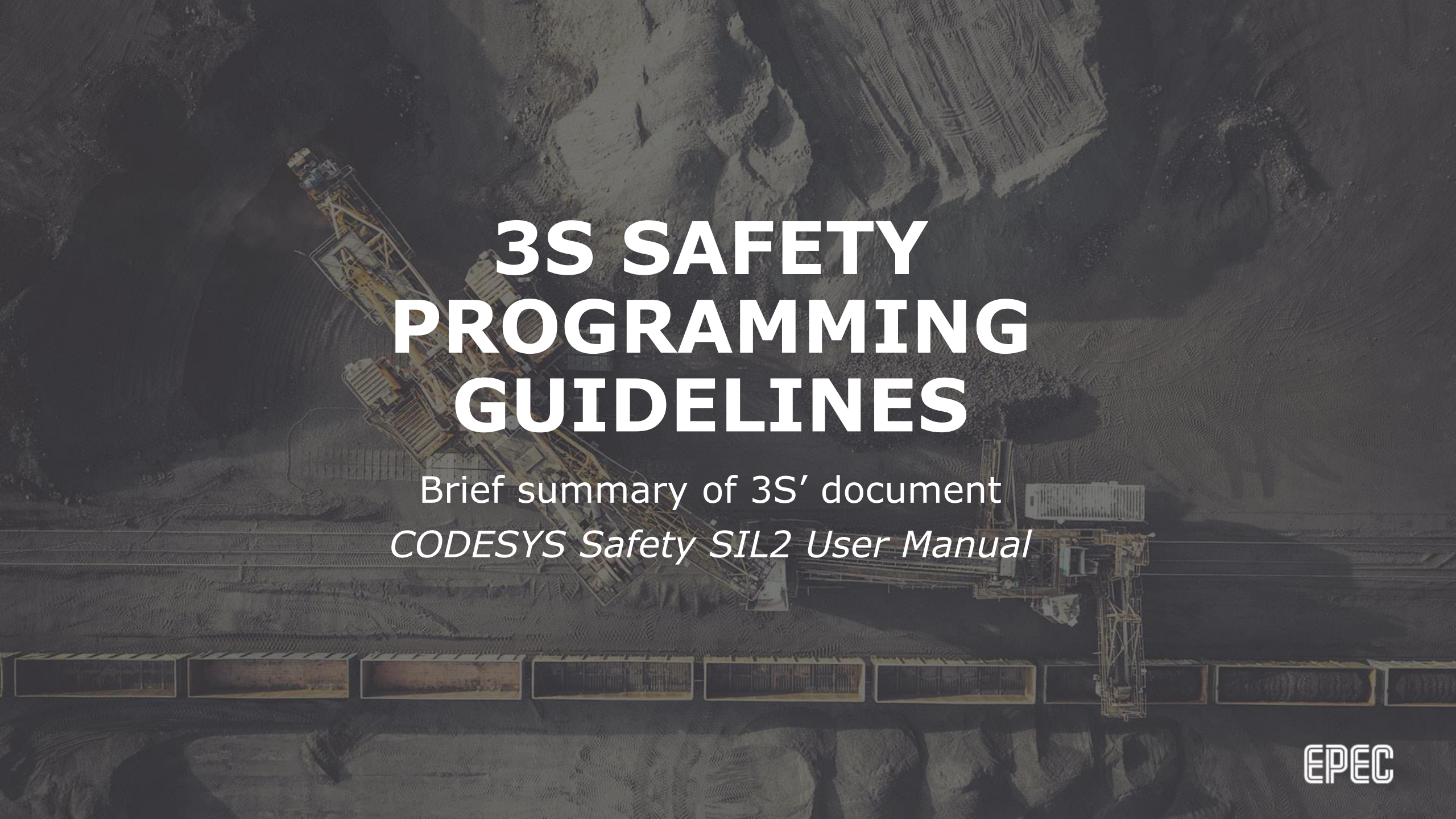


3S SAFETY PROGRAMMING GUIDELINES

Brief summary of 3S' document

CODESYS Safety SIL2 User Manual

EPEC

INTRODUCTION

- *CODESYS safety SIL2 User Manual* document contains
  - safety requirements that need to be observed when programming CODESYS Safety SIL2 controllers
  - programming recommendations that can simplify the certification of a failsafe controller
- It is assumed that the machine is certified according to EN13849 and/or EN62061
- Adherence to the safety requirements listed in *CODESYS safety SIL2 User Manual's Appendix 7.4* is absolutely necessary

STANDARDS

- EN13849
  - harmonized standard
  - regards the overall system, and refers to IEC61508 or EN62061 for the certification of the software
- EN62061
  - refers (in chapter 6.11.3.1.1) to IEC61508 for the use of a programming language with unlimited language scope
- Hence, EN62061 or IEC61508 are applied, depending on the complexity of the software

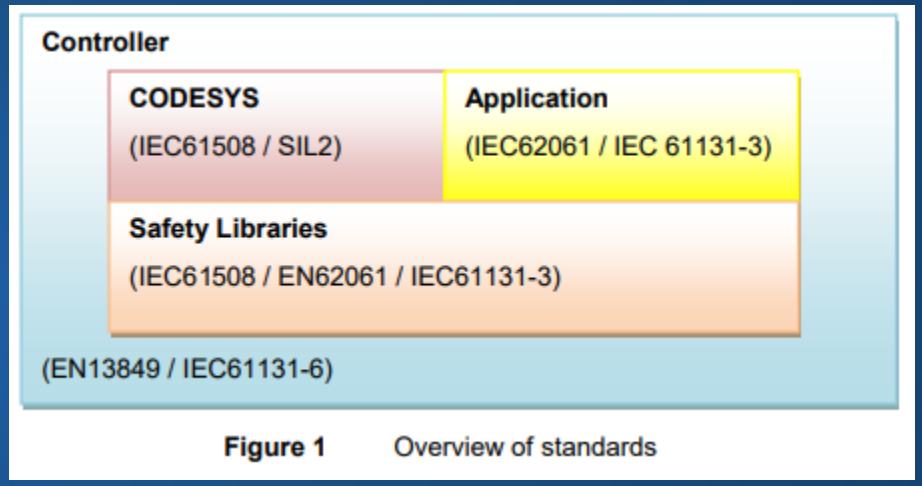


Figure 1 Overview of standards

Image source, *CODESYS safety SIL2 User Manual, page 9*

CORRESPONDENCE OF PL AND SIL

Performance Level, PL (EN13849)	Safety Integrity Level, SIL (EN62061)
a	-
b	1
c	1
d	2
e	3

TWO EXECUTION MODES

- CODESYS Safety SIL2 programmable controller has two execution modes
  1. Safety Mode
     - Controllers usually start up to this mode
     - The execution of a fail-safe application is permitted
  2. Debug Mode
     - Controller needs to be changed into debug mode to, for example, download or debug the application
     - The safe operation of a fail-safe application is not guaranteed

TWO PROGRAMMING LANGUAGE CLASSES

- EN13849 and EN62061 standards define two classes of programming languages
  - *LVL, Limited Variability Language*
    - graphical languages of the IEC61131-3, such as FBD or LD (ST with limited scope)
    - If the application itself uses exclusively LVLs, then a functional test is sufficient for them
  - *FVL, Full Variability Language*
    - Textual languages, such as ST, C, IL and Assembler
    - If the application uses FVLs, these application parts must be tested completely by means of module tests in accordance with IEC61508

THREE LEVELS OF FAIL-SAFE APPLICATION

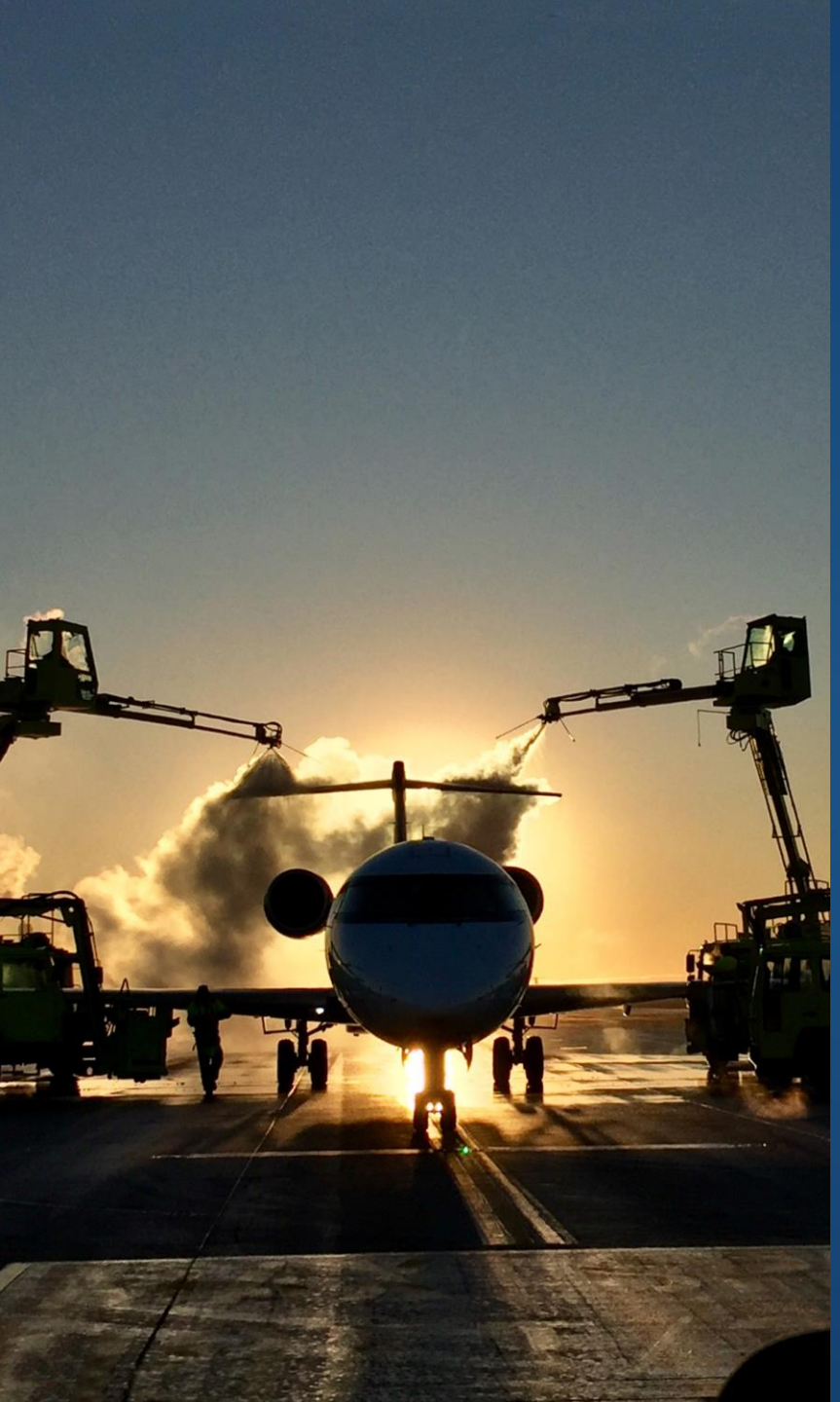
- Fail-safe applications can be subdivided into three levels
  1. Basic
  2. Extended
  3. System

- The programming languages for application development at basic and extended level are limited to FBD and LD (LVL)
- ST can also be used if the language scope is limited

1. BASIC LEVEL

- Programming is limited to LVL (FBD, Ladder, (ST*))
  - Certification can take place in accordance with **EN62061** without reference to IEC61508
- A fail-safe application consisting only of pre-certified boxes that are connected to one another in a simple manner
  - Limited in linking options and in the method of use of individual boxes in such a way that they remain easily readable and are thus less susceptible to error
- Testing
  - Application is tested in the integration test later on during the acceptance of the machine
  - No module tests needed in the case of moderate complexity

*ST can also be used if the language scope is limited appropriately



2. EXTENDED LEVEL

- Programming languages are still limited to LVL
- More links are permitted at *extended level* than at *basic level*
- Extended level application parts are usually distributed as libraries
- Certification can likewise take place in accordance with **EN62061** without reference to IEC61508
  - the effort required for certification is higher than at *basic level* due to the greater complexity of the software

EPEC

3. SYSTEM LEVEL

- No restrictions for programming languages
  - Due to the use of FVLs, EN62061 refers in chapter 6.11.3.1.1 to IEC61508 for the certification
- The developer is only restricted in his choice of programming language by the required suitability in accordance with IEC61508

EPEC

VALIDATION AND VERIFICATION

- The exact validation and verification measures demanded for the individual level depend on the safety guidelines that have to be fulfilled
- For the development of a fail-safe application with CODESYS Safety SIL2, however, at least code reviews are required in order to satisfy the programming guidelines and safety requirements described in *CODESYS safety SIL2 User Manual*

VALIDATION AND VERIFICATION

- Required, for example,
  1. Source code reviews are required for all software parts that have to be certified
     - For CODESYS code template review instructions, see *Epec Programming and Libraries Manual > Programming > Programming Safety Projects > Code Template Review Instructions*
  2. Fail-safe code must be completely free of warnings
  3. Libraries must have the "SIL2" property of the type BOOL set
     - Safety related library, SIL2 property is TRUE
     - Non-safety related library, SIL2 property is FALSE

PROGRAMMING GUIDELINES

- *CODESYS safety SIL2 User Manual's chapter 5* contains the programming guidelines for all IEC61131-3 languages that may be used under CODESYS Safety SIL2 for safe applications
- The rules are recommendations about, for example,
  - Naming and calling conventions
  - Variable declarations
  - Data types
  - Tasks

EPEC DOCUMENTATION

- **Epec Extranet, *Programming Manuals***
  - CODESYS Safety SIL2 User manual (PDF)
  - SC52 Safety Manual (PDF)
  - Programming and Libraries Manual (HTML, CHM)
- ***Epec Programming And Libraries Manual***
  - SDK installs to *C:\Program Files (x86)\Epec\SDKDocumentation*
  - Open via **MultiTool > Help**
  - *Programming book*
    - *Programming Safety Projects*
    - *Programming SC52 Safety Control Unit*
  - *Libraries book*
    - *S Series Specific Libraries*
    - *Common Libraries for Safety Project*

Thank you!

Any questions?

Contact our technical support
techsupport@epec.fi