

# Report problems and error messages to CODESYS

Any error reports to CODESYS should contain the following information:

- The **versions** used within the Project,
- A **clear error description** of what exactly occurs and when,
- The **exact 'Steps to repeat'** (STR) to reproduce the error,
- 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

-> see also [which additional data is helpful and why](#):

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

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

## Whom to contact for vendor-specific errors?

- **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 'third party' should be contacted for support.

## 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](#).

[^GotoTop^](#)

## Which additional data is helpful and why?

### 1. All used version information

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](#).

### 2. How to reproduce the problem

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.

### 3. Additional related specifics

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.

[^GotoTop^](#)

## For reporting a runtime exception

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

For any exception in the runtime, we need basic a complete repository/project archive and a Coredump, in which these exceptions have been marked with to be able to debug the problem:

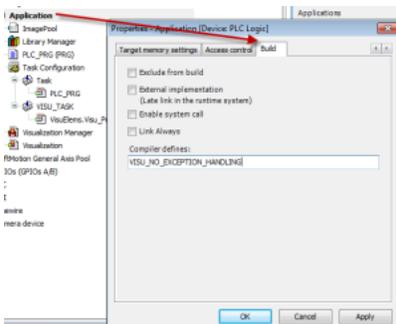
- send a full project archive, with with associated Callstack and a **Coredump**:



- When an Exception error occurs from one of the CODESYS visualization elements, please set the *compiler-define*:

**VISU\_NO\_EXCEPTION\_HANDLING**

and send a full project archive (with an Callstack and a Coredump):



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

**!** 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**.

**See also....**

- Our [Codesys Online Help \(OLH\) Website](#)
- Basic information to the [Codesys Development System \(Codesys Homepage\)](#)
- OLH: The [Online-help introduction for the Codesys Development System \(IDE\)](#)
- [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
- [OLH](#): [Visualization: a system overview, mechanism and display variants](#)
- How to [Debugging a Visu Exception](#)