BETA
SIMULATION SOLUTIONS

physics on screen

BETA
SUITE

Installation Guide

ANSA
PRE PROCESSOR

EPILYSIS
SOLVER

META
POST PROCESSOR

Table of Contents

Section 1
Pre-installation considerations

1.1. Intended audience and scope

This document is written for system administrators or people responsible for the installation and maintenance of the latest commercially available version of the software suite of BETA CAE Systems.

Installation of **BETA CAE Systems Software Suite**, i.e. **ANSA, EPILYSIS** and **META** requires basic knowledge of WINDOWS and/or UNIX-flavour operating systems, as well as familiarity with the basic operations of beta_lm_tools, the license management package of BETA CAE Systems.

The following topics are covered in this document:

- A brief reference to beta_lm_tools
- Minimum hardware requirements
- Installation on Unix-flavored systems
- Installation on Windows systems
- ANSA and META command line options
- ANSA and META run-time settings

1.2. Pre-installation considerations

Prior to installing the BETA CAE Systems Software Suite on a client the following items should be considered:

- Contact BETA CAE Systems (www.beta-cae.com) to request access to the secure www server in order to download the latest license management package and the latest version of ANSA/EPILYSIS/META and KOMVOS installers.

A. On the *server* side:

- Select a machine that will be used as *license server*: It should be a designated machine that resides in the same network as the BETA CAE Systems Software Suite client and onto which a recent version of beta_lm_tools must be installed. The corresponding *license daemon,* called *beta_lm,* will handle the initial contact and communication with the BETA CAE Systems Software Suite client through a TCP/IP network protocol.
- Obtain a valid *license file*: This is a file that contains all information necessary for the uninterrupted use of BETA CAE Systems Software Suite. This file, usually called *license.dat*, is copied on the license server and is read by the license daemon.
- Install the license daemon using the license file

B. On the *client* side:

- Properly configure the TCP/IP services: BETA CAE Systems Software Suite use the TCP/IP network protocol to communicate with the license server. Note that even if the license server is the same machine as the client, TCP/IP is still used.

In order to verify that your TCP/IP connection is working, open a command prompt and type:

```
ping <hostname>
```

where <hostname> is the name of the license server. If you receive a reply from the license server, then the connection is working.

1.3. Brief reference to *beta_lm_tools* and *beta_lm* license daemon installation

This paragraph is intended for users familiar with license management software. Such users can quickly browse through the following steps in order to download and install beta_lm_tools. For detailed information on the functionality and features of beta_lm_tools please refer to the related documentation.

1. Decide on the machine that will be used as license server. If a server redundancy scheme is to be used, decide on its type and on the machines that will be used as license servers.

2. Log on to BETA CAE Systems server and download beta_lm_tools for each machine and platform that will be used as a server.

3. On each of these servers, unpack beta_lm_tools and execute the command:

   beta_lm -host_key

   Store the outcome of the above command, which consists of the beta_lm version, the machine's hostname, the machine's ethernet card MAC address and a string of forty (40) characters, like:

   BETA LM v7.0
   BETA LM Host Name = Gauss
   MAC = 00:0f:b0:43:34:9a
   BETA LM Host Key = 200a025ec00b5f5e0f1b2e1b3d5e020bd04e5144

   On WINDOWS systems only, the administrator has the option to derive the above key based on another network interface, namely USB or WiFi. To produce such a key, use the command:

   beta_lm -host_key -ni [network interface, e.g. USB | WiFi]

4. Provide the above to BETA CAE Systems. Using these data, BETA CAE Systems will generate the corresponding license file, usually called license.dat

5. Copy the license.dat file on each server and install the beta_lm license daemon using the commands:

   Linux Systems:
   beta_lm -f [full_path_to]license.dat -L <[full_path_to]log_file>

   Windows Systems:
   beta_lm -install -f [full_path_to]license.dat -L <[full_path_to]log_file>

   Alternatively, the single name of the license (license.dat) and the license.log files can be used, if navigated to the directory where the license file is located.
   If the administrator used a USB or a WiFi network interface to derive the server host key, then this interface should also be declared during installation:

   beta_lm -install -ni [USB | WiFi] -f [full_path_to]license.dat -L <[full_path_to]log_file>

   Start the "BETA LM Service" from:
   Control Panel > Administrative Tools > Services

   Verify that beta_lm is up and running on each server, using the command:

   beta_lm_stat -h server_name

The outcome of the above command will display the features available within the license.dat file.
Note that in the "hardware failover" redundant server scheme, only the server that is currently the "master" will respond.

1.4. A note on ANSA_SRV environment variable

When the user calls any software product from BETA CAE Systems Software Suite to launch, the respective software product will search through the network for available licenses. To do so BETA CAE Systems Software Suite must first contact the license server. The location of the license server is given to BETA CAE Systems Software Suite through an environment variable called `ANSA_SRV`. The definition of `ANSA_SRV` for Windows platforms and for Unix-flavor is discussed in paragraphs 3.3.3 and 3.4.3 respectively.

Section 2
System Requirements

2.1. Minimum system requirements for BETA CAE Systems Software Suite

The following table presents the minimum operating system requirements for BETA CAE Systems Software Suite v16.x.x and later for various 64-bit platforms and OS:

	64bit OS Version
LINUX	
	glibc 2.3.3
WINDOWS	
	Windows Vista 64 SP1, Windows 7

2.2. Recommended GPU (OpenCL Support) requirements for BETA CAE Systems Software Suite

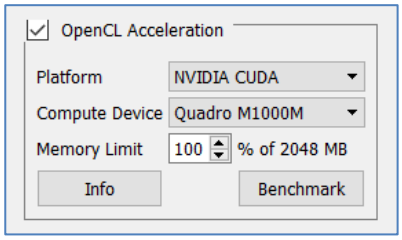
The following table presents the recommended GPU requirements for BETA CAE Systems Software Suite v16.x.x and later:

Platform	**Compute Device**
AMD	
	Radeon HD 7xxx, 6xxx Firepro W series, V series
NVIDIA	
	GTX 4xx, 5xx, 6xx Tesla (all) Quadro 2000 – 6000, k2000 - k5000
Intel*	
	i3, i5, i7

*Intel Platform is also supported, with the prerequisite that *Intel SDK for OpenCL Applications 2012-13* is already installed.

2.2.1 User control of GPU computing capabilities through ANSA

Users are able to control the usage of the GPU computing capabilities, as well as to perform a Benchmark, in order to evaluate the performance of different devices/platforms, via the Windows > Settings > System menu. In case the OpenCL Acceleration is activated, ANSA will use the GPUs, resulting in a high increase of the performance of specific functions.



The user is able to select/specify:

The **Platform**(s) used for the acceleration, the available **Compute Device**, as well as the maximum percentage of the device memory available (**Memory Limit**). The **Info** button is used in order to print the system's information.

With the **Benchmark** button, the user can evaluate the performance of different devices / platforms. Pressing *Benchmark,* ANSA will run a demo test and print the time needed to fulfill it.
! NOTE: The option *OpenCL Acceleration* is available only in those systems that can support this functionality. If this option is not available then the system cannot support this functionality. This could be because the CPU or the GPU are either old or outdated, or there is a problem with the drivers.

Section 3
BETA CAE Systems Software Suite Installation

3.1. Preparation

Make sure that the beta_lm license daemon and the corresponding license.dat file are properly installed on the machines that are specified as license servers. Please refer to the related documentation for additional details on the beta_lm license management system.

3.2. What to download

3.2.1. Download process

In order to get the latest software release of BETA CAE Systems Software Suite follow these steps:

1. Visit the website of BETA CAE Systems: www.beta-cae.com

2. At the top-right hand side, log in to the BETA CAE Systems secure web-server:

3. From the [Downloads] area, access the folder ANSA/EPILYSIS/META.

4. After accessing the latest available version, download the operating system-specific installation file.
Here is the range of files that can be downloaded and an explanation:

- BETA_CAE_Systems_v<version>_<OS>_x64.[sh, msi]
  The installation executable to install ANSA, EPILYSIS and META.
- BETA_CAE_Systems_meta_post_external_libs_v<version>_<OS>_x64.[sh, msi]
  The installation executable to install the external libraries of META which are used to read Abaqus files. If these are not installed and do not exist from previous installations, then META will not be able to read Abaqus files. If a new Abaqus version is supported in a new META version, then the respective external libraries have to be installed so that META will be able to read Abaqus files from this new version.
- BETA_CAE_Systems_meta__viewer<version>_<OS>_x64.[sh, msi]
  The installation executable to install the META viewer. No license is required to run the META viewer.

3.2.2. Integrity verification of the downloaded files

As you will notice when browsing through the folders, each installation file has a respective .md5sum file which can be used to verify that the corresponding file has been downloaded correctly. The verification can be done using the standardized Message Digest 5 (MD5) checksum hash function.

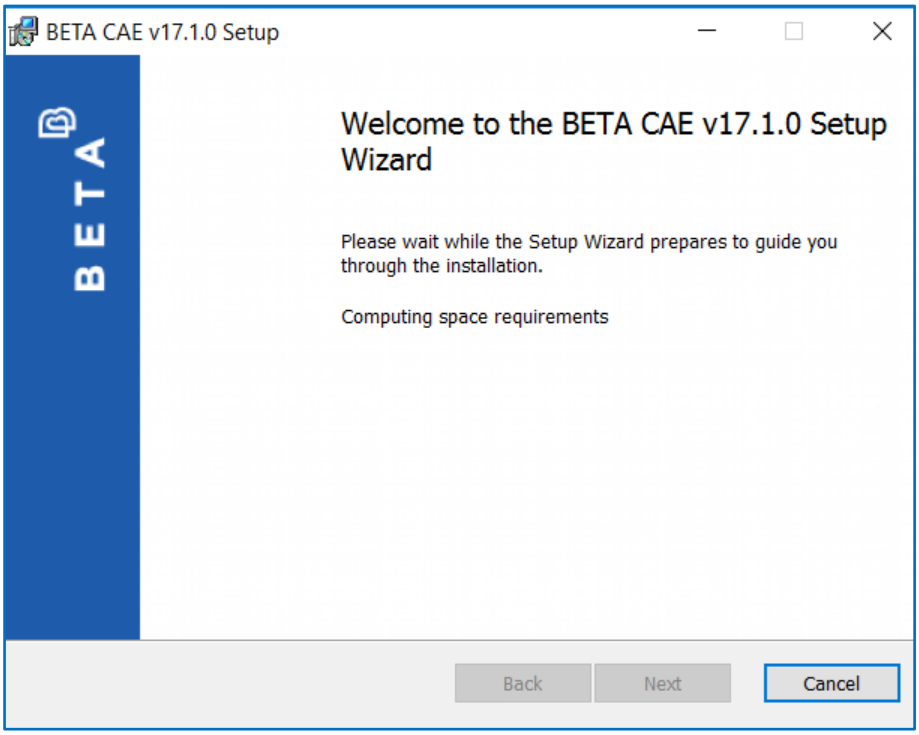
3.3. How to install and launch on MS-Windows

3.3.1. Installation

The installation can be done by either by users locally for their account or by the administrators (recommended) for all the users of a machine.

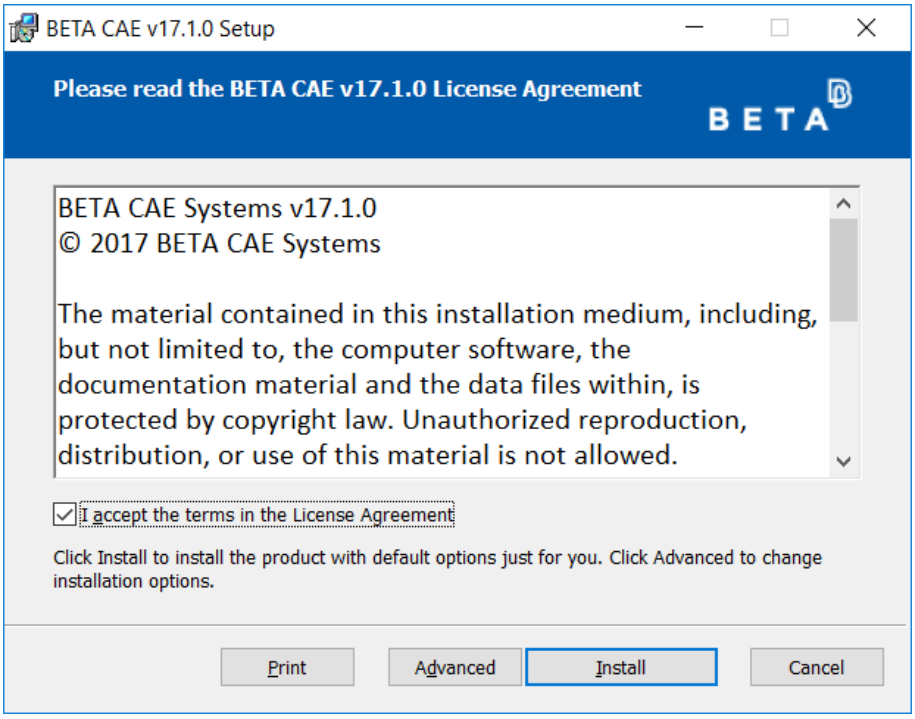
1. Execute the respective executable file.

2. Read-through the License Agreement and check the box below to proceed with the installation in one of two options:



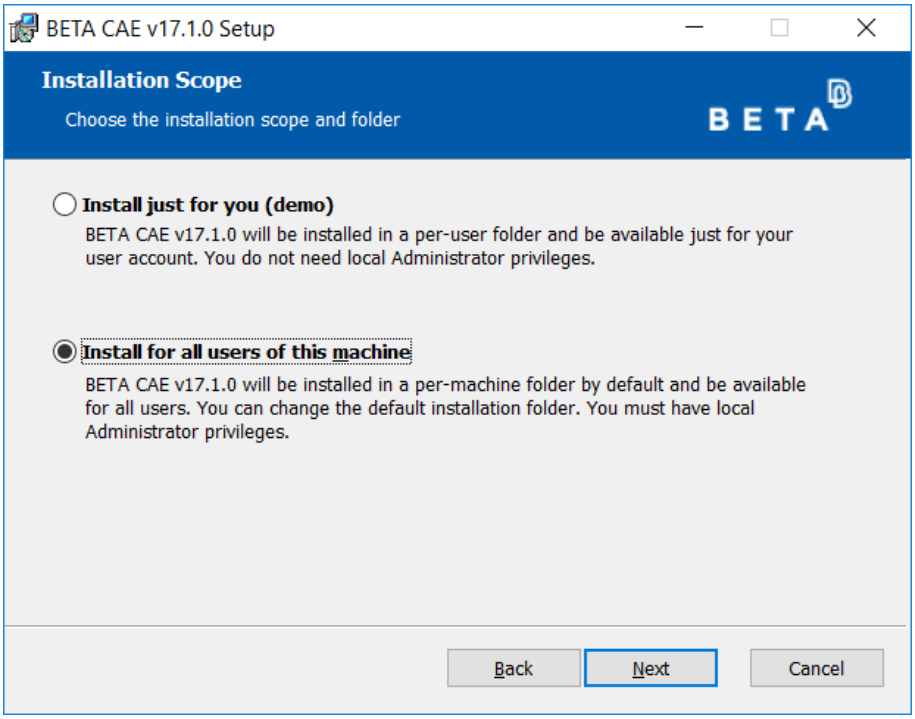
– To specify if the installation will be common to all users and to selectively install software features, press the Advanced button.

– To perform a full features' installation only for the current user, press the Install button:

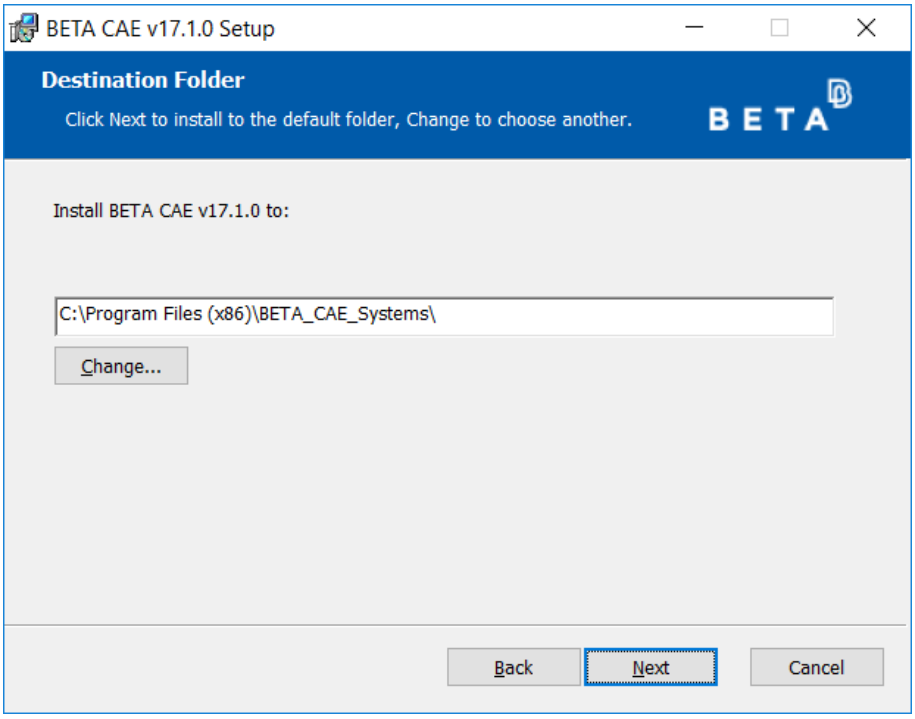


Note: The default installation location is: C:\Users\<username>\AppData\Local\Apps\BETA_CAE_Systems\

3. If the *Advanced* button is pressed and the current user has Administrative account privileges, it is possible to make the installation shared among all users:

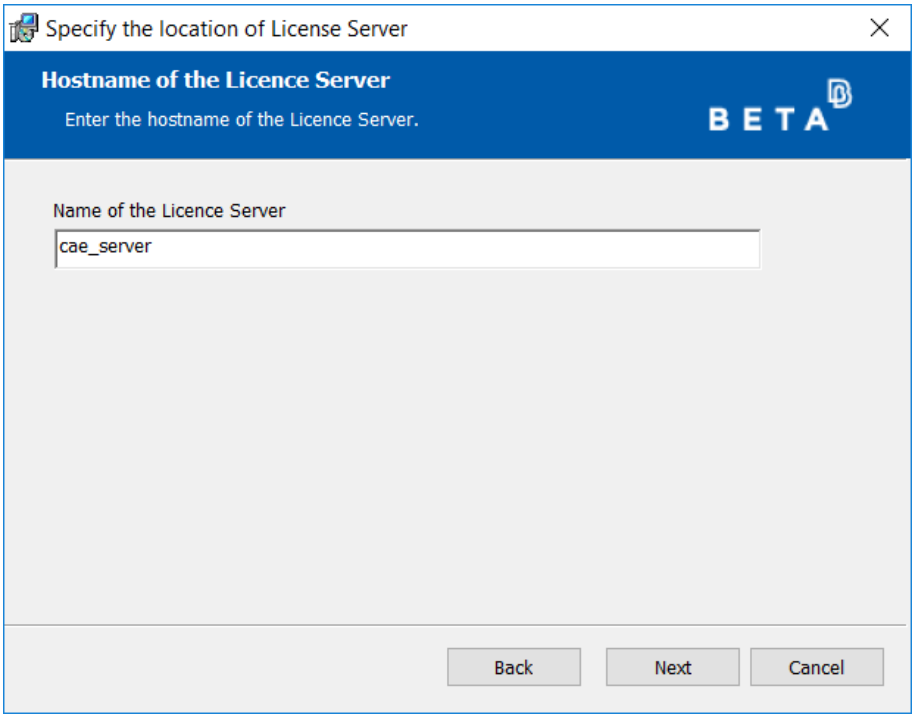


4. Additionally, in the next step, the user has the ability to specify the installation folder:



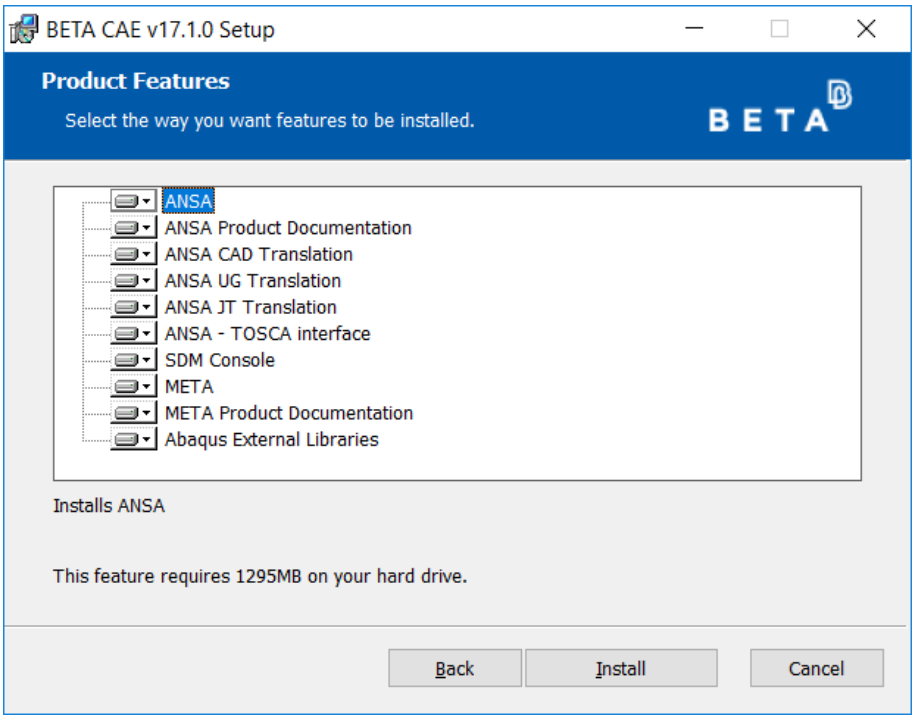
Attention! If the installation is run without Administrative privileges, the steps 3. and 4. are omitted and the installation location is the default C:\Users\<username>\AppData\Local\Apps\BETA_CAE_Systems\

5. Next, the user is prompted to define the hostname of the license server. *This definition is optional.*

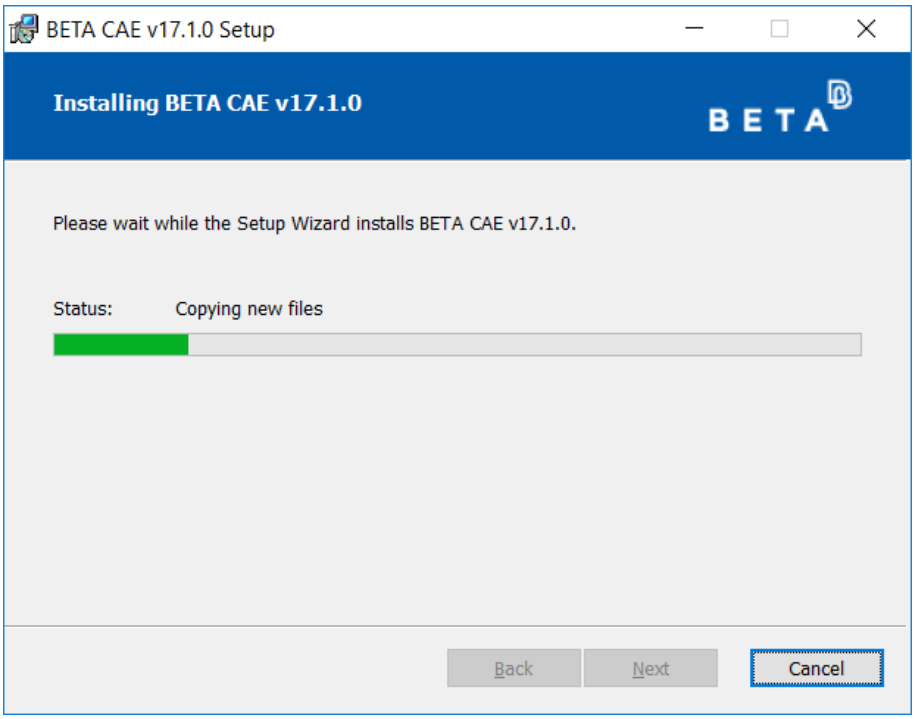


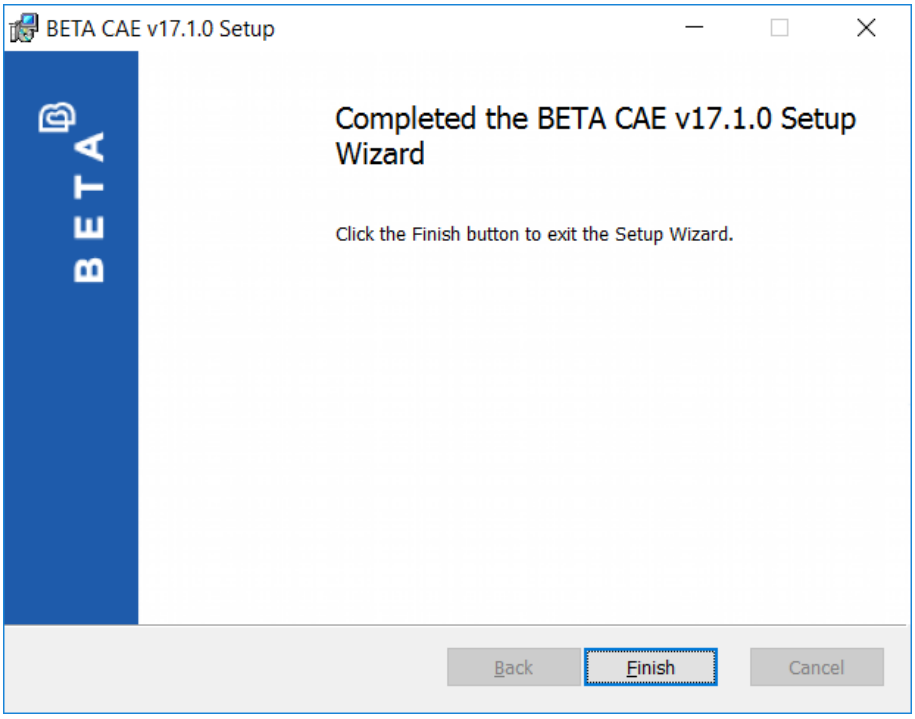
Note: When in redundant scheme, ALL server hostnames need to be specified, separated by commas.

6. The next step is to display the features list. By default, all features are installed, but it is possible to selectively disable the unnecessary ones, as shown in the following snapshot:



1. Press the Install button to complete the installation.





The installation can be performed either interactively using the Graphical User Interface of the installer or by using the command prompt in batch mode, which requires administrative privileges.

To run the installer in batch mode via the command prompt, the following one-line command can be used:

```
msiexec.exe /quiet /i
[full_path_to]\BETA_CAE_Systems_v17.1.0_win_x64.msi
ADDLOCAL=AnsaPre,AnsaDocs,AnsaCAD,MetaPost
LICENSESERVERVALUE=ansa_srv123
```

In order to get a log of the installation, append `/Log install_log.txt` to the previous command. Consequently you can check the log file for the process and completion of the installation.

`ADDLOCAL` takes a list of features to install. Available Features are:

1. `AnsaPre`
2. `AnsaDocs`
3. `AnsaCAD`
4. `AnsaUG`
5. `AnsaJT`
6. `AnsaTOSCA`
7. `MetaPost`
8. `MetaDocs`
9. `AbqExternaLibs` – Not since v21.0.0.

`LICENSESERVERVALUE` is the value that ANSA_SRV is set to.
`APPLICATIONFOLDER` is the location of the installation folder.
`ALLUSERS=1` option allows the installation to all users, provided that you are logged in as Administrator.

3.3.1.1. Uninstall process

Since version 14.x.x, BETA CAE Systems Software Suite is installed on Windows systems using the Windows Installer mechanism by running an *.msi file.

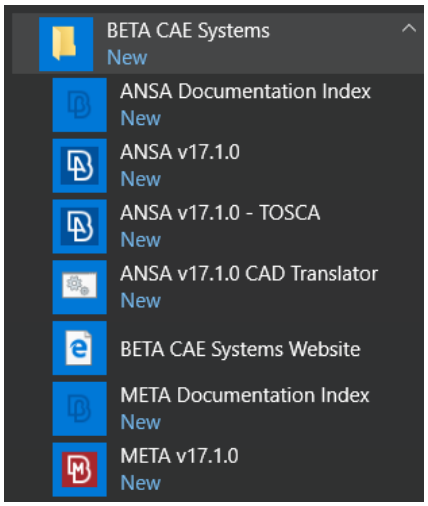
If you have installed versions later than v14 by the installer, you should open the *Control Panel>Programs>Programs and Features* window and select the ones you wish to uninstall.
If they have been installed by an administrator, you must use an administrator account to remove them.
If you have installed them as an ordinary user – i.e. locally in your home folder – you can also uninstall them without administrative privileges.

To uninstall a version via the command prompt as an administrator, you should issue the following one-line command.

```
msiexec.exe /quiet /uninstall
[full_path_to]\BETA_CAE_Systems_v17.1.0_win_x64.msi
```

3.3.2. Launching on MS-Windows

After a successful installation, a software group is created in the Windows Start Menu:



Select the corresponding software feature to launch.

3.3.3. The ANSA_SRV environment variable

As an introduction, when the end-user launches an application, this application will search through the network for available floating licenses. To do so, the licensed application must contact the license server which is indicated at the SERVER line of the license file. If this fails, then the application will continue to search in an orderly fashion at the following locations:

#Location	Notes
1.**BETA_LIC_FILE**	Environment Setting
2.**ANSA_HOME (or META_HOME)**	Environment Setting
3.**current directory**	
4.**BETA_LIC_SRV**	Environment Setting
5.**ANSA_SRV**	Environment Setting
6.**host with alias BETA_LIC_SRV**	
7.**host with alias ANSA_SRV**	
8.**localhost**	

If the search fails in all above locations, the application will not start.

The end-user has alternatively the option to declare a specific license server by its name or IP-address, using the -h flag:

```
application -h port@server
```

for example:

```
ansa -h port@server
```

When a redundancy scheme is used, the above command should be like:

```
ansa -h port@server1,port@server2,port@server3
```

In its simplest form, a licensed application needs to know where in the network the ANSA_SRV environment variable points to. In turn, ANSA_SRV should point to the server or servers used by the current servers (single server or redundant servers), as these are listed in the corresponding license.dat file.

Related examples are given in the following paragraphs.

3.3.3.1. Single server scheme

In a single server scheme the SERVER= line of the license.dat file indicates the server and port used by beta_lm to communicate with clients. For example:

```
SERVER=plank,PORT=6007
```

Consequently, on Windows platforms, the ANSA_SRV environment variable should be set as:

```
set ANSA_SRV=6007@plank
```

3.3.3.2. Redundant server scheme

beta_lm supports two server redundancy schemes; one uses multiple servers for license distribution and one is for hardware failover protection. The scheme which is currently used is reflected in the license file (license.dat) delivered by BETA CAE Systems to the customer:

```
SERVER=gauss,PORT=6007
SERVER=hilbert,PORT=6007
SERVER=riemann,PORT=6007
```

In such cases, the licensed application needs to know all alternate servers that may provide the requested license. There are three alternative ways to achieve this. Speaking of Windows platforms:

(a) Set the environment variable ANSA_SRV to point to all alternate servers:

```
set ANSA_SRV=6007@gauss,6007@hilbet,6007@riemann
```

(b) Keep a copy of the license.dat file into the client machine where the licensed application is called to run, and run the application using the -f option to point to license.dat e.g.:

```
ansa -f license.dat
```

(c) Keep a copy of the license.dat file into the client machine where the licensed application is called to run, and set the environment variable BETA_LIC_FILE to point to license.dat e.g.:

```
set BETA_LIC_FILE=license.dat
```

A note on the server searching order

When more than one servers are declared in the ANSA_SRV variable, the client application will request for a license by checking these servers in a specific search order: At first, the client will check the first declared server and then will start checking backwards from the last server to the second.

As an example consider a server scheme consisting of n servers. On Windows platofrms, the respective ANSA_SRV declaration should be:

```
set ANSA_SRV=server1,server2,....server(n-1),server(n)
```

In such cases the client will check the servers in the following order:

1.	server**1**	(first declared server)
2.	server**(n)**	(last declared server)
3.	server**(n-1)**	(second to the last server)
	.....	
n.	server**2**	(second server)

3.4. How to install and launch on Linux

3.4.1. Installation

1. After the download, you need to give the installation executables the execute permission (from terminal):

```
chmod +x BETA_CAE_Systems_v17.1.0_linux_x64.sh
```

2. Follow the step-by-step prompts to complete the installation.

3.4.2. Launching on Linux

In order to launch any software feature on a Linux platform, the user should locate the installation folder and use the corresponding .sh launching script. To launch ANSA for example, use ansa64.sh for the 64bit version, followed by the desired options (if any):

```
ansa64.sh [options]
```

The above script identifies the current OS and subsequently launches the appropriate software executable. It is strongly recommended to use this script so that all parameters or files required at start-up (e.g. location of configuration files or additional library files) are correctly set.

Run from terminal:
`BETA_CAE_Systems_v17.1.0_linux_x64.sh --help`
for a complete list of options.

When the software is called to run it will search through the network for available licenses. To do so, BETA CAE Systems Software Suite must first contact the license server, i.e. the machine where the beta_lm license daemon is installed. The location of the license server is given to BETA CAE Systems Software Suite through an environment variable called ANSA_SRV. See below for details.

Note: The user can also install the Unix package (e.g. BETA_CAE_Systems_v17.1.0_linux_x64.sh) from Windows workstation, under CYGWIN environment.

3.4.3. The ANSA_SRV environment variable

The definition of ANSA_SRV on Linux platforms is given through the setenv command as follows:

```
setenv ANSA_SRV port@server
```

where server is the machine hosting the license daemon and port is the TCP/IP port that will be used for the communication between ANSA and the server. If the port number is omitted then the default value of 6007 is considered.

The above describes the simple case when the license daemon is installed on a single (1) server. However, many companies use a “hardware failover” redundancy scheme that consists of three (3) servers, all having beta_lm installed and running. In this scheme, one of the servers becomes *master* and the other two become *slaves*. License requests are handled only by the *master* and, if the *master* fails, one of the *slaves* will take over as the new *master*. In such cases ANSA needs to know all three alternative servers, hence they need to be declared in the ANSA_SRV variable:

```
setenv ANSA_SRV port@server1,port@server2,port@server3
```

Again, if no port is used, the default value of 6007 is considered.

Recommendation on where to define ANSA_SRV

The definition of the ANSA_SRV variable on a client should be given *globally*, along with other environment variables that the client uses. Alternatively, each user can declare ANSA_SRV in his private account workspace, for example in the .login file or in the resource files of his preferred `shell` (`.bashrc`, `.cshrc`, `.tcshrc`).

Not recommended alternatives

A user can also edit the ANSA/META launching scripts (ansa.sh, ansa64.sh) and modify line:

```
if($?ANSA_SRV == 0) setenv ANSA_SRV ansa_srv
```

to

```
if($?ANSA_SRV == 0) setenv ANSA_SRV port@server
```

where server is the machine hosting the license daemon and port is the TCP/IP port that will be used for the communication between ANSA/META and the server. In hardware failover server redundancy schemes all servers must be declared.

Section 4
ANSA command line options

4.1. ANSA command line options

The following command line options are available in ANSA:

Option	Description
`-help` `--help`	Print help, with the available command line options, and exit.
`-v` `--version`	Output version information and exit.
`-gui <name>`	Start ANSA with predefined Layout GUI Settings as these are saved in the `file.xml` within `.BETA/ANSA/version_xx.y.z/` in the user's home directory.
`-xml <filename>`	Start ANSA using custom GUI settings from `filename`
`-nolauncher`	Open with last visited layout.
`--skip-copyingPDS`	Does not copy defaults and script files from previous version of ANSA, despite of their existence.
`--copyingPDS`	Copy defaults and script files from the previous version of ANSA by asking your confirmation.
`--copyingPDS-withoutasking`	Copy defaults and script files from the previous version of ANSA without asking your confirmation.

`-exec <argument>`
`-execpy <argument>`
`-execbs <argument>`
	Start ANSA and execute specified script commands one after the other:
	`ansa64.sh -execpy "load_script: 'script.py'" -execpy "function1(11, 4)" -execpy "function2(10, 5)"`
	In this case ANSA will start, will load the script `script.py` and then execute the functions `function1` and `function2`.
	Using these options implies the use of `-foregr` as well.
	
	There are some notable differences between these commands:
	• `-exec` can be used for Python or BETAscript language scripts. Optional function arguments are not supported.
	• `-execbs` can be used for BETAscript language scripts only. Optional function arguments are not supported.
	• `-execpy` can be used for Python language scripts only. Optional function arguments are supported.

`-execscript '<script>|<function>([args])'`	Start ANSA, load a script and run a function with the given arguments. This option can be used multiple times. `ansa64.sh -execscript '/scripts/script.py|function(5, 32)'`
`-transl <filepath>`	Search in the default settings locations for the file with the specified `filepath` and use it as a `TRANSL` file. The `filepath` can include a relative path. For example: `ansa64.sh -transl /myscripts/group.py` will search for a `TRANSL` file in the following paths: • `$ANSA_HOME/myscripts/group.py` • `/.BETA/ANSA/version_xx.y.z/myscripts/group.py` inside the user's home directory • `/myscripts/group.py` inside the current working directory.
`-translfp <filepath>`	Load the TRANSL file given in `filepath`. The programming language is determined by the filename's extension.

Option	Description
-nogui -b	Pure batch background operation (no ANSA window appears). It can be used in two ways: • with no other arguments ansa64.sh -nogui or > ansa.sh -b In this case, ANSA will search inside the ANSA_TRANSL script to locate the autoexec command and execute any user-script functions that are called for automatic execution. • in combination with –exec ansa64.sh -nogui -exec or > ansa.sh -b -exec
--batch-mode	Run ANSA without asking user, via prompts, for additional actions or confirmation.
-virtualx	Start ANSA using a Virtual X Server.
-virtualx_64bit	Start ANSA using a 64-bit Virtual X Server.
-virtualx_geometry <geometry>	Set Virtual X Server resolution (default 1280x1024).
-virtualx_dpi <dpi>	Set Virtual X Server “dpi” (default 75).
-virtualx_secure_password_file <filename>	Set Virtual X Server vnc password file.
-no_egl_offscreen	Disable the use of EGL off-screen hardware and OpenGL rendering in remote execution.
-force_egl	Force the use of EGL off-screen hardware OpenGL rendering in remote execution.
-egl_gpu <gpu_id>	Select GPU with gpu_id to use for EGL off-screen rendering. Use -1 for auto-select.
-i <filename>	Start ANSA and load a file.
-if <deck name>	Specify the format of the input file (used with -i).
-s <filename>	Start ANSA and execute a list of commands written in the session file <filename>.
-plugins <path>	Specify additional plugins directory from where ANSA will read additional plugins. If a conflict occurs, user plugins have priority.
--undo <yes/no>	Enable/Disable Undo functionality.
--cad-options	Display CAD Translators options.
--jt_open	Start ANSA with ANSA_JT_OPEN License Feature.
--nx_open	Start ANSA with ANSA_UG License Feature.
--rhino_open	Start ANSA with ANSA_CATIA_V4.
--cad_open	Start ANSA with General CAD Translation License Feature.
--multimodel <yes/no>	Switch that enables or disables multi modeling functionality.
-connect <port>	Connect to ANSA driver listening on this port. One of -nogui, -b options must also be provided.
-listenport <port>	Listen for ANSA driver on this port.
--websrv	Activate Web Interface on startup.
--websrv-backend-port <port>	Set the Web Interface backend port. If left unset or set to 0, a random available port will be selected.
--websrv-port <port>	Set Web Interface server port. If left unset or set to 0, a random available port will be selected.
--websrv-jail <directory>	Set the Web Interface directory jail. If left unset, the jail directory points to the working directory.
--websrv-autoclose <timeout msec>	Set the Web Interface timeout. If left unset, the web interface will close after 2000msec. If set to 0, the web interface timeout is disabled.
--websrv-pkey <filename>	Set the Web Interface private key to be used with https connections. In order to use https, both --websrv-pkey and --websrv-cert must be set.
--websrv-cert <filename>	Set the Web Interface certificate to be used with https connections. In order to use https, both --websrv-pkey and --websrv-cert must be set.
--clb-connect <ip:port>	Connect to an existing Collaboration server.
--clb-start-connect <ip:port>	Start new Collaboration server and connect to it.
--collaboration-config <string>	Enter the collaboration hub configuration.
--developer-license	Run under Developer License with reduced functionality.

Option	Description
`--rasterizer <algorithm>`	Use a CPU software renderer in case of unsupported or missing GPU. `algorithm` has to be `mesa-llvm`. `ansa64.sh –rasterizer mesa-llvm`
`-h <port@server>`	Use license server `server` on port `port`.
`-l`	Start ANSA occupying just a portion of the screen.
`-g <dim1xdim2>` `-geom <dim1xdim2>`	Start ANSA in a `dim1xdim2` window. For example: `ansa64.sh -g 1280x800` Minimum dimensions are `100x100` and maximum dimensions depend on user screen size and resolution.
`-ncl` `--noopencl`	Start ANSA without OpenCL.
`-foregr`	Execute as a foreground process in Linux, i.e. run directly and wait for it to complete, preventing you from doing other work in the terminal.
`-changedir <pathname>`	Change current working directory to `pathname`.
`-dmroot <pathname>`	Specify the folder `pathname` as the DM Root.
`-dmusername <username>`	Specify the username used for connecting to DM.
`-dmpassword <password>`	Specify the password used for connecting to DM.
`-dmrole <role>`	Specify the role for connecting to DM.
`-script_help <filename>`	Specify a help file for script editor.
`-ideas_corr`	Start ANSA and activate special treatment for IGES, VDA and VDA-FS v5.2 files generated by I-DEAS that contain errors at the parameterization of FACE's bounding curves.
`-lm_retry <seconds>`	Time in seconds that ANSA has to wait before re-trying to get a license in case of a license shortage.
`-lm_timed`	Use license server timed based credits. This requires BETA LM version v7.0 or newer and works only in batch mode.
`-np <cpu number>`	Specify the number of processors/threads that ANSA will use. If not defined, all processors are used.
`-fix_quadro_bug`	Workaround Nvidia Quadro bug.
`--no-xinitthreads`	Disable XInitThreads.
`-reduce_memory`	Reduce memory usage at about 25% with a negative performance impact.
`-vram <size_in_kbytes>`	Use this amount of video memory size in kilobytes. It is automatically detected be default.
`-df <filename>` `--defaults-file`	Read a defaults file with ANSA settings.
`-dft <filename>` `--defaults-translator-file`	Read a defaults file with CAD translators' settings.
`--icons-path <dirname>`	Set a directory where you can retrieve icons from.
`-performance_mode <mode>`	Adjust drawing performance and memory usage from these options: • 0: Default setting, fastest drawing, highest memory usage. • 1: No GPU memory usage. • 2: No GPU memory usage. • 3: Use CPU software renderer in case of unsupported or missing GPU.
`--no-performance-buffers`	Disable performance buffers. Use this parameter in case of any OpenGL instabilities.
`--iconify`	Start ANSA minimized.
`--confirm-license-agreement`	Confirm the license agreement without displaying it.
`--skip-release-highlights`	Do not show the Release Highlights window.
`--proxy <url>`	Enable the use of proxy. If left empty then the system's configured proxy will be used.

Section 5
META command line options

5.1. META command line options

The following command line options are available in META:

Option	Description
`-help` `--help`	Use this option to have all available command line options displayed in text.
`-v` `--version`	Display the software version information

`-h <port@server>`	Try to obtain a license from a different server than the one specified in the *ansa_srv* environment variable:
	Example:
	`meta -h 6007@beta_lic_server`
	`meta -h 6007@198.51.100.1`

`-lm_retry <time>`	Use this running option when there are no more licenses available for starting META, to specify the time interval in seconds for retrying to acquire a license.
`-lm_timed`	Use license server timed based credits. This requires BETA LM version v7.0 or newer and works only in batch mode.
`-l`	Start META to occupy only a portion of the screen.
`-m`	Start META in a maximized window.
`-nolog`	Do not create a META_post.log file.
`-noses`	Do not create a META_post.ses file.
`-nolauncher`	Start META without opening the launcher
`-noabaqusmessages`	Upon starting-up META prohibit displaying messages in the terminal window related to missing Abaqus libraries.
`-nooverlay`	Run META with the background window redraw option inactive. This function is useful when the operating system's window manager is not fully compatible with the overlay functionality. This option is applied on SUN workstations by default, if META is run from the provided script file.
`-g <WxH>` `-geom <WxH>`	Start META and set its window dimensions to WxH pixels. `meta -geom 1280x815` The minimum dimensions are 500x500, and maximum dimensions depend on the current screen resolution and monitor.
`-position <X,Y>`	Start the META window with its top-left corner at the specified X and Y screen position, measured in pixels.

Option	Description

`-d <path>`	After starting META the file manager of the *Read Results* card and the *Script Editor* will point to the specified path:
	`meta –d <path>`
	The above command line option may be substituted by setting an environment variable in META's start-up scripts.
	For Linux, in `meta_post64.sh`:
	`setenv POST_DIR <path>`
	For Windows, in `meta_post64.bat`:
	`set POST_DIR=<path>`

`-changedir <path>`	Change current working directory to `pathname`. Note that this setting affects the execution of sessions that include commands with relative paths.

`-groupdir <path>`	Use this option to define the Group directory.
	`meta -groupdir <path>`
	The above command line option may be substituted by setting an environment variable in META's start-up scripts.
	For Linux, in `meta_post64.sh`:
	`setenv META_GROUP_DIR <path>`
	For Windows, in `meta_post64.bat`:
	`set META_GROUP_DIR=<path>`

`-viewsdir <path>`	Use this option to define the Views directory.
	`meta -viewsdir <path>`
	The above command line option may be substituted by setting an environment variable in META's start-up scripts.
	For Linux, in `meta_post64.sh`:
	`setenv META_VIEWS_DIR <path>`
	For Windows, in `meta_post64.bat`:
	`set META_VIEWS_DIR=<path>`

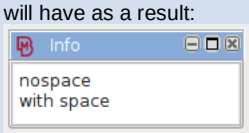
`-reportdir <path>`	Define the Report output directory.
	`meta -reportdir <path>`
	The above command line option may be substituted by setting an environment variable in META's start-up scripts.
	For Linux, in `meta_post64.sh`:
	`setenv METAPOST_REPORT_OUTPUT_DIR <path>`
	For Windows, in `meta_post64.bat`:
	`set METAPOST_REPORT_OUTPUT_DIR=<path>`

`-templatedir <path>`	Define the Report templates' directory.
	`meta -templatedir <path>`
	The above command line option may be substituted by setting an environment variable in META's start-up scripts.
	For Linux, in `meta_post64.sh`:
	`setenv METAPOST_TEMPLATE_DIR <path>`
	For Windows, in `meta_post64.bat`:
	`set METAPOST_TEMPLATE_DIR=<path>`

Option	Description

`-listenport <port>`
`-responseport <port>`
`-sendport <port@server, ...>`
	It is possible to control META from a remote host, in a client-server scheme. In such a scheme, all commands applied to the sending peer (client), will be sent and repeated in the receiving peers (servers). In this scheme, first the servers have to be set, and then the client. Otherwise the connection will be refused.
	To run META as the client, use the running option:
	`meta -sendport <port@host1,port@host2, ...>`
	To run META as the server, use the running option:
	`meta -listenport <port>`
	To write a log to a port, use the running option:
	`meta -responseport <port>`
	Alternatively or in parallel to running a META client for controlling some META servers, it is also possible to drive the META servers by sending META commands directly through a terminal. The command that should be applied in this case is:
	`${META_POST_HOME}/utils/meta_remote_control.sh <port@host1, port@host2, ...> "<META command>"`
	Remarks
	- It is also possible to interact directly with any META server that is receiving remote commands, albeit affecting only the META server.
	- The Client/Server scheme can be set also from within META, by applying the commands:
	`remote listen port`
	`remote send port@server1, port@server2,...`

`-fastses`	Use this option when running a session file to skip screen redraws which will result in META will executing the session faster.
	`meta –fastses -s <session_filename>`
	This running option has the same effect as including the following commands at the start and end a session file:
	`options session controldraw disable`
	`options session controldraw enable`

`-s <session_file>`	Start META and run the specified session file – a text file of META commands:
	`meta –s test.ses`
	If this command is used, then a series of arguments can be passed after all other arguments and made available as variables inside META, where `i` is the index of the argument in the command line, starting at `0`. Example:
	`meta –s test.ses –m nospace "with space"`
	Then, the variables `0` and `1` can be used in session for example as filenames.
	For example the following commands:
	`options message $0`
	`options message $1`
	will have as a result:
	 

`-xml <xml file>`	Use this option to start META and load a specific configuration (GUI layout, default settings, etc) XML file. For example, to load the settings of *some_file.xml*: `meta –xml settings_file.xml`
`--session-id`	Appends the process id to session's and log's filenames. This permits to run multiple META sessions in the same directory without conflicts.
`-script_help <filename>`	Use this option to specify a custom script help file.
`--websrv`	Activate Web Interface on startup.
`--websrv-backend-port <port>`	Set the Web Interface backend port. If left unset or set to 0, a random available port will be selected.

Option	Description
`--websrv-port <port>`	Set Web Interface server port. If left unset or set to 0, a random available port will be selected.
`--websrv-jail <directory>`	Set the Web Interface directory jail. If left unset, the jail directory points to the working directory.
`--websrv-autoclose <timeout msec>`	Set the Web Interface timeout. If left unset, the web interface will close after 2000msec. If set to 0, the web interface timeout is disabled.
`--websrv-pkey <filename>`	Set the Web Interface private key to be used with https connections. In order to use https, both `--websrv-pkey` and `--websrv-cert` must be set.
`--websrv-cert <filename>`	Set the Web Interface certificate to be used with https connections. In order to use https, both `--websrv-pkey` and `--websrv-cert` must be set.
`--proxy <url>`	Enable the use of proxy. If left empty then the system's configured proxy will be used.
`--clbconnect <ip:port>`	Connect to an existing Collaboration server.
`--clb-start-connect`	Start new Collaboration server and connect to it.
`-nopolyoffset`	Use this option to overcome any problems with the display of lines (i.e. in Wireframe mode).
`-foregr`	Execute as a foreground process in Linux, i.e. run directly and wait for it to complete, preventing you from doing other work in the terminal.
`-64`	Use this option to run the 64 bit version for Unix.
`-viewer`	Use this option to run META in Viewer mode, which requires no License credits.

`-virtualx`
`-virtualx_64bit`
	Run META on a Linux workstation that does not have a running X-server, virtual or real. For example:
	`meta -virtualx_64bit -b -s test.ses`
	will run META in batch mode and execute the test.ses session and will wait for it to finish. It is advisable to use the X-server that matches the host OS architecture (32/64 bit). There is also an older method of running META, available via the following executable shell scripts, found in the `${META_POST_HOME}/utils/linux` directory:
	`meta_post_vnc.sh`
	`meta_post_vnc_nobatch.sh`
	`meta_post_vfb.sh`
	When running META with the `-virtualx` option, the 32-bit glibc libraries are also needed.

`-virtualx_fontpath <directory>`	Used with the `-virtualx` running option to specify the path to the fonts that X-server will use.
`-virtualx_geometry <geometry>`	Used with the `-virtualx` running option to run the X-Server with a specific window size.
`-virtualx_dpi <dpi>`	Used with the `-virtualx` running option to run the X-Server with a specific dots-per-inch (dpi) setting.
`-virtualx_secure_password_file <filename>`	Set Virtual X-Server VNC password file for restricted access. - If the file does not exist, a password is requested and then saved in the file. - If the file exists, it is used for authentication.
`-no_egl_offscreen`	Disable use of EGL off-screen hardware OpenGL rendering in remote execution. This option significantly improves performance when saving images on a remote machine using `ssh -X`. A prerequisite for this is to use an NVIDIA driver newer than version 335.
`-force_egl`	Force the use of EGL off-screen hardware OpenGL rendering in remote execution.
`-egl_gpu <gpu_id>`	Select GPU with `gpu_id` to use for EGL off-screen rendering. Use -1 for auto-select.

`-b`
`-nogui`
`--batch-mode`
	Run META in batch mode. This option should be followed by `-s` option in order to specify the session file to run.
	`meta -b -s file.ses`

`-e`	Use this option when running META in batch mode in order to pause the session if an error is found. `meta -b -e -s file.ses`

Option	Description
-stereo <mode>	Use this option to run META in stereoscopic view mode. mode can be set to either quadbuffer or split. The default viewing distance is set to 60 [mm]. If the model is in meters, the eye distance needs to be set accordingly. meta - stereo quadbuffer To switch the stereoscopic mode off while in META use the command: view stereo off
-deck <option>	Use this running option to set the default deck. option can be: nastran, dyna, pamcrash, radios, abaqus, madymo, patran, metadb.
-batch_hw	Forces the use of a hardware OpenGL renderer when in batch mode. This option is obsolete since version 17.1.0, where it is activated by default when in batch mode.
-batch_sw	Forces the use of a CPU software renderer when in batch mode.
-dmroot <pathname>	Use this option to define the DM META directory.
-mountmap	Set a map between Linux and Windows mount points. META will then substitute any references to these mount points in commands as necessary. Example: meta –mountmap "\\share\files:/share/files" meta –mountmap "Z:\:/share/files"
-reset_gui	Use this running option to reset the *META_post.xml* file.
-p <path>	Use this option to load a project file upon start up: meta –p /path/to/project.metadb
-t <template file> <template arguments>	Load geometry and results using a META template file. meta -t "template.csv" "GEOM=input.key"
--developer-license	Start META using a Script Development License.
--rasterizer <algorithm>	Use a CPU software renderer in case of unsupported or missing GPU. algorithm has to be mesa-llvm. meta –rasterizer mesa-llvm
-fix_quadro_bug	Enable a workaround for issues experienced running META on Nvidia's Quadro® series of graphics cards.
--no-xinitthreads	Avoid using XInitThreads. A bug in certain X11 Window System versions for Linux would cause META to freeze.
-ncl --noopencl	Run META with disabled OpenCL.
-exec <command1;command2; ...>	Option to execute some META commands upon start up.
-VS<var_name>=<var_value> -VN<var_name>=<var_value>	Define a META-variable and its value, where: This argument can be used multiple times. If a variable is defined more than once, the last definition will overwrite the previous ones. Examples: meta –VNcurve_id=5 meta –VSspace_use="string with spaces"
-reduce_memory	This option will have the following effects on META when the *Nodal Calculation* setting in the *Scalar* tab of the *Read Results* window is set to *Average*: a) 25% less memory consumption for storing deformation results. b) 5-10% faster reading of deformation results, with some negative impact on performance when animating. If this option is used in combination with performance modes 0 and 1 (see command line option -performance_mode), the performance mode will automatically switch to 2.
-keep_nodal_scalar	Option for storing in memory the Nodal Scalar results. This option will improve the animation drawing performance with a penalty in memory consumption.
-np <number>	Running option for specifying the number of CPUs that META will occupy.
-vram <size_in_kbytes>	Use this amount of video memory size in kilobytes. It is automatically detected be default.

Option	Description
`-performance_mode <mode>`	Adjust drawing performance and memory usage from these options: • 0: Default and fastest setting, full GPU acceleration, GPU memory usage. • 1, 2: Slower setting, CPU memory usage. • 3, 4: Use software rendering.
`--confirm-license-agreement`	Confirm the license agreement without displaying it.

Section 6
Interaction of ANSA-META with other Software

6.1. Interaction with TOSCA Structure

A special environment of ANSA and META is available for the TOSCA Structure integration. This environment provides limited functionality of ANSA/META and requires a special license. However, the standard ANSA/META installation provides full functionality of ANSA/META – TOSCA integration.

To set up the TOSCA Environment do the following:

1. Install BETA CAE Systems Software Suite as described in Sections 1 to 6.
2. Set up the environment variable tosca to the installation directory of TOSCA.
3. Launch ANSA/META using the -gui TOSCA option. In ANSA case, for example:

```
> ansa.sh -gui TOSCA
```

   Alternatively, and depending on the hardware platform, use the `ansa_tosca.bat`, `ansa_tosca.sh`, `ansa_tosca64.bat`, or `ansa_tosca64.sh`

A `TOSCA.xml` and `TOSCA.defaults` files are located at the `/config` subfolder of the installation directory. These files control the TOSCA environment. The subfolder `/config /images/` contains all icons that are used to the TOSCA environment.

When no `ANSA_HOME` environment variable has been defined the `xml` and `defaults` files are read from the `/config` subfolder.
If `ANSA_HOME` is set up the /images directory should be copied at the `ANSA_HOME` location.

In order to define the solver command that will be used during the validation run, edit the `TOSCA.defaults` file and fill-in the corresponding line:

```
NASTRAN_COMMAND =
ABAQUS_COMMAND =
ANSYS_COMMAND =
PERMAS_COMMAND =
```

Section 7
BETA CAE Systems Software Suite Installation and Startup: Hints & Tips

7.0. Introduction

This section describes how to address a software installation problem. Also, some technical information on the internals of the installation process is provided.

7.1. Windows

On Windows, the BETA CAE Systems Software Suite is distributed in a Microsoft Installer (.msi) format. It can be handled as any other .msi file you have come across.

7.1.1. Installation failed

Some common reasons that cause the installation to fail are:

- insufficient disk space or
- insufficient user privileges to install the software or
- incomplete download / error during download

To confirm that the download was correct, calculate the MD5 checksum of the installer and compare it with the .md5sum file found in the original download location. To calculate the MD5 checksum in a windows computer, you may use the built-in certUtil command line (notice the MD5 at the end):

```
certutil -hashfile path\to\file.msi MD5
```

Example:

```
certutil -hashfile C:\Downloads\BETA_CAE_Systems_v19.0.0_win_x64.msi MD5
```

If you have checked all of these cases (and possibly some variations of them) then please provide us with specific information on the machine you are running the installer on, in order to troubleshoot the problem:

1. What is the OS version?
2. Which user are you using to install the software? If not the Administrator, does the user have administrative privileges?
3. Are you doing a full install or have you selected specific components?
4. Did older Beta CAE Systems software versions install properly in the same environment?

Try running the installation, from a command prompt, with the following command:

```
msiexec /i BETA_CAE_Systems_v17.1.0_win_x64.msi /Log bcs_v17.1.0_install_log.txt
```

This will trace the install process and hopefully log the error in the text file. Include this log file in your report.

7.1.2. Installer not responding

If you find yourself believing that the installer is not responding, then please be patient and wait for it. Keep in mind that its size is over the gigabyte.

7.1.3. In need of a System Restart

We generally don't recommend or encourage system restarts. However, in many situations, a restart solves problems that are hard to report or impossible to reproduce. At your discretion, you might consider doing a System Restart, before going through the steps of reporting a problem.

7.1.4. Installation options missing / Administrator options not available

The installer, is compatible to several Windows OS i.e. Vista, XP, 7, 8.1,10 and behaves in a different way depending on the user's privileges that runs it.

This is in accordance with the Windows User Access Policy and the behavior may be different depending on the version of Windows being run.

Under Windows 7, a normal user may only install the application in its home folder:
%HOMEPATH%\AppData\Local\Apps\BETA_CAE_Systems\

An administrator may install the software for all users in the machine. For a user to elevate his/her permissions to the administrator level, Windows offer several options. Some of them are:

- Login as administrator
- Right click on the .msi file and select 'Run as administrator'
- Open a Command prompt window (cmd) in administrator mode and invoke the installer

Under Windows 10, if a user would like to install BETA CAE Systems Software Suite for all users, the following steps need to be performed:

- Open Command Prompt (CMD) with the option 'Run As Administrator'. This is required even if the user has administrative provileges.
- Navigate in the CMD to reach the folder, where the .msi file is located.
- Execute the .msi file.
- The *Advanced* button is available with both options included.

Note: If the above process is executed from Windows Explorer, regardless of the account type, the option for the installation for all users will not be available.

7.1.5. Administrator account may be disabled

Under Windows 7 or Windows 10, depending on the way your account is set up, Administrator privileges may not be enough and you may need to enable the Administrator account. To do this, drill down to:

Control Panel > Administrative Tools > Computer Management > Local Users and Groups

Right click on the Administrator > Properties and "uncheck" the option "*Account is disabled*".

7.1.6. Installation on a network shared folder / Share an installation

It is common for people to have valid reasons to share their installation on a network shared drive. Also, they may want to share it on another machine without going through the installation process. This possibility is supported by the installer.

To do this, you may install on one machine and then copy the BETA_CAE_Systems folder to your target location.

When avoiding running the installer, you will be missing some features. There will be no uninstall option; you will need to delete the folders by yourself. The .ansa and .metadb extensions will not be registered on your system and won't be opened by double clicking them. The META viewer plugin won't be installed as a system component.

7.2. Linux

On Linux, the BETA CAE Systems Software Suite is distributed as a self-extracting .tar.gz archive. This is down by having the .tar.gz wrapped inside a bash script.

Therefore, in order to run the installer, you need to either:

```
chmod +x BETA_CAE_Systems_v17.1.0_linux_x64.sh ./BETA_CAE_Systems_v17.1.0_linux_x64.sh
```

or the equivalent:

```
sh BETA_CAE_Systems_v17.1.0_linux_x64.sh
```

Also, running the command:
```
sh BETA_CAE_Systems_v17.1.0_linux_x64.sh --help
```

will give you a set of installer options you might find useful.

7.2.1. Installation failed

The Linux installer is basically a scripted untar process. Typical things that might go wrong are:

- the sh command does not correspond to the Bash shell
- insufficient disk space
- insufficient permissions to complete the task
- incomplete download / error during download

To confirm that the download was correct, calculate the MD5 checksum of the installer and compare it with the .md5sum file found in the original download location. To calculate the MD5 you may use the md5sum utility:

```
md5sum BETA_CAE_Systems_v19.0.0_linux_x64.sh
```

If you have checked all of these cases (and possibly some variations of them) then please provide us with information on the machine you are running the installer.

1. What is the version of the OS?
2. Which user are you using to install the software?
3. Does the script provide you any error messages? If yes, include these in your report.
4. What is the output of:
    a. echo $SHELL
    b. /bin/sh –version
5. Did older Beta CAE Systems Software versions install properly in the same environment?

7.2.2. Installer not responding

If you find yourself believing that the installer is not responding, then please be patient and wait for it. Keep in mind that its size is over the gigabyte.

7.2.3. Installation on a network shared folder / Share an installation

It is common for people to have valid reasons to share their installation on a network shared drive. Also, they may want to share it on another machine without going through the installation process. This possibility is supported by the installer.

To do this, you may install on one machine and then copy the BETA_CAE_Systems folder to your target location.

When avoiding running the installer, you will be missing the META Viewer plugin as an installed user

component.

BETA CAE Systems International AG
D4 Business Village Luzern, Platz 4
CH-6039 Root D4, Switzerland
T +41 41 545 3650, F +41 41 545 3651
ansa@beta-cae.com
www.beta-cae.com

physics on screen