and deformation localized around boundary conditions).

- `MSC_LICENSE_FILE` environment variable must be defined and must point to the license server address (including port number).
- `FLEXLM_TIMEOUT` environment variable can be needed if using geographically distant license server to avoid failing of license checkout if answer of network is too slow. A meaningful value for `FLEXLM_TIMEOUT` is 5000000.

The environment variables can be accessed by Control panel in “System and Security” section. The “Advanced settings” will contain a choice to set “Environment variables...”. Please refer to Figure 4-3 to Figure 4-5 for an example related to Windows 10 operating system.

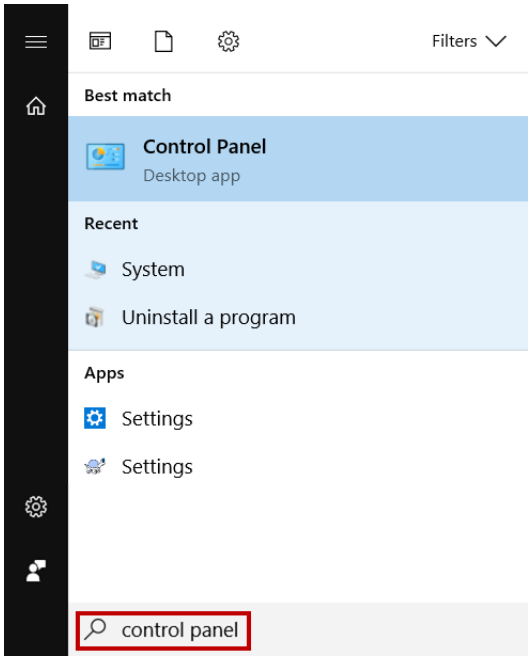


Figure 4-3 Setting the ANS_USER_PATH environment variable under Windows 10 operating system - step 1.

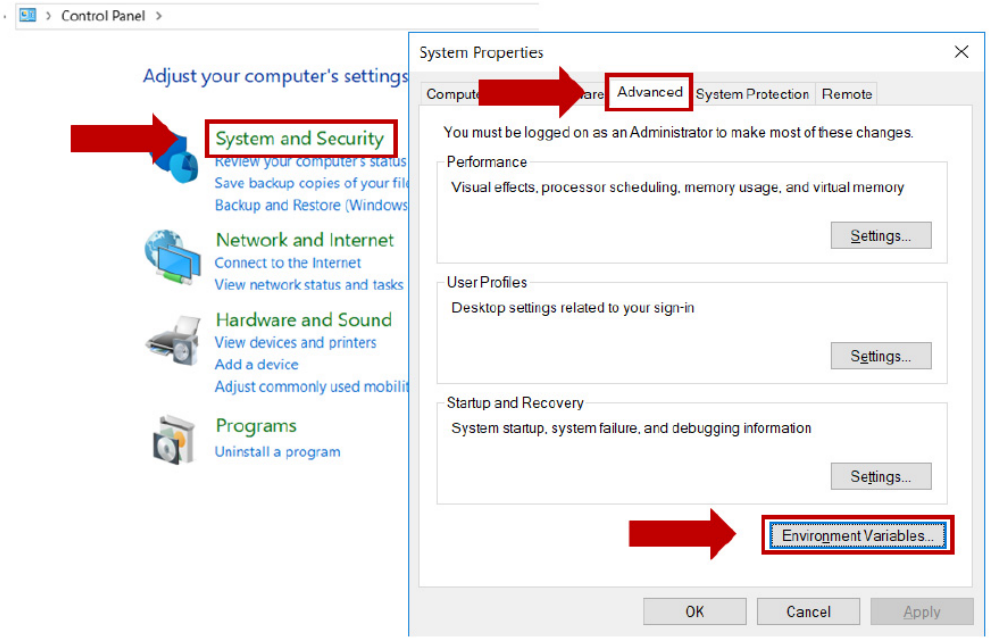


Figure 4-4 Setting the ANS_USER_PATH environment variable under Windows 10 operating system - steps 2 and 3.

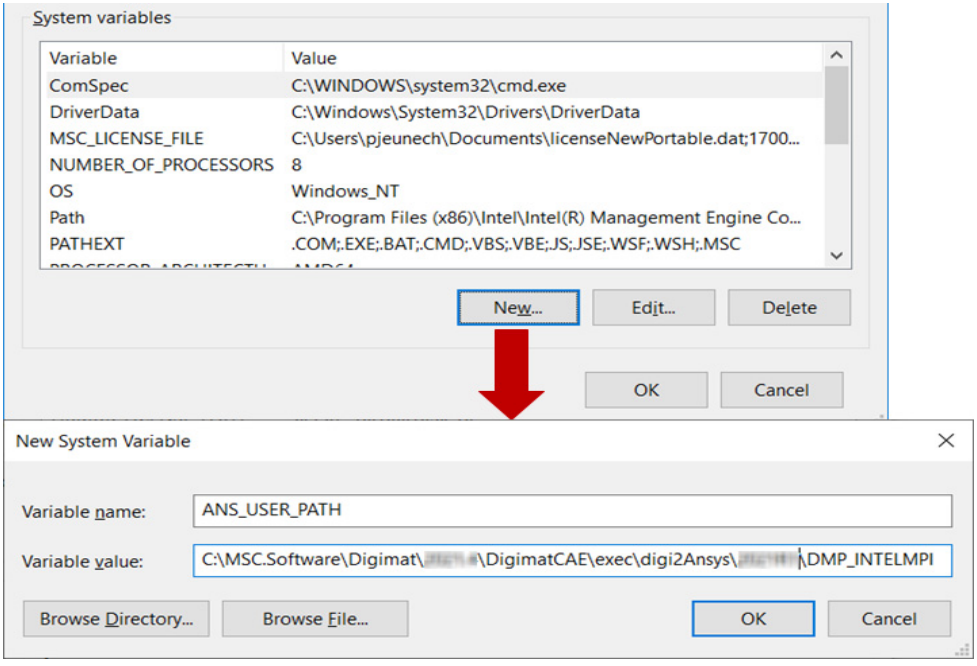


Figure 4-5 Setting the ANS_USER_PATH environment variable under Windows 10 operating system - steps 4 and 5.

Alternatively, when using command line, the Windows command prompt can be used to set the environment variables. This is done by executing a command line (`C:\Windows\system32\cmd.exe`), e.g., for computations using Intel-MPI parallelization:

```
set FLEXLM_TIMEOUT=5000000
set MSC_LICENSE_FILE=27500@localhost
set PATH=DIGIMAT_DIR\DigimatCAE\exec\digi2Ansys;%PATH%
set ANS_USER_PATH=DIGIMAT_DIR\DigimatCAE\exec\digi2Ansys\2023R1\DMP_INTELMPI
```

where `DIGIMAT_DIR` is the Digimat installation directory, e.g.,

```
C:\MSC.Software\Digimat\2023.1
```

Definition of `FLEXLM_TIMEOUT` environment variable can be needed if using geographically distant license server to avoid failing of license checkout if answer of network is too slow. A meaningful value for `FLEXLM_TIMEOUT` is 5000000. Definition of `MSC_LICENSE_FILE` is needed if it is not defined as a global environment variable.

If `digi2ansys.dll` library and third-party libraries

- `boost_chrono-mt-x64.dll`
- `boost_filesystem-mt-x64.dll`
- `boost_iostreams-mt-x64.dll`
- `boost_regex-mt-x64.dll`
- `boost_system-mt-x64.dll`
- `boost_thread-mt-x64.dll`
- `boost_zlib-mt-x64.dll`
- `digimatMathTools.dll`
- `digimatPocoFoundation.dll`
- `lapi.dll`
- `VMAP.dll`

are moved to another directory, `PATH` variable has to be updated by adding the location of this new directory.

Distributed computation with MS-MPI can be executed only if the Microsoft MPI is installed. This can be downloaded at the Microsoft homepage.

If a bad version of MS-MPI is used, the computation will stop immediately.

Multiple installation of Digimat software In case the user wants to have several versions of Digimat installed in parallel, the `PATH` (system or user) variable has to be adjusted manually and must point at first to the location of the desired version of Digimat installation for the ANSYS interface, e.g.,

```
C:\MSC.Software\Digimat\2023.1\DigimatCAE\exec\digi2Ansys
```

Note:	When uninstalling Digimat, this information given in the PATH variable is not deleted.

Digimat-CAE/ANSYS Workbench ACT Plug-in for Windows

Digimat offers an ANSYS Workbench Customization Toolkit (ACT) extension for ANSYS Workbench. This extension simplifies the integration of Digimat materials in an structural analysis, the post-processing of some Digimat history variables, and provides a better integration with Digimat-MAP. It does not require any specific licensing feature.

To install the DigimatACTplugin extension, follow this procedure:

1. Open ANSYS Workbench.
2. Browse the Extensions menu and select **Install Extension...** (Figure 4-6).
3. Browse to the DigimatACTplugin.wbex file, provided in the Digimat directory, in the sub-folder:

   `DigimatCAE\exec\digi2Ansys\workbenchACTplugin`.

This will install the extension by copying the corresponding files in ANSYS installation directory:

`%appdata%\Ansys\v231\ACT\extensions`

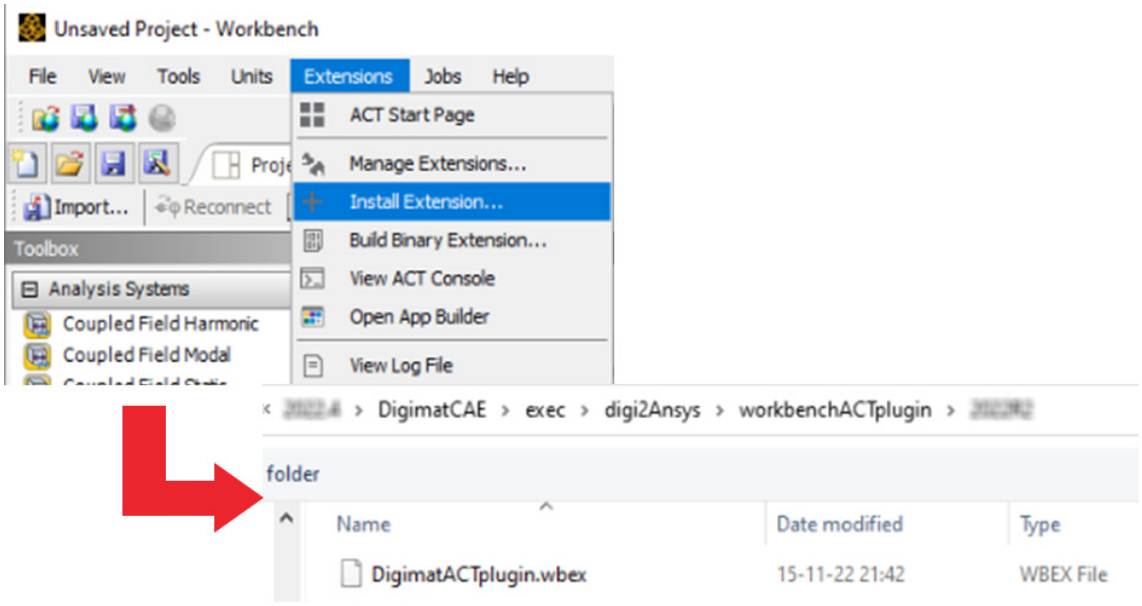


Figure 4-6 Installing the DigimatACTplugin extension for ANSYS Workbench.

For each new ANSYS Workbench session where the Digimat ACT plugin is needed, it must be loaded by browsing the Extensions menu, selecting **Manage Extensions...**, and ticking the box in front of **Digimat-ACTplugin** (Figure 4-7).

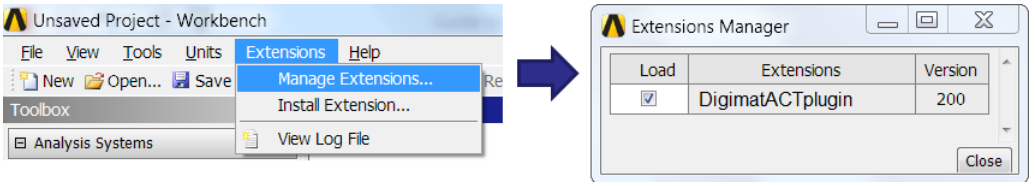


Figure 4-7 Loading the DigimatACTplugin extension for ANSYS Workbench.

Remarks:

- The Digimat ACT plugin is specific to given ANSYS Workbench and Digimat versions; it needs being reinstalled when upgrading ANSYS or Digimat.
- In complement of definition of `ANS_USER_PATH` environment variable, it is mandatory to define 3 other global environment variables:
  - `DIGIMAT_BIN_20231` environment variable pointing to the Digimat directory containing `DIGIMAT_Settings.ini` file, e.g.,

    `C:\MSC.Software\Digimat\2023.1\Digimat\exec`
  - `MSC_LICENSE_FILE` pointing to Digimat license server/file
  - If using geographically distant license server it can be needed to define `FLEXLM_TIMEOUT` environment variable to avoid failing of license checkout if answer of network is too slow. A meaningful value for `FLEXLM_TIMEOUT` is `5000000`.

Digimat-CAE/ANSYS for Linux

Under Linux, the Digimat-CAE/ANSYS interface is distributed using the shared library libansuser.so. This library is located in Digimat installation directory and depends on targeted parallelization method:

```
DIGIMAT_DIR/DigimatCAE/exec/digi2ansys/INTELMPI
DIGIMAT_DIR/DigimatCAE/exec/digi2ansys/SMP
```

where `DIGIMAT_DIR` is Digimat installation directory, e.g., `/opt/msc/Digimat/2023.1`.

In order to use Digimat-CAE/ANSYS, environment variables `MSC_LICENSE_FILE`, `DIGIMAT_BIN_20231` and `ANS_USER_PATH` must be defined. `MSC_LICENSE_FILE` must point to the address of Digimat license server. `ANS_USER_PATH` must point to the directory containing the `libansuser.so`. `DIGIMAT_BIN_20231` environment variable must point to a valid `DIGIMAT_Settings.ini` file. `FLEXLM_TIMEOUT` environment variable is needed if using geographically distant license server to avoid failing of license checkout if answer of network is too slow. A meaningful value for `FLEXLM_TIMEOUT` is 5000000.

This can be done for example by executing a script containing the following line, e.g., for `INTELMPI` computation:

```
export MSC_LICENSE_FILE=27500@localhost
export DIGIMAT_BIN_20231=/opt/msc/Digimat/2023.1/Digimat/exec
export FLEXLM_TIMEOUT=5000000
export ANS_USER_PATH=/opt/msc/Digimat/2023.1/DigimatCAE/exec/digi2ansys/INTELMPI
```

```
export LD_LIBRARY_PATH=/opt/msc/Digimat/2023.1/DigimatCAE/exec/digi2ansys/INTELMPI:
      /opt/msc/Digimat/2023.1/Digimat/lib:$LD_LIBRARY_PATH
/opt/ansys/v231/ansys/bin/ansys231 -dis -mpi intelmpi -np 4 -b -i test.inp -o ansys-out.txt
```

Launching Jobs with Remote Solve Manager

In order to run Digimat/ANSYS coupled analyses from Workbench with Remote Solve Manager (RSM), some Digimat input files (material file, orientation file, etc.) need to be copied on the remote node. Those files are not copied by default by ANSYS RSM.

The Workbench user interface provides a way to set those additional files to be copied. On the local machine, open Workbench Mechanical module, then under Home / Solve process settings / Advanced (for Background or Remote setting), define the following fields (see Figure 4-8):

- Additional files to Upload:

  .mat;.dof;*.xml;*_m??;*.bou;*.ele.???;*.o2d;*.mtc;*.stf;*.dsf;*.m2d
- Additional files to Download:

  *.mat;*DigimatMaterial*.log

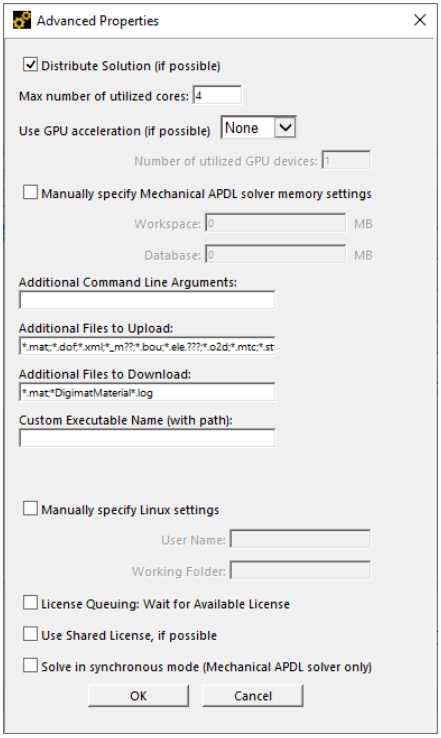


Figure 4-8 Defining file to be copied by ANSYS RSM, from ANSYS Workbench advanced solve process settings.

This operation needs to be performed once for each ANSYS version.

Digimat-CAE/Marc

The objective of this section is to explain how to install the interface between Digimat and Marc.

Installation Procedure

Digimat-CAE/Marc is the module containing the Digimat capabilities and the required libraries in order to be used with the Marc implicit solver. For more information on supported versions and platforms, please refer to the below section Supported Versions.

Supported Versions

Digimat 2023.1 supports the following releases of Marc software:

- Marc 2021.4
- Marc 2022.2
- Marc 2022.4

Digimat 2023.1 supports the following releases of Marc Mentat software (for Digimat to Marc Mentat plugin):

- Marc Mentat 2021.4
- Marc Mentat 2022.2
- Marc Mentat 2022.4

Digimat 2023.1 supports the following platforms for Marc interface:

- Windows 10 (64-bit)
- Windows Server 2019 (64-bit)
- Linux Red Hat 7.9 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux Red Hat 8.4 using GLIBC ≥ 2.28 and GLIBCXX ≥ 3.4.25
- Linux SUSE 12 SP1 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP2 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP4 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 15 SP1 using GLIBC ≥ 2.26 and GLIBCXX ≥ 3.4.28

Digimat 2023.1 supports following parallelization methods:

- For Windows platforms
  - Shared Memory Parallelization (SMP)
  - Distributed Memory Parallelization (DMP) using Intel-MPI (default in Marc)
  - Distributed Memory Parallelization (DMP) using MS-MPI
- For Linux platforms
  - Shared Memory Parallelization (SMP)

- Distributed Memory Parallelization (DMP) using Intel-MPI

Windows Platforms

The Digimat-CAE/Marc functionalities are embedded inside a new Marc executable `digi2marc.exe` shipped with Digimat installation. The `digi2marc.exe` executable is located in:

```
DIGIMAT_INSTALL_DIR\DigimatCAE\exec\digi2marc
```

where `DIGIMAT_INSTALL_DIR` is Digimat installation directory, e.g., `C:\MSC.Software\Digimat\2023.1`. User has then to choose Marc version and MPI versions in the directory tree. For example, if Marc 2022.4 is selected using INTEL-MPI,

```
DIGIMAT_INSTALL_DIR\DigimatCAE\exec\digi2marc\2022.4\INTELMPI\digi2marc.exe
```

file must be selected.

Prerequisites to use Digimat-CAE/Marc interface

Microsoft MPI is requested if user intents to perform parallel computation using MS-Message Passing Interface (MS-MPI). It can be downloaded at the following address: Microsoft homepage.

If `digi2marc.exe` file is moved to another directory, it is required to also move in this directory the following files:

- `digi2marc.dll`
- `digi2marcOEM_Lic.dll`
- `MeshDataStructure.dll`
- `plyCalibrator.dll`
- `libiomp5md.dll`

and the third-party component libraries:

- `boost_chrono-mt-x64.dll`
- `boost_filesystem-mt-x64.dll`
- `boost_iostreams-mt-x64.dll`
- `boost_regex-mt-x64.dll`
- `boost_system-mt-x64.dll`
- `boost_thread-mt-x64.dll`
- `boost_zlib-mt-x64.dll`
- `digimatMathTools.dll`
- `digimatPocoFoundation.dll`
- `lapi.dll`
- `VMAP.dll`

Launching Digimat-CAE/Marc Computations on Windows Platforms

For a single processor application, in a DOS command prompt, execute the commands:

```
set FLEXLM_TIMEOUT=5000000
MARC_INSTALL_DIR\tools\run_marc.bat -j model.dat -prog DIGI2MARC_DIR\digi2marc
```

MARC_INSTALL_DIR is the Marc installation directory. DIGI2MARC_DIR the directory where digi2marc.exe is located. First line is needed when using geographically distant license server to avoid failing of license checkout is answer of network is too slow. A meaningful value for FLEXLM_TIMEOUT is 5000000.

SMP computations can be executed by using the command in a command prompt:

```
set FLEXLM_TIMEOUT=5000000
MARC_INSTALL_DIR\tools\run_marc.bat-j model.dat-prog DIGI2MARC_DIR\digi2marc -nts N -nte N
```

where N is the number of threads. First line is needed when using geographically distant license server to avoid failing of license checkout is answer of network is too slow. A meaningful value for FLEXLM_TIMEOUT is 5000000.

DDM computations using INTELM-MPI can be executed by using the command in a command prompt:

```
set FLEXLM_TIMEOUT=5000000
MARC_INSTALL_DIR\tools\run_marc.bat -j model.dat -prog DIGI2MARC_DIR\digi2marc -nps N
```

where N is the number of CPUs. First line is needed when using geographically distant license server to avoid failing of license checkout is answer of network is too slow. A meaningful value for FLEXLM_TIMEOUT is 5000000.

DDM run using MS-MPI can be executed by using the command in a command prompt:

```
set FLEXLM_TIMEOUT=5000000
MARC_INSTALL_DIR\tools\run_marc.bat-j model.dat -prog DIGI2MARC_DIR\digi2marc -nps N -mpi ms-mpi
```

where N is the number of CPUs. First line is needed when using geographically distant license server to avoid failing of license checkout is answer of network is too slow. A meaningful value for FLEXLM_TIMEOUT is 5000000.

Launching Digimat-CAE/Marc computations using Marc Mentat

To use Marc Mentat to launch computations, it is needed to supersede the original executable marc.exe file:

1. Create a backup of the original file

   MARC_INSTALL_DIR\bin\win64i8\marc.exe (e.g., marc_orig.exe).

2. Copy the file digi2marc.exe in the directory MARC_INSTALL_DIR\bin\win64i8. Rename it marc.exe.

3. Copy in the directory MARC_INSTALL_DIR\lib\win64i8 the files

   - digi2marc.dll
   - digi2marcOEM_Lic.dll
   - MeshDataStructure.dll
   - plyCalibrator.dll
   - libiomp5md.dll

   and the third-party component libraries

- boost_chrono-mt-x64.dll
- boost_filesystem-mt-x64.dll
- boost_iostreams-mt-x64.dll
- boost_regex-mt-x64.dll
- boost_system-mt-x64.dll
- boost_thread-mt-x64.dll
- boost_zlib-mt-x64.dll
- digimatMathTools.dll
- digimatPocoFoundation.dll
- lapi.dll
- VMAP.dll

Linux Platforms

The Digimat-CAE/Marc functionalities are embedded inside a new Marc executable `digi2marc.exec` shipped with Digimat installation. For example, for Marc 2022.4, the `digi2marc.exec` executable is located in:

```
DIGIMAT_INSTALL_DIR/DigimatCAE/exec/digi2marc/2022.4
```

where `DIGIMAT_INSTALL_DIR` is the installation directory of Digimat 2023.1, e.g.,

```
/opt/msc/Digimat/2023.1
```

Launching Digimat-CAE/Marc computations on Linux platforms

To launch computations under Linux platforms, write a script containing the following lines:

```
export MSC_LICENSE_FILE=27500localhost
export FLEXLM_TIMEOUT=5000000
MARC_INSTALL_DIR/tools/run_marc -j nameProblem.dat -prog
DIGIMAT_INSTALL_DIR/DigimatCAE/exec/digi2marc/2022.4/digi2marc -nps N
```

where

- `MSC_LICENSE_FILE` environment variable points to license server address (including port).
- `DIGIMAT_INSTALL_DIR` is the Digimat installation directory.
- `N` the number of CPUs.
- The `nps` argument is not mandatory for 1 CPU. This argument can be replaced by `nts`, `nte` or `nsolver` according to the considered parallelization.
- If using geographically distant license server it can be needed to define `FLEXLM_TIMEOUT` environment variable to avoid failing of license checkout is answer of network is too slow. A meaningful value for `FLEXLM_TIMEOUT` is 5000000.

Installation of the Digimat Plug-in for Marc Mentat

The plugin files are stored in Digimat installation in directory:

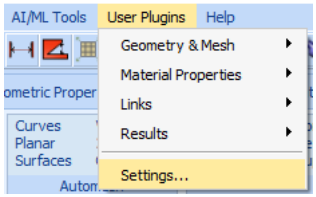
```
DIGIMAT_INSTALL_DIR\DigimatCAE\exec\digi2marc\mentat_plugin
```

where `DIGIMAT_INSTALL_DIR` is Digimat installation, e.g.,

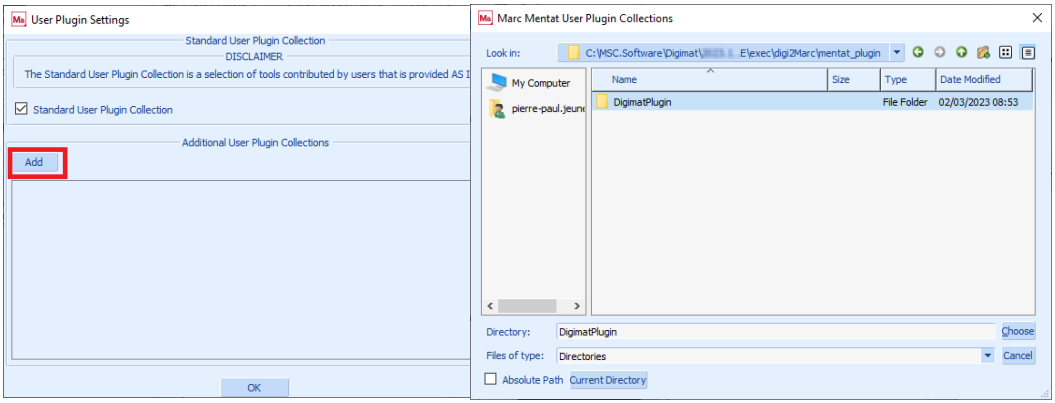
```
C:\MSC.Software\Digimat\2023.1
```

To setup Digimat Plug-in, the following 3 steps must be achieved in Mentat GUI:

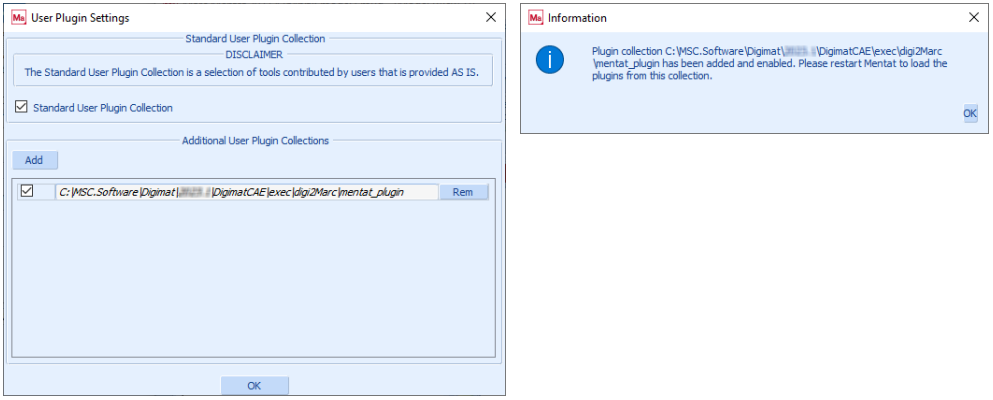
1. In **User plugins** menu, go to **Settings...**.



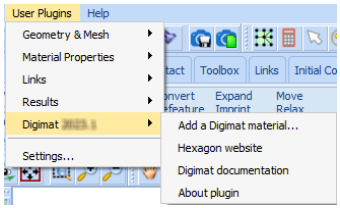
2. Click on **Add**, and select the directory `DIGIMAT_INSTALL_DIR\DigimatCAE\exec\digi2Marc\mentat_plugin\DigimatPlugin`.



If plug-in load is successful, the following message will be displayed.



3. Close and restart Marc Mentat. If plugin installation is successful, Digimat plugin installation will be available in **User Plugins** tab as shown in the figure below.



Installation of the Digimat Plug-in for Marc Mentat 2021.4

The plugin files are stored in Digimat installation in directory:

```
DIGIMAT_INSTALL_DIR\DigimatCAE\exec\digi2marc\mentat20214_plugin
```

where `DIGIMAT_INSTALL_DIR` is Digimat installation, e.g.,

```
C:\MSC.Software\Digimat\2023.1
```

During the Digimat installation procedure, those files are also copied in a Digimat folder in the Marc Mentat installation directory selected during Digimat installation, e.g.,:

```
MARC_ROOT_INSTALL_DIR\mentat2021.4\digimat
```

where `MARC_ROOT_INSTALL_DIR` is the Marc root installation directory, e.g.,

```
C:\Program Files\MSC.Software
```

If the Marc Mentat installation directory does not contain a Digimat folder, copy it from the Digimat installation (see `mentat_plugin` directory here above). This might happen if Marc Mentat was installed after Digimat or if several versions of Marc Mentat are installed; in which case, the Digimat folder was created only in the installation directory of the latest version.

To be able to use the plugin from within Marc Mentat, perform the following steps.

1. Insert the content of the file

   ```
   MARC_ROOT_INSTALL_DIR\Marc\2021.4.0\mentat2021.4\digimat\mentatPlugin.xml
   ```

   at the end of the file

   ```
   MARC_ROOT_INSTALL_DIR\Marc\2021.4.0\mentat2021.4\menus\menubar.xml
   ```

   (just before the last line, containing `</menubar>`). Create first a backup of the original file `menubar.xml` (e.g., `menubar_orig.xml`) and possibly disable the **Read only** status in the file properties.

2. Compile the Digimat plugin menu file `digimat.ms` by running the following MS-DOS commands:

   ```
   cd "MARC_ROOT_INSTALL_DIR\Marc\2021.4.0\mentat2021.4\digimat"
   ..\bin\mentat -compile main.msb
   ```

   It may be required to start the MS-DOS command prompt with administrator rights, especially if Marc is installed in the `C:\Program Files` folder.

3. Copy the new file `main.msb` in `MENTAT_INSTALL_DIR\menus\win64` after having created a backup of the original file `main.msb` (e.g., `main_orig.msb`).

To launch computations set up using the plugin from within Marc Mentat, supersede the original executable `marc.exe` (see section about launching Digimat-CAE/Marc computations via Marc Mentat here above).

Remark: It is mandatory to define global `DIGIMAT_BIN_20231` environment variable pointing to the Digimat directory containing initial `DIGIMAT_Settings.ini` file, e.g.,

```
C:\MSC.Software\Digimat\2023.1\Digimat\exec
```

If using geographically distant license server it can be needed to define `FLEXLM_TIMEOUT` environment variable to avoid failing of license checkout is answer of network is too slow. A meaningful value for `FLEXLM_TIMEOUT` is 5000000.

Digimat-CAE/MSC Nastran SOL400

The objective of this section is to explain how to install the interface between Digimat and MSC Nastran SOL400.

Supported Versions

Since Digimat libraries are shipped with the MSC Nastran installer after the Digimat release, please refer to the MSC Nastran SOL400 documentation to know which Digimat version and which platforms are supported.

Digimat 2023.1 supports following parallelization methods:

- Shared Memory Parallelization (SMP)
- Distributed Memory Parallelization (DMP)

For Digimat 2023.1, parallel processing using DMP is not stable and currently not recommended. It is recommended to use SMP parallelization.

Installation Procedure

Digimat and MSC Nastran SOL400 can be coupled together by the use of dynamic libraries. These libraries contain Digimat capabilities, allowing to use Digimat materials in MSC Nastran SOL400 analyses. This link is performed on both Linux and Windows operating systems. Dynamic libraries are distributed by Hexagon in MSC Nastran installation.

Remark: `FLEXLM_TIMEOUT` environment variable can be needed if using geographically distant license server to avoid failing of license checkout if answer of network is too slow. A meaningful value for `FLEXLM_TIMEOUT` is 5000000.

Executing a job under Windows 64-bit using command line

```
NASTRAN_INST_DIR\20XXX\bin\nast20XXX.exe nameOfInputDeck.bdf
```

where `NASTRAN_INST_DIR` is the installation directory, e.g., `C:\Program Files\MSC.Software\MSC_Nastran`.

Executing a job under Windows 64-bit using MSC Nastran configuration GUI

Launch MSC Nastran. Then select the input deck to be used and click on Run. Note that, as Digimat commands are not recognized by Patran, it is not possible to launch a coupled Digimat MSC Nastran SOL400 computation from Patran.

Digimat-CAE/Samcef

Starting Digimat 2022.2, SAMCEF interface is no more supported.

Digimat-CAE/LS-DYNA

The objective of this section is to explain how to install the interface between Digimat and LS-DYNA. For more information concerning supported LS-DYNA releases and platforms, please refer to the below section Supported Versions.

Supported Versions

Digimat 2023.1 supports the following releases of LS-DYNA software:

- Under Windows
  - LS-DYNA R11.1
  - LS-DYNA R12.1
  - LS-DYNA R13.1.1
- Under Linux:
  - LS-DYNA R11.1
  - LS-DYNA R12.1
  - LS-DYNA R13.1.1

Digimat 2023.1 supports the following platforms for LS-DYNA interface:

- Windows 10 (64-bit)
- Windows Server 2019 (64-bit)
- Linux Red Hat 7.9 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP1 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP2 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP4 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19

Digimat 2023.1 supports following parallelization methods:

- For Windows platforms
  - Distributed Memory Parallelization (MPP) using Intel-MPI
  - Distributed Memory Parallelization (MPP) using Platform-MPI
  - Distributed Memory Parallelization (MPP) using MS-MPI
- For Linux platforms
  - Distributed Memory Parallelization (MPP) using Intel-MPI
  - Distributed Memory Parallelization (MPP) using Platform-MPI
  - Hybrid Memory Parallelization using Intel-MPI (only for LS-DYNA R12.1)
  - Hybrid Memory Parallelization using Platform-MPI (only for LS-DYNA R12.1)

Installation Procedure

Digimat-CAE/LS-DYNA is the module containing the Digimat capabilities and the required interfaces in order to be linked with the LS-DYNA explicit and implicit solvers. Depending on the platform that is used, linking is performed either in a dynamic or a static way.

Installation Procedure for Linux Platforms

Under Linux operating system Digimat-CAE/LS-DYNA interface is provided as a set of dynamic libraries.

LS-DYNA executables and libraries

The dynamic Digimat-CAE/LS-DYNA libraries and associated LS-DYNA executables are summarized in Table 4-1 and Table 4-2. Note that library name is independent of targeted MPI.

Table 4-1 LS-DYNA executables files to run with Digimat coupled analysis

LS-DYNA Version	Precision	MPI	Executable name
R11.1	Double	Platform-MPI	`ls-dyna_mpp_d_R11_1_0_x64_centos65_ifort160_sse2_platformmpi_sharelib`
		Intel-MPI	`ls-dyna_mpp_d_R11_1_0_x64_centos65_ifort160_sse2_intelmpi-2018_sharelib`
	Single	Platform-MPI	`ls-dyna_mpp_s_R11_1_0_x64_centos65_ifort160_sse2_platformmpi_sharelib`
		Intel-MPI	`ls-dyna_mpp_s_R11_1_0_x64_centos65_ifort160_sse2_intelmpi-2018_sharelib`
R12.1	Double	Platform-MPI	`ls-dyna_mpp_d_R12_1_0_x64_centos78_ifort160_sse2_platformmpi_sharelib`
		Intel-MPI	`ls-dyna_mpp_d_R12_1_0_x64_centos78_ifort160_sse2_intelmpi-2018_sharelib`
	Single	Platform-MPI	`ls-dyna_mpp_s_R12_1_0_x64_centos78_ifort160_sse2_platformmpi_sharelib`
		Intel-MPI	`ls-dyna_mpp_s_R12_1_0_x64_centos78_ifort160_sse2_intelmpi-2018_sharelib`
	Double	Hybrid Platform-MPI	`ls-dyna_hyb_d_R12_1_0_x64_centos78_ifort160_sse2_platformmpi_sharelib`
		Hybrid Intel-MPI	`ls-dyna_hyb_d_R12_1_0_x64_centos78_ifort160_sse2_intelmpi-2018_sharelib`
	Single	Hybrid Platform-MPI	`ls-dyna_hyb_s_R12_1_0_x64_centos78_ifort160_sse2_platformmpi_sharelib`
		Hybrid Intel-MPI	`ls-dyna_hyb_s_R12_1_0_x64_centos78_ifort160_sse2_intelmpi-2018_sharelib`

Table 4-1 LS-DYNA executables files to run with Digimat coupled analysis

LS-DYNA Version	Precision	MPI	Executable name
R13.1.1	Double	Platform-MPI	`ls-dyna_mpp_d_R13_1_1_x64_centos78_ifort190_sse2_platformmpi_sharelib`
		Intel-MPI	`ls-dyna_mpp_d_R13_1_1_x64_centos78_ifort190_sse2_intelmpi-2018_sharelib`
	Single	Platform-MPI	`ls-dyna_mpp_s_R13_1_1_x64_centos78_ifort190_sse2_platformmpi_sharelib`
		Intel-MPI	`ls-dyna_mpp_s_R13_1_1_x64_centos78_ifort190_sse2_intelmpi-2018_sharelib`

Table 4-2 LS-DYNA library files to run with Digimat coupled analysis (provided in Digimat installation).

LS-DYNA Version	Precision	Executable name
R11.1	Double	`libmppdyna_d_138999.137278.so`
	Single	`libmppdyna_s_138999.137278.so`
R12.1	Double	`libmppdyna_d_R12.1-190-gadfcdf9018_sse2.so`
	Single	`libmppdyna_s_R12.1-190-gadfcdf9018_sse2.so`
R13.1.1	Double	`libmppdyna_d_R13.1.1-6-ge41832f8f0_sse2.so`
	Single	`libmppdyna_s_R13.1.1-6-ge41832f8f0_sse2.so`

Supported MPI versions are:

- Platform-MPI 9.1.2
- Intel-MPI 2018.1

LS-DYNA executables to use with Digimat materials need to be requested from LSTC distributor. Those executables are contained in a `tar.gz` archive. This archive also contains pristine shared library. This pristine shared library must be replaced by shared libraries provided in Digimat installation. Executables depends on LS-DYNA version, targeted precision (single or double precision) and used MPI (Platform-MPI, Intel-MPI, hybrid, MPP). Archive names and sizes are listed in Table 4-3.

Table 4-3 LS-DYNA archive files to run with Digimat coupled analysis

LS-DYNA Version	Precision	MPI	Archive name	Archive version size
R11.1	Double	Platform-MPI	ls-dyna_mpp_d_R11_1_0_x64_centos65_ifort160_sse2_platformmpi_sharelib.tar.gz	95783 Ko
		Intel-MPI	ls-dyna_mpp_d_R11_1_0_x64_centos65_ifort160_sse2_intelmpi-2018_sharelib.tar.gz	99268 Ko
	Single	Platform-MPI	ls-dyna_mpp_s_R11_1_0_x64_centos65_ifort160_sse2_platformmpi_sharelib.tar.gz	63223 Ko
		Intel-MPI	ls-dyna_mpp_s_R11_1_0_x64_centos65_ifort160_sse2_intelmpi-2018_sharelib.tar.gz	67104 Ko
R12.1	Double	Platform-MPI	ls-dyna_mpp_d_R12_1_0_x64_centos78_ifort160_sse2_platformmpi_sharelib.tar.gz_extractor.sh	121457 Ko
		Intel-MPI	ls-dyna_mpp_d_R12_1_0_x64_centos78_ifort160_sse2_intelmpi-2018_sharelib.tar.gz_extractor.sh	121456 Ko
	Single	Platform-MPI	ls-dyna_mpp_s_R12_1_0_x64_centos78_ifort160_sse2_platformmpi_sharelib.tar.gz_extractor.sh	84938 Ko
		Intel-MPI	ls-dyna_mpp_s_R12_1_0_x64_centos78_ifort160_sse2_intelmpi-2018_sharelib.tar.gz_extractor.sh	84934 Ko
	Double	Hybrid Platform-MPI	ls-dyna_hyb_d_R12_1_0_x64_centos78_ifort160_sse2_platformmpi_sharelib.tar.gz_extractor.sh	125413 Ko
		Hybrid Intel-MPI	ls-dyna_hyb_d_R12_1_0_x64_centos78_ifort160_sse2_intelmpi-2018_sharelib.tar.gz_extractor.sh	125413 Ko
	Single	Hybrid Platform-MPI	ls-dyna_hyb_s_R12_1_0_x64_centos78_ifort160_sse2_platformmpi_sharelib.tar.gz_extractor.sh	87760 Ko
		Hybrid Intel-MPI	ls-dyna_hyb_s_R12_1_0_x64_centos78_ifort160_sse2_intelmpi-2018_sharelib.tar.gz_extractor.sh	87759 Ko

Table 4-3 LS-DYNA archive files to run with Digimat coupled analysis

LS-DYNA Version	Precision	MPI	Archive name	Archive version size
R13.1.1	Double	Platform-MPI	ls-dyna_mpp_d_R13_1_1_x64_centos78_ifort190_sse2_platformmpi_sharelib.tar.gz_extractor.sh	128191 Ko
		Intel-MPI	ls-dyna_mpp_d_R13_1_1_x64_centos78_ifort190_sse2_intelmpi-2018_sharelib.tar.gz_extractor.sh	128191 Ko
	Single	Platform-MPI	ls-dyna_mpp_s_R13_1_1_x64_centos78_ifort190_sse2_platformmpi_sharelib.tar.gz_extractor.sh	89713 Ko
		Intel-MPI	ls-dyna_mpp_s_R13_1_1_x64_centos78_ifort190_sse2_intelmpi-2018_sharelib.tar.gz_extractor.sh	89707 Ko

Configure Digimat-CAE/LS-DYNA computations

The following operations have to be performed to be able to use the shared Digimat-CAE/LS-DYNA library on Linux system, e.g., for LS-DYNA R13.1.1 in double precision using Platform-MPI parallelization:

- Update `LD_LIBRARY_PATH` to point to LS-DYNA/Digimat coupled libraries and to Digimat third-party libraries:

```
$ export LD_LIBRARY_PATH=DIGIMAT_DIR/Digimat/lib/:
    DIGIMAT_DIR/DigimatCAE/exec/digi2dyna/R13.1.1/Double/PCMPI
```

  where `DIGIMAT_DIR` is Digimat installation directory, e.g., `/opt/msc/Digimat/2023.1`.

- Ensure that `LD_LIBRARY_PATH` is correctly passed to each computation node. This can be done by adding on MPI command line option
  - For Platform-MPI:

```
-e LD_LIBRARY_PATH=$LD_LIBRARY_PATH
```

  - For Intel-MPI:

```
-genvall
```

Launching Digimat-CAE/LS-DYNA under Linux

To launch a Linux job of Digimat-CAE/LS-DYNA, the very same procedure as to launch a LS-DYNA standalone job can be followed. The only differences are:

- Use the correct version of LS-DYNA, i.e., the one that looks for a user material library (see S for LS-DYNA executable list)
- Make sure that this executable sees the right `LD_LIBRARY_PATH` environment variable as explained above
- Below is an example of a bash script containing all commands:

```
#!/bin/bash
```

```
export DIGIMAT_BIN_20231=DIGIMAT_DIR/Digimat/exec
export MSC_LICENSE_FILE=27500@localhost
export FLEXLM_TIMEOUT=5000000
export LD_LIBRARY_PATH=DIGIMAT_DIR/Digimat/lib:
   DIGIMAT_DIR/DigimatCAE/exec/digi2dyna/R13.1.1/Double/PCMPI:
   $LD_LIBRARY_PATH
ls-dyna_mpp_d_R13_1_1_x64_centos78_ifort190_sse2_platformmpi_sharelib $@
```

where `DIGIMAT_DIR` is the Digimat installation directory, e.g., `/opt/Digimat/2023.1`. `FLEXLM_TIMEOUT` definition can be needed when using geographically distant license server to avoid failing of license checkout if answer of network is too slow. A meaningful value for `FLEXLM_TIMEOUT` is 5000000. Make this script executable:

```
chmod a+x launch_script.sh
```

and call this script rather than the LS-DYNA executable, with the very same arguments, i.e.,

```
mpirun -np 4 path_to_my_launch_script.sh i=input.k
```

Windows Platform

Prerequisites for usage of Digimat and LS-DYNA

To link Digimat and LS-DYNA libraries together on Windows platforms, a linker is needed. Before linking Digimat-CAE/LS-DYNA, Microsoft Visual Studio 2022 must be installed (see Chapter 6: Installing Microsoft Visual Studio 2022 on how to install this software).

To be able to run parallel computations, a MPI must be installed, depending on the targeted MPI:

- Microsoft MPI (needed for MS-MPI versions)
  - An installer can be downloaded at Microsoft home page
- Platform MPI (needed for PC-MPI versions)
  - An installer can be downloaded at IBM home page. This software is not free.
- Intel-MPI
  - An installer can be downloaded at Intel home page

Building Digimat/LS-DYNA executable

Under Windows operating system a static linking procedure must be followed to be able to use Digimat/LS-DYNA interface.

For each version provided, the process to link the Digimat-CAE/LS-DYNA executable is the same:

- Digimat-CAE to LS-DYNA directory is located in:

  `C:\MSC.Software\Digimat\2023.1\DigimatCAE\exec\digi2dyna`
- On the machine where linking procedure is achieved, following operations are needed:

- Make sure Microsoft Visual Studio 2022 is installed (see Chapter 6: Installing Microsoft Visual Studio 2022 on how to install this software). Installation of Microsoft Visual Studio 2022 must be achieved only once. When upgrading Digimat version, it is not needed to reinstall Microsoft Visual Studio 2022, but only to update Digimat-CAE to LS-DYNA libraries.
- Get LS-DYNA libraries `libdyna.lib` and `libansys.lib` from your LS-DYNA distributor. Those libraries depends on LS-DYNA version, targeted precision (single or double precision) and used MPI (Platform-MPI, Intel-MPI or MS-MPI). Requested archive containing the LS-DYNA libraries to use with Digimat material are listed in Table 4-4.
- Copy the libraries in the Digimat directory corresponding to the targeted LS-DYNA version, e.g., for LS-DYNA R13.1.1 in Double precision using Platform-MPI:

  ```
  C:\MSC.Software\Digimat\2023.1\DigimatCAE\exec\digi2dyna\R13.1.1\Double\PCMPI
  ```

- Double-click on the `nmake_x64.bat`.
- An executable should be created named mppdyna.exe or `mppdyna_d.exe` depending on the version (single or double precision).
- Result of link procedure is redirected to `out.txt` file next to the `nmake_x64.bat` file.
- If the executable is not created, check the following:
  - The path to the installation of Microsoft Visual Studio 2022 in the `.bat` script if Microsoft Visual Studio 2022 has not been installed in the default directory.
  - Error messages are written in `out.txt` file. In case of issue, contact digimat.support@hexagon.com by including this `out.txt` file.
- Once linking procedure is achieved, generated executable can be moved to any other machine in appropriate installation directory. If copying generated executables, pay attention to also copy all the DLL that are located in directory of initial build.

Table 4-4 LS-DYNA archive files to run with Digimat coupled analysis

LS-DYNA Version	Precision	MPI	Archive name	Archive version size
R11.1	Double	Platform-MPI	`ls-dyna_mpp_d_R11_1_0_139588_winx64_ifort2017vs2017_pmpi_lib.zip`	108220 Ko
		MS-MPI	`ls-dyna_mpp_d_R11_1_0_139588_winx64_ifort2017vs2017_msmpi_lib.zip`	108311 Ko
		Intel-MPI	`ls-dyna_mpp_d_R11_1_0_139588_winx64_ifort2017vs2017_impi_lib.zip`	108227 Ko
	Single	Platform-MPI	`ls-dyna_mpp_s_R11_1_0_139588_winx64_ifort2017vs2017_pmpi_lib.zip`	108890 Ko
		MS-MPI	`ls-dyna_mpp_s_R11_1_0_139588_winx64_ifort2017vs2017_msmpi_lib.zip`	108958 Ko
		Intel-MPI	`ls-dyna_mpp_s_R11_1_0_139588_winx64_ifort2017vs2017_impi_lib.zip`	108885 Ko

Table 4-4 LS-DYNA archive files to run with Digimat coupled analysis

LS-DYNA Version	Precision	MPI	Archive name	Archive version size
R12.1	Double	Platform-MPI	ls-dyna_mpp_d_R12.1_190-gadfcdf9018_winx64_ifort170_pmpi_lib.zip	132319 Ko
		MS-MPI	ls-dyna_mpp_d_R12.1_190-gadfcdf9018_winx64_ifort170_msmpi_lib.zip	132299 Ko
		Intel-MPI	ls-dyna_mpp_d_R12.1_190-gadfcdf9018_winx64_ifort170_impi_lib.zip	132302 Ko
	Single	Platform-MPI	ls-dyna_mpp_s_R12.1_190-gadfcdf9018_winx64_ifort170_pmpi_lib.zip	108363 Ko
		MS-MPI	ls-dyna_mpp_s_R12.1_190-gadfcdf9018_winx64_ifort170_msmpi_lib.zip	108467 Ko
		Intel-MPI	ls-dyna_mpp_s_R12.1_190-gadfcdf9018_winx64_ifort170_impi_lib.zip	108364 Ko
R13.1.1	Double	Platform-MPI	ls-dyna_mpp_d_R13.1.1_6-ge41832f8f0_winx64_ifort190_pmpi_lib_installer.exe	120165 Ko
		MS-MPI	ls-dyna_mpp_d_R13.1.1_6-ge41832f8f0_winx64_ifort190_msmpi_lib_installer.exe	120146 Ko
		Intel-MPI	ls-dyna_mpp_d_R13.1.1_6-ge41832f8f0_winx64_ifort190_impi_lib_installer.exe	120704 Ko
	Single	Platform-MPI	ls-dyna_mpp_s_R13.1.1_6-ge41832f8f0_winx64_ifort190_pmpi_lib_installer.exe	100548 Ko
		MS-MPI	ls-dyna_mpp_s_R13.1.1_6-ge41832f8f0_winx64_ifort190_msmpi_lib_installer.exe	100628 Ko
		Intel-MPI	ls-dyna_mpp_s_R13.1.1_6-ge41832f8f0_winx64_ifort190_impi_lib_installer.exe	100557 Ko

- The `lstc_client.exe` executable next to the Digimat-CAE/LS-DYNA one is needed. It can be found in LS-DYNA installation.

Supported MPI versions are:

- Platform-MPI 9.1
- Intel-MPI 2018.1 for LS-DYNA R11.1 and R12.1
- Intel-MPI 2019.6 for LS-DYNA R13.1.1
- MS-MPI 8.1 for LS-DYNA R11.1
- MS-MPI 10.0 for LS-DYNA R12.1 and R13.1.1

Launching Digimat-CAE/LS-DYNA under Windows

To launch a Windows job of Digimat-CAE/LS-DYNA using command line, the following bat script can be used, for example for LS-DYNA R13.1.1 in double precision using Platform-MPI on 4 processors:

```
set FLEXLM_TIMEOUT=5000000
```

```
set PATH=C:\Program Files (x86)\Platform Computing\Platform-MPI\bin;%PATH%
set MSC_LICENSE_FILE=27500localhost
C:\Program Files (x86)\Platform Computing\Platform-MPI\bin\mpirun -np 4
    DIGIMAT_DIR\DigimatCAE\exec\digi2dyna\R13.1.1\Double\PCMPI\mppdyna_d.exe i=input.k
```

where:

- First line can be needed when using geographically distant license server to avoid failing of license checkout if answer of network is too slow. A meaningful value for `FLEXLM_TIMEOUT` is 5000000.
- Second line add PATH to MPI dynamic libraries.
- Third line defines `MSC_LICENSE_FILE` environment variable pointing to license server address (including port).
- Last line runs Digimat-CAE/LS-DYNA job.
- `DIGIMAT_DIR` is Digimat installation directory, e.g., `C:\MSC.Software\Digimat\2023.1`.

Generation of Encryption Key File for Macro Solution

When using Digimat-RP with Macro solution, material properties are encrypted. Generation of key is done by doing following blank run of LS-DYNA in Digimat working directory:

```
C:\Program Files (x86)\Platform Computing\Platform-MPI\bin\mpirun -np 1
    DIGIMAT_DIR\DigimatCAE\exec\digi2dyna\R13.1.1\Double\PCMPI\mppdyna_d.exe pgpkey
```

Then check that the file `lstc_pgpkey.asc` is created. Note that the key file is independent of LS-DYNA version, so, it can be generated with any LS-DYNA executable.

Digimat-CAE/PAM-CRASH

The objective of this section is to explain how to install the interface between Digimat and PAM-CRASH.

Supported Versions

Digimat 2023.1 supports the following releases of PAM-CRASH software:

- PAM-CRASH 2020.0
- PAM-CRASH 2021.0
- PAM-CRASH 2022.1

Digimat 2023.1 supports the following platforms for PAM-CRASH interface:

- Windows 10 (64-bit)
- Windows Server 2019 (64-bit)
- Linux Red Hat 7.9 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux Red Hat 8.4 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP1 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP2 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP4 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 15 SP1 using GLIBC ≥ 2.26 and GLIBCXX ≥ 3.4.28

Digimat 2023.1 supports following parallelization methods:

- For Windows platforms: Distributed Memory Parallelization (MPP) using Intel-MPI (default parallelization method)
- For Linux platforms: Distributed Memory Parallelization (MPP) using Intel-MPI (default parallelization method)

Installation Procedure

Digimat-CAE/PAM-CRASH is the module containing the Digimat capabilities and the required interfaces in order to be linked with the PAM-CRASH explicit solver. For more information on supported versions and platforms, please refer to section Supported Versions.

Digimat-CAE/PAM-CRASH is the material library containing the Digimat linear and nonlinear multi-scale material modeling capabilities. For Digimat 2023.1, a set of dynamic libraries is provided for the Digimat- CAE/PAM-CRASH interface, both for single and double precision for all supported PAM-CRASH versions. The installation procedure for the dynamic library is straightforward. While installing Digimat to PAM-CRASH, it installs the necessary libraries, including

- Under Windows:
  - For models using solid elements: `libdigimat85_N_sp.dll and libdigimat85_N_dp.dll` according to targeted precision (single or double)

    - For models using shell elements: `libdigimat185_N_sp.dll and libdigimat185_N_dp.dll` according to targeted precision (single or double)
- Under Linux:
    - For models using solid elements: `libdigimat85_N_sp.so` and `libdigimat85_N_dp.so` according to targeted precision (single or double)
    - For models using shell elements: `libdigimat185_N_sp.so` and `libdigimat185_N_dp.so` according to targeted precision (single or double)

with N ∈ [0, 3].

Those libraries will be loaded by the explicit solver (`psolid.exe`) at runtime. The library `libdigimat[1]85_N.dll/.so` and its dependencies contains definition of all possible user defined subroutines. This will allow the user to use all linear and nonlinear small-strain material models capabilities available in Digimat for small-strain analyses. Defining Digimat material model in PAM-CRASH then is just like defining any other PAM-CRASH material model.

Launching Digimat-CAE/PAM-CRASH Computations on Windows Platform

To run coupled Digimat to PAM-CRASH computations in command line, it is advised to write a `.bat` file containing all the information needed by PAM-CRASH to allow the call to user subroutine.

Running Digimat-CAE/PAM-CRASH in double precision and 4 CPUs can be done using following script:

1. @echo off
2. set `MSC_LICENSE_FILE=27500@localhost`
3. set `PATH=DIGIMAT_INST_DIR\DigimatCAE\exec\digi2Pamcrash;`

   `DIGIMAT_INST_DIR\DigimatCAE\exec\digi2Pamcrash\2022.0\bin\windows-x64-intel;%PATH%`
4. set `FLEXLM_TIMEOUT=5000000`
5. set `OMP_NUM_THREADS=1`
6. set `PAM_USER_PLUGIN_ROOT=DIGIMAT_INST_DIR\DigimatCAE\exec\digi2Pamcrash\2022.0`
7. `"PAM_INST_DIR\2022.0\Solver\bin\bin\pamcrash.bat" -np 4 -fp 2 test.pc > "test.out"`

where

- `DIGIMAT_INST_DIR` is Digimat installation directory, e.g.,

  `C:\MSC.Software\Digimat\2023.1`
- `PAM_INST_DIR` is PAM-CRASH installation directory, e.g.,

  `C:\Program Files (x86)\ESI Group\Virtual-Performance`

The first command suppress screen display of command. The second line defines the path to Digimat license path through `MSC_LICENSE_FILE` environment variable. Definition of `MSC_LICENSE_FILE` is needed if it is not defined as a global environment variable. The third command add to `PATH` environment variable the path to Digimat library `digi2pamDouble[1]85_N.dll` and Digimat third-party libraries.

The fourth command defines FLEXLM_TIMEOUT environment variable. This can be needed if using geographically distant license server to avoid failing of license checkout if answer of network is too slow. A meaningful value for FLEXLM_TIMEOUT is 5000000. The fifth command specify the number of threads to be used, only 1 thread is supported by Digimat. The sixth command gives to PAM-CRASH the path to the libdigimat[1]85_N.dll file. The last command is the command to run PAM-CRASH analysis, the -fp 2 option sets the precision to double precision and the -np 4 makes running on 4 processes. The command sends the output to the .out file. If error is encountered in Digimat, it will be written in this file.

Launching Digimat-CAE/PAM-CRASH Computations on Linux Platform

To define environment variables needed to run Digimat to PAM-CRASH coupled analysis, user can write a launch script as the one described below:

```
1.  #!/bin/bash
2.  export PAM_USER_PLUGIN_ROOT=DIGIMAT_INST_DIR/DigimatCAE/exec/
    digi2pamcrash/2022.0
3.  export LD_LIBRARY_PATH=DIGIMAT_INST_DIR/DigimatCAE/exec/digi2pamcrash/:
      DIGIMAT_INST_DIR/Digimat/lib:
      DIGIMAT_INST_DIR/DigimatCAE/exec/digi2pamcrash/2022.0/bin/linux-x64-
      intel: $LD_LIBRARY_PATH
4.  export PAM_LMD_LICENSE_FILE=27007@host
5.  export MSC_LICENSE_FILE=27500@host
6.  export FLEXLM_TIMEOUT=5000000
7.  export OMP_NUM_THREADS=1
8.  export PAMROOT=/opt/pamcrash/2022.0
9.  export PAMHOME=$PAMROOT
10. $PAMROOT/pamcrash_safe/2022.0/Linux_x86_64/bin/pamcrash
    -np 4 -fp 2 -lic CRASHSAF test.pc > test.out
```

where DIGIMAT_INST_DIR is Digimat installation directory, e.g.,

```
/opt/msc/Digimat/2023.1
```

The first command is the header to execute bash scripts. The second command defines the location of the libdigimat[1]85_N.so shared library. The third command add the path to Digimat third-party libraries to LD_LIBRARY_PATH environment variable. It is needed to also add the full path to libdigimat[1]85_N.so. The next two commands define the path to Digimat and PAM-CRASH licenses. The sixth command defines FLEXLM_TIMEOUT environment variable. This can be needed if using geographically distant license server to avoid failing of license checkout if answer of network is too slow. A meaningful value for FLEXLM_TIMEOUT is 5000000. The seventh command specifies the number of threads to be used, only 1 thread is supported by Digimat. The PAMROOT gives the path to PAM-CRASH directory. The last command launches PAM-CRASH computation. The -fp 2 option sets the precision

to double precision and the `-np 4` makes running on 4 processes. The command sends the output to the `.out` file. If error is encountered in Digimat, it will be written in this file.

Digimat-CAE/MSC Nastran SOL1XX

The objective of this section is to explain how to install the interface between Digimat and MSC Nastran SOL1XX.

Supported Versions

Digimat 2023.1 officially supports MSC Nastran 2018.1 until MSC Nastran 2022.4 versions.

Digimat should also work with older versions of MSC Nastran. However, these other versions have not been fully tested and, therefore, are not officially supported.

Digimat 2023.1 supports the same platforms as the above versions of MSC Nastran.

Digimat 2023.1 supports the same parallelization methods as the above versions of MSC Nastran.

Installation Procedure

Digimat and MSC Nastran SOL1XX can be coupled together without the need for installing anything in particular on top of the classical MSC Nastran and Digimat installations.

Executing a Job

Once coupled with one or more Digimat materials, the modified MSC Nastran input deck can be run with MSC Nastran under Windows 64-bit or Linux 64-bit environments like any other input deck, without taking any specific action. Please refer to the MSC Nastran documentation for more information.

For example, to execute a sequential job under Windows 64-bit using the command line, the user needs to type:

```
NASTRAN_INSTALL_DIR/MSC_Nastran/2022.4/bin/nast20224.exe
inputDeck_DigimatCoupled.bdf
```

where `NASTRAN_INSTALL_DIR` is the installation directory of MSC Nastran and `inputDeck_DigimatCoupled.bdf` is the name of the MSC Nastran input deck created by Digimat-RP after coupling the structural model with one or more Digimat materials.

Digimat-CAE/OptiStruct

The objective of this section is to explain how to install the interface between Digimat and OptiStruct which uses Digimat-RP.

Installation Procedure

When not using the plug-in, Digimat and OptiStruct can be coupled together without the need for installing anything in particular on top of the classical OptiStruct and Digimat installations.

Supported Versions

Digimat 2023.1 officially supports the following releases of OptiStruct software:

- OptiStruct 13.0
- OptiStruct 14.0

Digimat should also work with older versions of OptiStruct. However, these other versions have not been fully tested and, therefore, are not officially supported.

Digimat 2023.1 supports the same platforms as the above versions of OptiStruct.

Executing a Job

Once coupled with one or more Digimat materials, the OptiStruct model can be run with OptiStruct under Windows 64-bit or Linux 64-bit environments like any other OptiStruct model, without taking any specific action. Please refer to the OptiStruct documentation for more information.

For example, to execute a sequential job under Windows 64-bit using the command line, the user needs to type:

```
ALTAIR_INSTALL_DIR\14.0\hwsolvers\scripts\optistruct.bat
model_DigimatCoupled.fem
```

where `ALTAIR_INSTALL_DIR` is the Altair installation directory and `model_DigimatCoupled.fem` is the name of the OptiStruct model created by Digimat-RP after coupling the structural model with one or more Digimat materials.

Digimat-CAE/PERMAS

The objective of this section is to explain how to install the interface between Digimat and PERMAS.

Supported Versions

Digimat 2023.1 officially supports the following releases of PERMAS:

- PERMAS V17
- PERMAS V18

Digimat should also work with older versions of PERMAS. However, these other versions have not been fully tested and, therefore, are not officially supported.

Digimat 2023.1 supports the same platforms as the above version of PERMAS.

Installation Procedure

Digimat and PERMAS can be coupled together without the need for installing anything in particular on top of the classical PERMAS and Digimat installations.

Executing a Job

Once coupled with one or more Digimat materials, the PERMAS model can be run with PERMAS under Windows 64-bit or Linux 64-bit environments like any other PERMAS model, without taking any specific action. Please refer to the PERMAS documentation for more information.

For example, to execute a sequential job under Windows 64-bit using the command line, the user needs to type:

```
INTES_INSTALL_DIR\V17\bin\permas.bat model_DigimatCoupled.uci
```

where `INTES_INSTALL_DIR` is the INTES installation directory and `model_DigimatCoupled.uci` is the name of the PERMAS model created by Digimat-RP after coupling the structural model with one or more Digimat materials.

Digimat-CAE/CAE fatigue

The objective of this section is to explain how to install the interface between Digimat and CAE fatigue. For more information concerning supported CAE fatigue releases and platforms, please refer to the below section Supported Versions.

Supported Versions

Digimat 2023.1 supports the following releases of CAE fatigue software:

- CAE fatigue 2021.2, which supports Abaqus 6.14 / 2016 / 2017 / 2018 / 2019 / 2020 and 2021 FE results files as long as the Digimat-CAE/CAE fatigue interface is concerned;
- CAE fatigue 2021.4, which supports Abaqus 6.14 / 2016 / 2017 / 2018 / 2019 / 2020 and 2021 FE results files as long as the Digimat-CAE/CAE fatigue interface is concerned.

Digimat 2023.1 supports the following platforms for CAE fatigue interface:

- Windows 10 (64-bit)
- Windows Server 2019 (64-bit)
- Linux Red Hat 7.9 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux Red Hat 8.4 using GLIBC ≥ 2.28 and GLIBCXX ≥ 3.4.25
- Linux SUSE 12 SP1 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP2 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP4 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 15 SP1 using GLIBC ≥ 2.26 and GLIBCXX ≥ 3.4.28

Parallelization using SMP is not supported. Only parallelization using DMP (with Intel-MPI) is supported.

Installation Procedure

Link with Digimat is done directly done in CAE fatigue GUI. See section Setting-up Input Files for Coupled Analyses in *Digimat-CAE User's Guide*.

Digimat-CAE/nCode DesignLife

The objective of this section is to explain how to install the interface between Digimat and nCode DesignLife. For more information concerning supported nCode DesignLife releases and platforms, please refer to the below section Supported Versions.

Supported Versions

Digimat 2023.1 supports the following releases of nCode DesignLife software:

- nCode DesignLife 2020.1, which supports Abaqus 2017/2018/2019/2020 and ANSYS 2020R1 FE results files as long as the Digimat-CAE/nCode DesignLife interface is concerned.
- nCode DesignLife 2021, which supports Abaqus 2017/2018/2019/2020 and ANSYS 2020R1 FE results files as long as the Digimat-CAE/nCode DesignLife interface is concerned.
- nCode DesignLife 2022, which supports Abaqus 2017/2018/2019/2020/2021 and ANSYS 2022R1 FE results files as long as the Digimat-CAE/nCode DesignLife interface is concerned.
- nCode DesignLife 2022.1, which supports Abaqus 2017/2018/2019/2020/2021/2022 and ANSYS 2022R1 FE results files as long as the Digimat-CAE/nCode DesignLife interface is concerned.

Digimat 2023.1 supports the following platforms:

- Windows 10 (64-bit)
- Windows Server 2019 (64-bit)

From nCode DesignLife 2022.1, following platforms are also supported:

- Linux Red Hat 7.9 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux Red Hat 8.4 using GLIBC ≥ 2.28 and GLIBCXX ≥ 3.4.25
- Linux SUSE 12 SP1 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP2 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 12 SP4 using GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19
- Linux SUSE 15 SP1 using GLIBC ≥ 2.26 and GLIBCXX ≥ 3.4.28

Parallelization using SMP is not supported. Only parallelization using DMP (with Intel-MPI) is supported.

Installation Procedure

Within the Digimat installation, the dynamic library digi2ncode.dll is located in the directory

```
INSTALL_DIR\2023.1\DigimatCAE\exec\digi2ncode
```

where `INSTALL_DIR` is the Digimat installation directory. To be able to use Digimat to nCode DesignLife interface, `DIGIMAT2NCODE_SHARED_LIBS` environment variable needs to be defined. This variable enables nCode DesignLife to locate Digimat/nCode DesignLife library, to load it and to run coupled Digimat-CAE/nCode DesignLife analyses.

Path to third-party libraries

- boost_chrono-mt-x64.dll
- boost_filesystem-mt-x64.dll
- boost_iostreams-mt-x64.dll
- boost_regex-mt-x64.dll
- boost_system-mt-x64.dll
- boost_thread-mt-x64.dll
- boost_zlib-mt-x64.dll
- digimatMathTools.dll
- digimatPocoFoundation.dll
- lapi.dll
- VMAP.dll

must also be added to the PATH environment variable. If these libraries are moved to another directory, PATH variable has to be updated according to this new directory.

Remark: FLEXLM_TIMEOUT environment variable can be needed if using geographically distant license server to avoid failing of license checkout if answer of network is too slow. A meaningful value for FLEXLM_TIMEOUT is 5000000.

5 Supported Platforms

- Digimat GUI
- Digimat-MF (batch mode, no GUI)
- Digimat-FE (batch mode, no GUI)
- Digimat-VA (for remote job submission, no GUI)
- Digimat-AM (for remote job submission, no GUI)
- Digimat-CAE

Digimat GUI

Following platforms are supported:

- Windows 10 (64-bit)
- Windows Server 2019 (64-bit)

Digimat-MF (batch mode, no GUI)

Following platforms are supported:

- Windows 10 (64-bit)
- Windows Server 2019 (64-bit)
- Linux Red Hat 7.9 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux Red Hat 8.4 (GLIBC ≥ 2.28 and GLIBCXX ≥ 3.4.25)
- Linux SUSE 12 SP1 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux SUSE 12 SP2 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux SUSE 12 SP4 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux SUSE 15 SP1 (GLIBC ≥ 2.26 and GLIBCXX ≥ 3.4.28)

Digimat-FE (batch mode, no GUI)

Following platforms are supported:

- Windows 10 (64-bit)
- Windows Server 2019 (64-bit)
- Linux Red Hat 7.9 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux Red Hat 8.4 (GLIBC ≥ 2.28 and GLIBCXX ≥ 3.4.25)
- Linux SUSE 12 SP1 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux SUSE 12 SP2 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux SUSE 12 SP4 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux SUSE 15 SP1 (GLIBC ≥ 2.26 and GLIBCXX ≥ 3.4.28)

Digimat-VA (for remote job submission, no GUI)

Following platforms are supported:

- Linux Red Hat 7.9 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux Red Hat 8.4 (GLIBC ≥ 2.28 and GLIBCXX ≥ 3.4.25)
- Linux SUSE 12 SP1 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)

- Linux SUSE 12 SP2 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux SUSE 12 SP4 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux SUSE 15 SP1 (GLIBC ≥ 2.26 and GLIBCXX ≥ 3.4.28)

Digimat-AM (for remote job submission, no GUI)

Following platforms are supported:

- Linux Red Hat 7.9 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux Red Hat 8.4 (GLIBC ≥ 2.28 and GLIBCXX ≥ 3.4.25)
- Linux SUSE 12 SP1 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux SUSE 12 SP2 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux SUSE 12 SP4 (GLIBC ≥ 2.17 and GLIBCXX ≥ 3.4.19)
- Linux SUSE 15 SP1 (GLIBC ≥ 2.26 and GLIBCXX ≥ 3.4.28)

Digimat-CAE

For supported Operating system, refer to installation section of each CAE code.

6 Windows Prerequisites

- Installing Microsoft Visual Studio 2022
- Microsoft .NET Framework 4.8

Installing Microsoft Visual Studio 2022

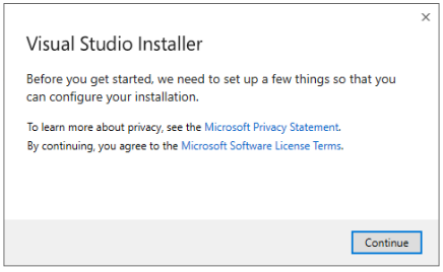
Microsoft Visual Studio 2022 has to be installed to use Digimat to LS-DYNA interface in order to build LS-DYNA executables containing Digimat libraries. Several versions of Microsoft Visual Studio 2022 can be used: Professional or Enterprise.

Visual Studio is not available for free for enterprise (see https://visualstudio.microsoft.com/license-terms/vs2022-ga-community/ for Microsoft definition of "enterprise"). The customer has to buy a license from Microsoft to be allowed to build Digimat to LS-DYNA executable

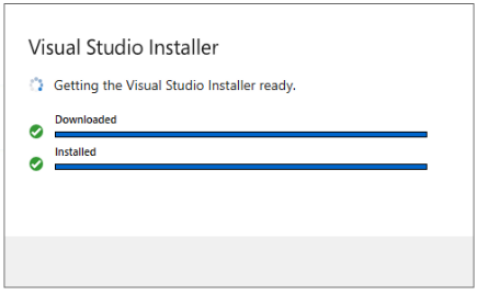
Remark: The executable can be built on one machine and then copied on all machines on which Digimat to LS-DYNA has to run (i.e., a single Visual Studio license may be sufficient). Number of license(s) required in such a scenario should be verified with Microsoft.

Installation procedure is described below. If Microsoft Visual Studio 2022 is already installed, you can skip this step.

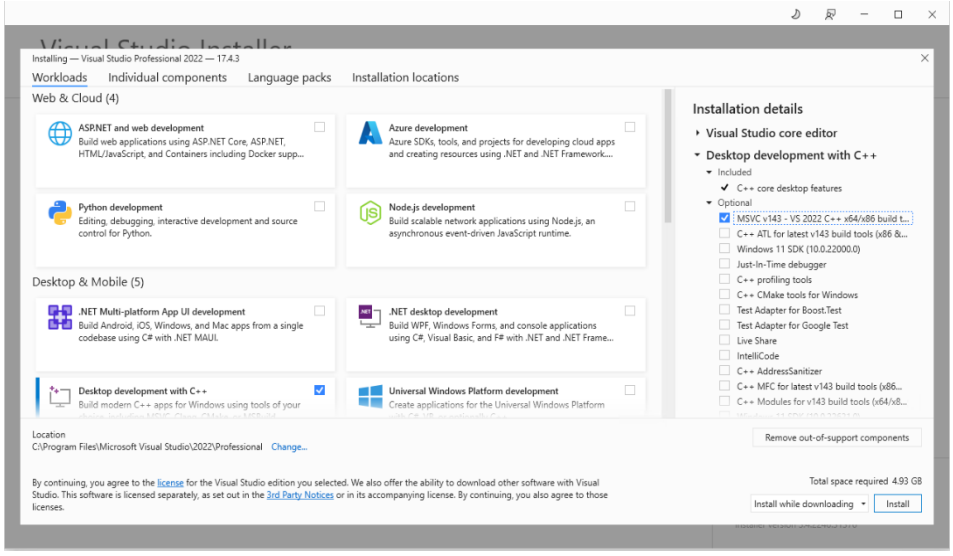
1. You can download Microsoft Visual Studio 2022 from the Microsoft Homepage. You may have to create a Microsoft account to be able to access this page.
2. Select Visual Studio Enterprise or Professional 2022 (version 17.2) and click on **Download**.
3. Once download done, run the file `vs_Professional.exe` or `vs_Enterprise.exe` according to selected Visual Studio version.
4. Accept license terms.

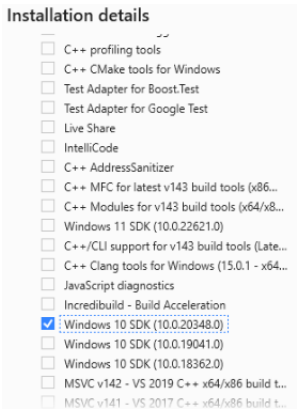


5. Installation starts.

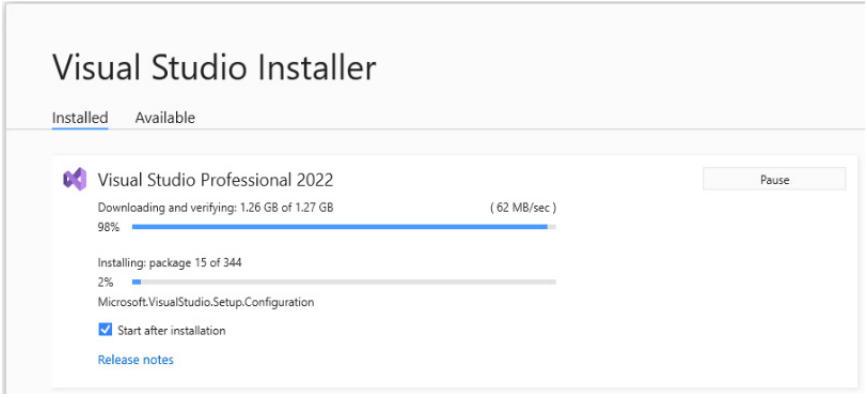


6. Select components to be able to link Digimat/LS-DYNA executables: **MSVC v143 - VS 2022 C++ x64/x86 build tools** and **Windows 10 SDK**.

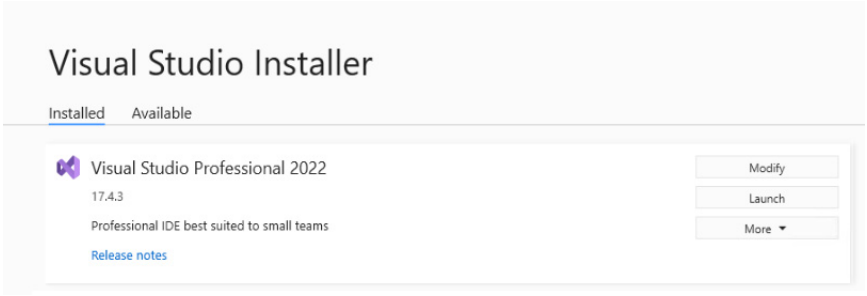




7. Installation starts.



8. Close installation once done.



Microsoft .NET Framework 4.8

Microsoft .NET Framework 4.8 or higher is required to use Digimat-RP, Digimat-VA, and Digimat-AM.

Check the availability of the Microsoft .NET Framework 4.8 from the Windows Control panel (see Figure 6-1).

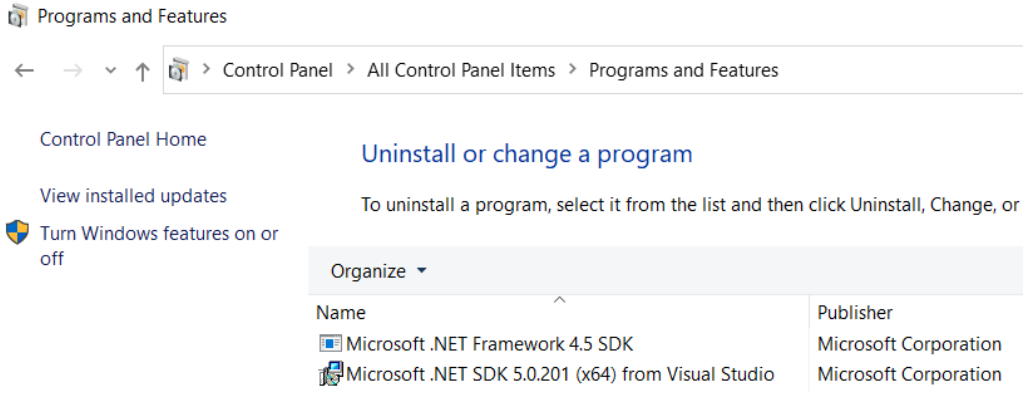


Figure 6-1 Check if Microsoft .NET 4.8 or higher is installed.

If it is not installed, download it from the Microsoft download page and install before using Digimat-RP, Digimat-VA or Digimat-AM.

7 Known Limitations

- Documentation
- Licensing & GUI

Documentation

External Linking

- Links in Examples manual are not functional with Microsoft Edge.
- Some external links may point to data not yet embedded in the manual as delivered with the current version of Digimat. To retrieve the missing file or information of your interest please contact digimat.support@hexagon.com.

Tutorials

- The tutorials are not fully up to date with Digimat version 2023.1. However, as the general workflow should not have drastically changed, the tutorials have been added for an educational purpose.
- Please refer to the examples database to access ready to run Digimat models.
- For more information and further help, please contact digimat.support@hexagon.com.

Licensing & GUI

Digimat Licensing

- Use of `FLEXLM_DIAGNOSTICS` environment variable is not supported.
- Masterkey usage is not supported for Digimat products.
- When using geographically distant license server, it can be needed to define `FLEXLM_TIMEOUT` environment variable. This environment variable is automatically setup in all Digimat graphical user interface and for all Digimat-CAE coupled finite analysis that are run from Digimat-RP.