

Report problems and error messages to CODESYS



Check the FAQ's and the FORGE page before reporting bugs.

Please check if the occurring errors have already been covered in the **FAQs** or if they have already been recorded and answered under our **Forge community**.

In addition, it can be helpful to inform you about the use of function blocks, methods and add-ons (provided by Codesys) in the Codesys Online Help (OLH).

Here, the usage, configuration, description and possible applications as well as general problems, that users might expect to encounter, are described.

Any error reports to CODESYS should contain the following information:

- The **versions** used within the Project (IDE, PLC Runtime, Add-on Versions, SoftMotion or Visualization Profile, etc.),
- A **clear error description** of what exactly occurs and when,
- The **exact 'Steps to repeat'** (STR) to reproduce the error (see also, why accurate 'steps for reproducing a problem' are important for any Technical Support),
- If possible, a **simplified/reduced example project** (as a full project archive) with the necessary scope/extent to be able to reproduce the error in it.

Besides the above information, the principle of "more is better" applies here.

In order to be able to answer inquiries efficiently, it is useful to provide all relevant data:

- [All used version information](#)
- [How to reproduce the problem](#)
- [Additional related specifics](#)
- [Reporting a runtime exception](#)
- [Reporting an exception in the Codesys Visualization](#)



See also the section for [which additional data is helpful and why](#):

The more detailed information the support team receives, the faster the problem can be isolated, and a resolution provided.

Which additional data is helpful and why?

To be able to detect problems which may only be related with the used version(s), or to be able to exclude known issues, the following version information should be provided:

- Version of the CODESYS programming environment (IDE).
- Version of the operating system, the used runtime and processor of the target system.
- Version of the used CODESYS Packages (like SoftMotion, Ethernet, Visualization, etc.).
- Version of the used Visualization Profile for visualization problems within the Visualization Editor and Visu Display Variants.

We always need accurate information and data to recreate the error profile:

- Exact 'Steps to repeat' to be able to reproduce the error.
The more precisely the individual steps are described, the less need there is to ask for error reproductions and the less room there is for interpretation when retesting.
- A full project archive (not only the CODESYS project!).
This contains the libraries used, the application structure of used functions, etc.
- For more complex processes, in addition to the written steps, a short screen recorded video, showing the steps and then the error that occurs, can be very useful.
- Note that sporadic errors are very difficult to solve.
In these cases, you should endeavor to determine the conditions under which it occurs, so that the support team has a chance to reproduce it and investigate.

See also [Why accurate 'steps for reproducing a problem' are important for any Technical Support](#)

Depending on the existing problem, different additions to be reported are necessary (or at least helpful in order to correctly understand and comprehend error profiles):

- If available, controller logs, with the appropriate log filter settings, should always be given.

- Screenshots of the error when it occurs in the Codesys IDE.
- In case of fieldbus related errors (or incorrectly displayed values in the different editors of the fieldbus), recording of the communication via 'Wireshark', that ranges from the download/login/start of the controller to after the occurrence of the error behavior, are needed.

Reporting a runtime exception

❗ By any exception in the runtime, first check if it is caused by an application structure in a POU, FB or library. To do this, use the **Implicit Checks** available in the Codesys IDE.

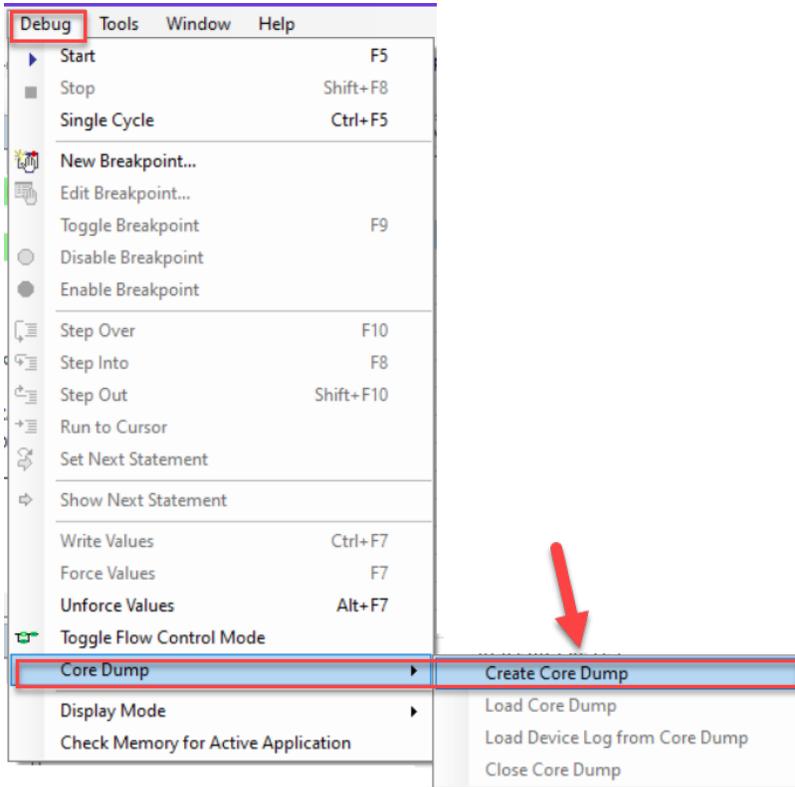
Exception Example:

Severity	Time Stamp	Description	Component
Warning		"SOURCEPOSITION" App=[Application] area=3, offset=215884	CmpIecTask
Error		"EXCEPTION" [AccessViolation] occurred: App=[Application], Task=[EtherCAT_Task]	CmpIecTask
Error		"EXCEPTION" [IoUpdateError] in application [Application]	CmpApp

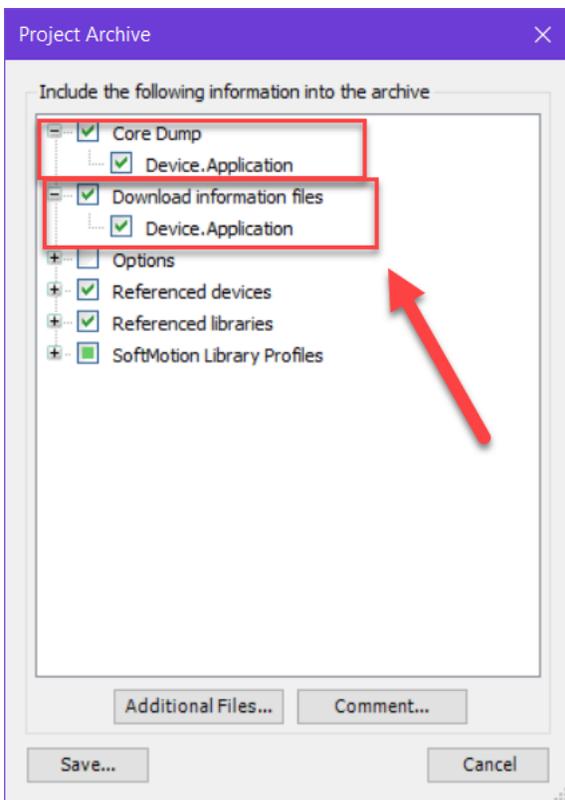
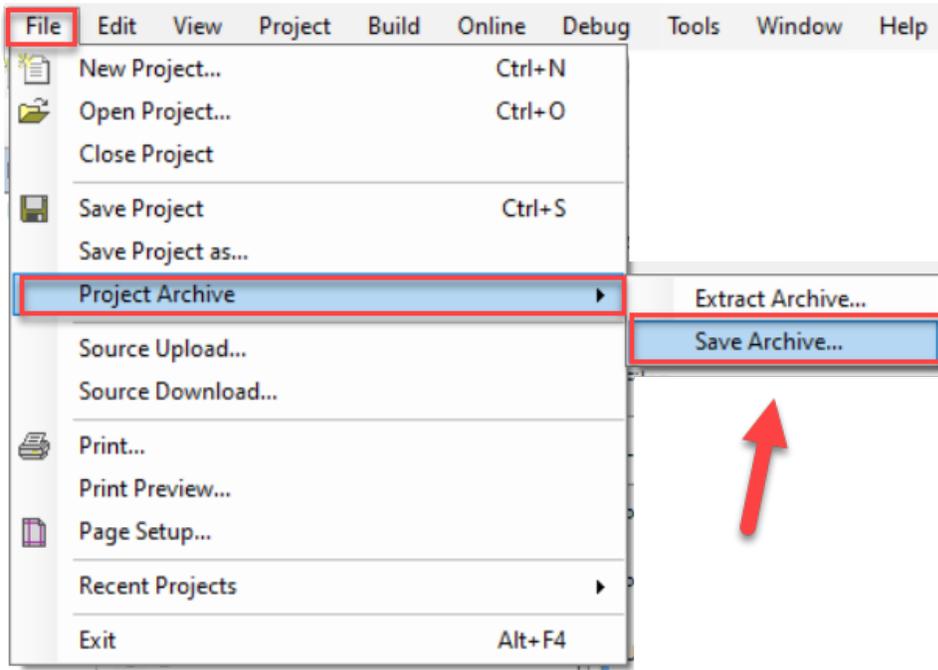
i See also our OLH on how to 'Create a Core Dump', and how to 'Creating a Core Dump of the running application manually'.

Please provide:

- The CoreDump, created at the time the exception occurred.



- Send a full project archive, with the associated *Callstack* and a Core Dump, created at the time of the crash.



If you need technical support, please purchase a **Support Ticket** in the CODESYS Store. Note that certain products in the CODESYS Store already include a Support Ticket.

To report **software bugs** or if you have a question concerning the products in the CODESYS Store, click on the **"My Question"** button next to the shopping cart in the **CODESYS Store**.

FAQ:

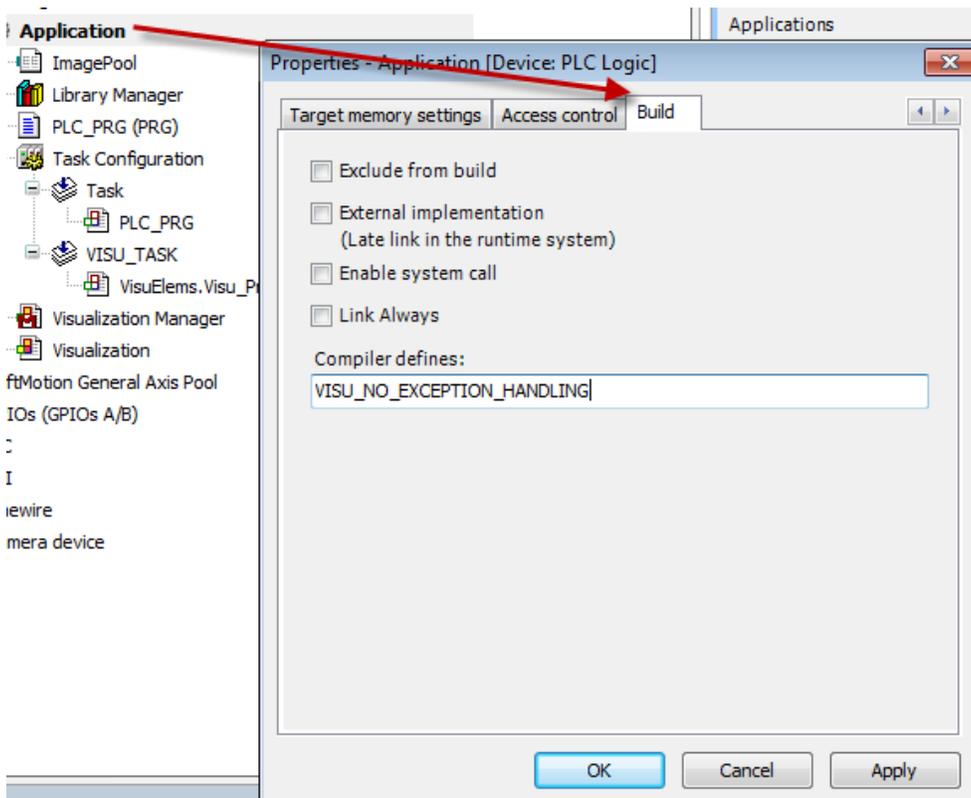
i For vendor (or 'third party') specific errors, it must distinguish where the error occurs:

- **The errors must be reproducible under the Codesys IDE and with a CODESYS SL PLC.**
 - For errors that occur only under other IDEs, which were modified via brand labelling, the corresponding manufacturer must be contacted.
 - For errors that occur only with a device manufacturer's PLC, the device manufacturer should be contacted to pursue a resolution for the problem.

This is due to the fact that other IDE manufacturers use Codesys as a platform, but can build their own restrictions or extensions that deviate from the default interaction, behavior and standards of the Codesys IDE, and therefore can not be pursued by us.

- **Errors that occur with a library, not developed by CODESYS GmbH**
 - For errors that occur with a library, which was written by a 'third party', this library manufacturer should be contacted for further support.

- Please set the *compiler-define*: **VISU_NO_EXCEPTION_HANDLING**



- Send a [full project archive](#), with the associated *Callstack* and a *Core Dump*, created at the time of the crash.

More detailed information can be found under [debugging a Visu Exception](#) FAQ.

See also....

- [CODESYS Online Help \(OLH\) Website](#)
- OLH: To set or find the used 'Compiler Version' within a Codesys Project
- OLH: To set or find the used 'Visualization Profile' within a Codesys Project
- OLH: The use-case and possibilities of implicit monitoring functions - 'Implicit Checks'
- OLH: How to 'Analyzing Errors with Core Dump' and how to 'Create a Core Dump' in the first place
- Working with an [Incomplete CoreDump \(core file incomplete\)](#)
- OLH: the PLC Log and how to Reading the PLC Log
- OLH: Visualization: a system overview, mechanism and display variants
- How to '[Debugging a Visu Exception](#)'
- OLH: Transferring Projects
- [Why accurate 'steps for reproducing a problem' are important for any Technical Support](#)
- [How to: Analyze a BSOD/Crash of the application](#)

