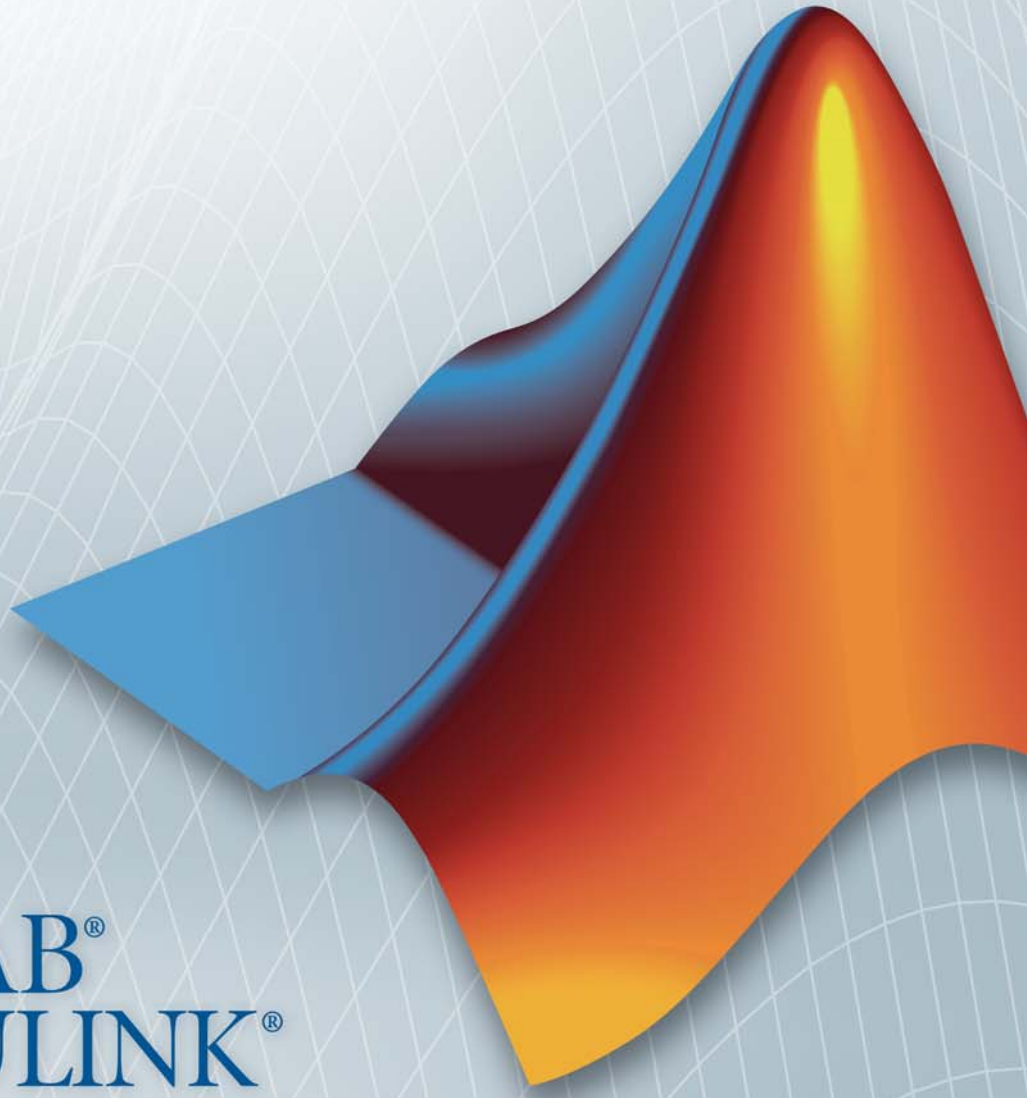
IEC Certification Kit

User's Guide

R2011b



MATLAB®
&SIMULINK®

MathWorks®

How to Contact MathWorks

`www.mathworks.com`	Web
`comp.soft-sys.matlab`	Newsgroup
`www.mathworks.com/contact_TS.html`	Technical Support

`suggest@mathworks.com`	Product enhancement suggestions
`bugs@mathworks.com`	Bug reports
`doc@mathworks.com`	Documentation error reports
`service@mathworks.com`	Order status, license renewals, passcodes
`info@mathworks.com`	Sales, pricing, and general information

508-647-7000 (Phone)

508-647-7001 (Fax)

The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

IEC Certification Kit User's Guide

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See `www.mathworks.com/trademarks` for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see `www.mathworks.com/patents` for more information.

Revision History

March 2009	Online only	New for Version 1.0 (Applies to Releases 2007a+, 2008a, 2008b, 2009a)
September 2009	Online only	Revised for Version 1.1 (Applies to Releases 2008a, 2008b, 2009a, 2009a+, 2009b)
March 2010	Online only	Revised for Version 1.2 (Applies to Release 2010a)
September 2010	Online only	Revised for Version 1.3 (Applies to Releases 2009bSP1, R2010a, 2010b)
April 2011	Online only	Revised for Version 1.4 (Applies to Releases 2010bSP1, 2011a)
September 2011	Online only	Revised for Version 2.0 (Applies to Release 2011b)

Contents

1 Getting Started

2 Certification Process

3 Validating Software Tools

4 Accessing and Managing Certification Artifacts

5 Supporting Certification-Related Development Activities

6 Function Reference

7 Functions — Alphabetical List

1

Getting Started

IEC Certification Kit Product Overview

In this section...

What Is the IEC Certification Kit Product?

IEC Certification Kit provides tool-qualification artifacts, certificates, and test suites, and generates traceability matrices. The kit helps you qualify MathWorks® code generation and verification products and streamline certification of your embedded systems to ISO 26262, IEC 61508, and related functional-safety standards. Certificates and assessment reports from the certification authority TÜV SÜD support Embedded Coder™, Polyspace® products, Simulink® Design Verifier™, Simulink® PLC Coder™, and Simulink® Verification and Validation™. Supported safety standards include ISO 26262, IEC 61508, EN 50128, and IEC 61511.

IEC Certification Kit provides ISO 26262 tool classification and qualification work products, together with test suites. It includes templates that let you adapt the work products to meet specific project needs. You can also generate project-specific artifacts, including traceability matrices covering requirements, models, and generated code. You can combine the project- and product-specific artifacts to produce a complete ISO 26262 tool qualification package for embedded system certification.

Note Neither compliance with nor certification to the applicable safety standard ensure the safety of the software or the system under consideration. However, the applicable safety standard may be considered a state-of-the-art or generally accepted rules of technology (GART) for the development of safety-related systems in your industry. A certification might be used as evidence that state-of-the-art procedures were applied during system development.

To view the certification artifacts that are part of the IEC Certification Kit product, use the Certification Artifacts Explorer. For more information, see "Accessing Certification Artifacts Using the Certification Artifacts Explorer" on page 4-2.

For more information on how to leverage the IEC Certification Kit product, see Chapter 2, "Certification Process".

ISO 26262

What Is ISO 26262?

ISO 26262 is an emerging international functional safety standard titled *Road vehicles — Functional safety*. ISO 26262 is the adaptation of IEC 61508 to comply with needs specific to the application sector of *E/E systems*[1] within road vehicles.

ISO® developed the ISO/FDIS 26262 final draft international standard in 2011. It consists of ten parts, referred to as ISO/FDIS 26262-1 to ISO/FDIS 26262-10.

1. Systems that consists of electrical and electronic elements, including: programmable electronic elements, power supplies, input devices, communication paths, and output devices.

Part 2 (ISO/FDIS 26262-2 *Management of functional safety* specifies the requirements on functional safety management for automotive applications. Part 6 (ISO/FDIS 26262-6) *Product development: software level* pertains to software development, verification, and validation. It includes guidance for projects using Model-Based Design[2] and code generation. Part 8 (ISO/FDIS 26262-8) *Supporting processes* addresses multiple cross-functional topics, including the classification and qualification of software tools.

The required degree of rigor for software development, verification, and validation varies, depending on how critical the software is. It is expressed in terms of Automotive Safety Integrity Levels (ASILs) A to D. For example, a measure or technique listed in ISO 26262 might be recommended for ASIL A and ASIL B, and highly recommended for ASIL C and ASIL D.

ISO 26262 Compliance Considerations

ISO/FDIS 26262-2 lays out confirmation measures to be carried out in order to claim compliance with the standard.

ISO 26262 Tool Qualification Considerations

ISO/FDIS 26262-8 provides a framework for software tool classification and qualification to provide evidence that a software tool is suitable for use when developing safety-related software. In this way, confidence can be achieved in the correct execution of the activities and tasks supported by this tool (see ISO/FDIS 26262-8, clause 11).

To determine the required level of confidence in a software tool (tool confidence level, TCL), the applicant shall analyze the use cases for the software tool. The analysis determines:

- If a malfunctioning software tool and the erroneous output of the tool can lead to the violation of a safety requirement.
- The probability of preventing or detecting such errors in the output.

The evaluation considers tool-internal measures (for example, monitoring), as well as tool-external measures (for example, guidelines, tests, reviews) that

2. Referred to as *model-based development*.

the applicant implements in the development process for the safety-related software.

The required TCL, together with the ASIL of the software developed using the tool, determines whether tool qualification is needed and allows the selection of the appropriate qualification methods.

Regardless of the tool qualification, the tool user is and remains fully responsible for the safety of the system and its embedded software.

IEC 61508

What Is IEC 61508?

IEC 61508 is an international, industry-independent functional safety standard, titled *Functional safety of electrical/electronic/programmable electronic safety-related systems*. The seven parts of the standard (referred to as IEC 61508-1 to IEC 61508-7) were published in 2010.

IEC 61508-3 *Software Requirements* concerns software development, verification, and validation. By constraining the processes used for software development and quality assurance, the intention of the IEC 61508-3 standard is to:

- Reduce the number of errors introduced during software development.
- Increase the number of errors revealed by verification and validation activities.

IEC 61508 is a prescriptive standard, providing detailed lists of techniques and measures with recommendations. The required degree of rigor for software development, verification, and validation varies, depending on how critical the software is. The standard expresses the degree of rigor in terms of Safety Integrity Levels (SILs). For example, IEC-61508-3 might recommend

a measure or technique for SIL 1 and 2, and highly recommend it for SIL 3 and 4.

To help with the selection of techniques and measures appropriate for a required SIL, annexes A and B of IEC 61508-3 provide software safety integrity tables. The tables list the techniques and measures recommended for each SIL. The standard organizes the tables based on the different software lifecycle phases. IEC 61508-7 *Overview of techniques and measures* provides detailed descriptions of selected measures and techniques.

IEC 61508 Compliance Considerations

IEC 61508 certification confirms that a product or system complies with objectives set by the standard.

You can get IEC 61508 compliance certified by an independent, external certification authority, such as Technischer Überwachungsverein (TÜV) in Germany. Upon granting certification, the certification authority issues a certificate and, if applicable, a certificate report. A certificate report is a technical report that accompanies the certificate. The certificate report documents details of the certification process and constraints for the certificate.

An applicant might self-certify a system. Self-certification requires the applicant to demonstrate IEC 61508 compliance to an internal assessor, without requiring external certification. In this case, aspects of the standard might be relaxed or tightened.

Regardless of how an applicant achieves certification, the applicant shall document compliance with the relevant set of IEC 61508 requirements. For software, the applicant typically creates customized instances of software safety integrity tables. The tables describe how you interpreted and applied each recommended technique and measure for the software under development. If a highly recommended technique or measure is not used, the rationale shall be documented and agreed upon with the certification authority or internal assessor.

The customized software safety integrity tables serve as partial evidence to demonstrate that the objectives of the standard are met. To facilitate certification, the applicant should submit an initial version of the tables early

in the software development lifecycle to the certification authority or internal assessor for discussion and approval.

IEC 61508 Tool Certification Considerations

The intention of the IEC 61508 standard is to regulate the development of safety-related systems, not the development of software tools used to design, verify, and validate these systems. However, IEC 61508 includes some requirements on the usage of software tools. In particular, IEC 61508-3, clause 7.4.4 provides requirements for tools used to develop safety-related software, including a tool classification scheme and requirements for tool validation.

IEC 61508-3, table A.3 highly recommends certified tools and translators for safety integrity levels SIL 2 and higher.

Different tool certification approaches have been proposed and pursued in practice. A recent approach is in-context certification of tools. In-context certification is based on a specific workflow or set of workflows to be used when applying the tool to develop or verify software for IEC 61508 compliant or certified applications. For an in-context certification, the certification package includes a reference workflow document in addition to a certificate and certificate report. The applicant shall ensure that the tool is used within the workflows referenced and the constraints specified in their respective certificates.

Regardless of the tool certification, the tool user is and remains fully responsible for the safety of the system and its embedded software.

EN 50128

What Is EN 50128?

EN 50128 is a European safety standard titled *Railway applications - Communications, signalling and processing systems - Software for railway control and protection systems*. The standard specifies procedures and technical requirements for the development of programmable electronic systems for use in railway control and protection applications. EN 50128, developed by the European Committee for Electrotechnical Standardization

(CENELEC), is part of a series of standards that represent the railway application-specific interpretation of the IEC 61508 standard series.

IEC 61511

What Is IEC 61511?

IEC 61511 is an international functional safety standard titled *Functional safety - Safety Instrumented Systems for the process industry sector*. IEC 61511 has been developed as a process sector implementation of IEC 61508. The standard consists of three parts, referred to as IEC 61511-1 to IEC 61511-3. Part 1 (IEC 61511-1) covers framework, definitions, and system, hardware, and software requirements.

IEC Certification Kit Components

- "Certification Artifacts for the Embedded Coder Product" on page 1-10
- "Certification Artifacts for the Simulink® PLC Coder Product" on page 1-11
- "Certification Artifacts for the Simulink® Design Verifier Product" on page 1-12
- "Certification Artifacts for the Simulink® Verification and Validation Product" on page 1-13
- "Certification Artifacts for the Polyspace® Client for C/C++ and Polyspace® Server for C/C++ Products" on page 1-14

The IEC Certification Kit product includes the following certification artifacts and tools:

- Certification and qualification evidence
- Documents and templates
- Tools for certification-related development activities
- Tools for managing certification artifacts
- Test cases and test procedures to support tool validation

The certification artifacts and tools support you when using the following MathWorks products in the context of the ISO 26262, IEC 61508, EN 50128, or IEC 61511 standards:

- Embedded Coder
- Simulink PLC Coder
- Simulink Design Verifier
- Simulink Verification and Validation
- Polyspace® Client™ for C/C++; Polyspace® Server™ for C/C++

Specific versions of the preceding MathWorks products have been certified or prequalified by TÜV SÜD, a German-based certification authority, according to one or more of the above mentioned standards.

The IEC Certification Kit product contains certification artifacts to document compliance with the respective standards. The applicant can submit certification artifacts, or derivatives thereof, as evidence of compliance with ISO/FDIS 26262-6, ISO/FDIS 26262-8, IEC 61508-3, EN 50128, or IEC 61511-1.

The IEC Certification Kit product provides the following capability to support certification-related development activities:

Generating traceability matrices for tracing among model objects, generated code, and model requirements (see “Generating a Traceability Matrix” on page 5-2).

The IEC Certification Kit product provides a Certification Artifacts Explorer, a tool for accessing and managing certification artifacts (see Chapter 4, “Accessing and Managing Certification Artifacts”).

The IEC Certification Kit product provides test procedures that can be used to automate tool validation tests for Embedded Coder and Simulink Verification and Validation (see Chapter 3, “Validating Software Tools”).

Note The `rights.txt` file, located at *`matlabroot`*`/toolbox/qualkits/iec`, describes allowed uses of the IEC Certification Kit product.

Certification Artifacts for the Embedded Coder Product

TÜV SÜD has certified specific versions of the Embedded Coder product for use in development processes that are required to comply with ISO/FDIS 26262, IEC 61508, EN 50128, or derived standards. These product versions are also prequalified according to ISO/FDIS 26262-8 for Automotive Safety Integrity Levels ASIL A through ASIL D.

The IEC Certification Kit product contains certification artifacts for the following versions of the Embedded Coder product:

Version 6.1 (R2011b)

Previous releases of the Embedded Coder product are certified or prequalified. For supporting certification artifacts, see previous releases of the IEC Certification Kit product.

Note The Embedded Coder product was not developed using an IEC 61508 certified process.

Certification artifacts for the Embedded Coder product are in the following folder:

`matlabroot``/toolbox/qualkits/iec/ecoder/r2011b/`

Details on the certification artifacts are in the certificate reports.

Component	File
Certificate	`certkitiec_ecoder_certificate.pdf`
Certificate Report	`certkitiec_ecoder_certreport.pdf`
Reference Workflow Documentation	`certkitiec_ecoder_workflow.pdf`
Conformance Demonstration Template	`certkitiec_ecoder_cdt.rtf/.pdf`

Component	File
ISO 26262 Tool Qualification Package	`certkitiec_ecoder_tqp.rtf/.pdf`

Test Procedure / Test Cases	`certkitiec_ecoder_tests.m`
	`certkitiec_ecoder_modelList.m`
	`/tests/*`
	`/outputs/*`
	`/baseline/*`

Certification Artifacts for the Simulink PLC Coder Product

TÜV SÜD certified specific versions of the Simulink PLC Coder product for use in development processes that are required to comply with IEC 61508 or IEC 61511.

The IEC Certification Kit product contains certification artifacts for the following versions of the Simulink PLC Coder product:

Version 1.2.1 (R2011b)

Previous releases of the Simulink PLC Coder product are certified. For supporting certification artifacts, see previous releases of the IEC Certification Kit product.

Note The Simulink PLC Coder product was not developed using an IEC 61508 certified process.

Certification artifacts for the Simulink PLC Coder product are in the following folder:

matlabroot`/toolbox/qualkits/iec/plccoder/r2011b/`

Details on the certification artifacts are in the certificate reports.

Component	File
Certificate	`certkitiec_plccoder_certificate.pdf`
Certificate Report	`certkitiec_plccoder_certreport.pdf`

Component	File
Reference Workflow Documentation	`certkitiec_plccoder_workflow.pdf`
Conformance Demonstration Template	`certkitiec_plccoder_cdt.rtf/.pdf`

Certification Artifacts for the Simulink Design Verifier Product

TÜV SÜD has certified specific versions of the Simulink Design Verifier product for use in development processes that are required to comply with ISO/FDIS 26262, IEC 61508, EN 50128, or derived standards. These product versions are also prequalified according to ISO/FDIS 26262-8 for Automotive Safety Integrity Levels ASIL A through ASIL D.

The IEC Certification Kit product contains certification artifacts for the following versions of the Simulink Design Verifier product:

Version 2.1 (R2011b)

Note The Simulink Design Verifier product was not developed using an IEC 61508 certified process.

Certification artifacts for the Simulink Design Verifier product are in the following folder:

matlabroot`/toolbox/qualkits/iec/sldv/r2011b/`

Details on the certification artifacts are in the certificate reports.

Component	File
Certificate	`certkitiec_sldv_certificate.pdf`
Certificate Report	`certkitiec_sldv_certreport.pdf`
Reference Workflow Documentation	`certkitiec_sldv_workflow.pdf`
Conformance Demonstration Template	`certkitiec_sldv_cdt.rtf/.pdf`
ISO 26262 Tool Qualification Package	`certkitiec_sldv_tqp.rtf/.pdf`

Certification Artifacts for the Simulink Verification and Validation Product

TÜV SÜD has certified specific versions of the Simulink Verification and Validation product for use in development processes that are required to comply with ISO/FDIS 26262, IEC 61508, EN 50128, or derived standards. These product versions are also prequalified according to ISO/FDIS 26262-8 for Automotive Safety Integrity Levels ASIL A through ASIL D.

The IEC Certification Kit product contains certification artifacts for the following versions of the Simulink Verification and Validation product:

Version 3.2 (R2011b)

Note The Simulink Verification and Validation product was not developed using an IEC 61508 certified process.

Certification artifacts for the Simulink Verification and Validation product are in the following folder:

matlabroot/toolbox/qualkits/iec/slvnv/r2011b/

Details on the certification artifacts are in the certificate reports.

Component	File
Certificate	certkitiec_slvnv_certificate.pdf
Certificate Report	certkitiec_slvnv_certreport.pdf
Reference Workflow Documentation	certkitiec_slvnv_workflow.pdf
Conformance Demonstration Template	certkitiec_slvnv_cdt.rtf/.pdf
ISO 26262 Tool Qualification Package	certkitiec_slvnv_tqp.rtf/.pdf
Test Procedure / Test Cases	certkitiec_slvnv_tests*.rpt/.xls /tests/* /outputs/*

Certification Artifacts for the Polyspace Client for C/C++ and Polyspace Server for C/C++ Products

TÜV SÜD certified specific versions of the Polyspace Client for C/C++ and the Polyspace Server for C/C++ products for use in development processes that are required to comply with ISO/FDIS 26262, IEC 61508, EN 50128, or derived standards. These product versions are also prequalified according to ISO/FDIS 26262-8 for Automotive Safety Integrity Levels ASIL A through ASIL D.

The IEC Certification Kit product contains certification artifacts for the following versions of the Polyspace Client for C/C++ and the Polyspace Server for C/C++ products:

Version 8.2 (R2011b)

Previous releases of the Polyspace products are certified or prequalified. For supporting certification artifacts, see previous releases of the IEC Certification Kit product.

Note The Polyspace Client for C/C++ and the Polyspace Server for C/C++ products were not developed using an IEC 61508 certified process.

Certification artifacts for the Polyspace Client for C/C++ and Polyspace Server for C/C++ products are in the following folder:

matlabroot`/toolbox/qualkits/iec/polyspace/r2011b/`

Component	File
Certificate	`certkitiec_polyspace_certificate.pdf`
Certificate Report	`certkitiec_polyspace_certreport.pdf`
Reference Workflow Documentation	`certkitiec_polyspace_workflow.pdf`
Conformance Demonstration Template	`certkitiec_polyspace_cdt.rtf/.pdf`
ISO 26262 Tool Qualification Package	`certkitiec_polyspace_tqp.rtf/.pdf`

Required Knowledge

Before using the IEC Certification Kit product, make sure that you have:

- Knowledge about developing safety-related software.
- Knowledge of the applicable safety standard:
  - ISO 26262 *Road vehicles - Functional safety*
  - IEC 61508 *Functional safety of electrical/electronic/programmable electronic safety-related systems*
  - EN 50128 *Railway Applications - Communications, Signalling and Processing Systems - Software for Railway Control and Protection Systems*
  - IEC 61511 *Functional safety - Safety Instrumented Systems for the process industry sector*
- Experience with MathWorks products that you use to develop, verify, or validate software for systems that are required to comply with the applicable standard.

If you have an Embedded Coder license, also review the following information:

- "Developing Models and Code That Comply with the ISO 26262 Standard" in the Embedded Coder documentation
- "Developing Models and Code That Comply with the IEC 61508 Standard" in the Embedded Coder documentation

2

Certification Process

Certification Process Using the IEC Certification Kit Product

In this section...

Defining Certification Objectives and Requirements

Before using the IEC Certification Kit product, define your certification objectives and requirements.

- Identify the scope of your certification activities, such as the application to certify.
- Decide on the applicable safety standards and the required Safety Integrity Level (SIL) or Automotive Safety Integrity Level (ASIL).
- Determine the software development processes and software tool chain to use.
- Define tool certification or qualification requirements, including the tools and versions to certify or qualify.

Certifying or Qualifying Software Tools

The ISO 26262 and IEC 61508 standards include requirements or recommendations to use certified or qualified tools. You can use tool certification evidence from the IEC Certification Kit product to document compliance with the requirements or recommendations concerning tool certification or qualification.

Note Using certified or qualified tools does not ensure the safety of the application under development.

The IEC Certification Kit product provides tool certification and prequalification evidence for the following MathWorks products:

- Embedded Coder
- Simulink PLC Coder
- Simulink Design Verifier
- Simulink Verification and Validation
- Polyspace Client for C/C++; Polyspace Server for C/C++

The IEC Certification Kit product follows an in-context approach to tool certification and qualification. This approach is based on specific workflows to be used when applying the certified and qualified tools to develop or verify software for ISO 26262 and IEC 61508 applications. The applicant must ensure that the tools are used within the referenced workflows and constraints specified in the certificates.

ISO 26262 Tool Qualification Artifacts

The IEC Certification Kit product provides support for creating ISO 26262 tool qualification artifacts for the following products:

- Embedded Coder
- Simulink Design Verifier
- Simulink Verification and Validation
- Polyspace Client for C/C++; Polyspace Server for C/C++

For details, see the ISO 26262 Tool Qualification Package documents for these products.

Note Some safety standards, including IEC 61508, do not have a formal concept of certification credits. The amount of credit for the use of certified or qualified tools is dependent on the applicant's development, verification and validation processes, and how the applicant uses the tools within those processes. The applicant should propose and discuss an initial version of the compliance package, including tool qualification data, to the certification authority or internal assessor early in the development lifecycle.

IEC 61508 Tool Certification Artifacts

The IEC Certification Kit product provides support for creating the following artifacts related to tool certification according to IEC 61508.

Products	Purpose	References	Artifacts and Documents[1]
Embedded Coder	Tool certification evidence for code generator	• IEC 61508-3 Clause 7.4.4 • IEC 61508-3 Table A-3 (4a) "Certified tools and certified translators"	• Certificate Z10 11 06 67052 010 • Certificate report MN72051C
	Documentation of reference workflow	N/A	*Application-Specific Verification and Validation of Models and Generated C and C++ Code*
	Evidence for using the code generator within the referenced workflows and within the constraints specified in its certificate	N/A	Customized and completed *Conformance Demonstration Template*

Products	Purpose	References	Artifacts and Documents[1]
Simulink PLC Coder	Tool certification evidence for code generator	• IEC 61508-3 Clause 7.4.4 • IEC 61508-3 Table A-3 (4a) "Certified tools and certified translators"	• Certificate Z10 11 01 67052 007 • Certificate report MN76171C
	Documentation of reference workflow	N/A	*Application-Specific Verification and Validation of Models and Generated PLC Code*
	Evidence for using the code generator within the referenced workflows and within the constraints specified in its certificate	N/A	Customized and completed *Conformance Demonstration Template*
Simulink Design Verifier	Tool certification evidence for model verification tool	• IEC 61508-3 Clause 7.4.4 • IEC 61508-3 Table A-3 (4a) "Certified tools and certified translators"	• Certificate Z10 11 06 67052 009 • Certificate report MN83534C
	Documentation of reference workflow	N/A	*Application-Specific Generation and Verification of Test Cases*
	Evidence for using the verification tool within the referenced workflows and within the constraints	N/A	Customized and completed *Conformance*

Products	Purpose	References	Artifacts and Documents[1]
	specified in its certificate		*Demonstration Template*
Simulink Verification and Validation	Tool certification evidence for model verification tool	• IEC 61508-3 Clause 7.4.4 • IEC 61508-3 Table A-3 (4a) "Certified tools and certified translators"	• Certificate Z10 11 06 67052 009 • Certificate report MN83534C
	Documentation of reference workflow	N/A	*Simulink Verification and Validation Reference Workflow*
	Evidence for using the verification tool within the referenced workflows and within the constraints specified in its certificate	N/A	Customized and completed *Conformance Demonstration Template*
Polyspace Client for C/C++; Polyspace Server for C/C++	Tool certification evidence for code verification tool	• IEC 61508-3 Clause 7.4.4 • IEC 61508-3 Table A-3 (4a) "Certified tools and certified translators"	• Certificate Z10 11 06 67052 011 • Certificate Report MN74651C
	Documentation of reference workflow	N/A	*Verification of C and C++ Code Using Polyspace Products*
	Evidence for using the verification tool within the referenced workflows and within the constraints	N/A	Customized and completed *Conformance*

Products	Purpose	References	Artifacts and Documents[1]
	specified in its certificate		*Demonstration Template*

[1]For file names and locations, see "IEC Certification Kit Components" on page 1-8.

3

Validating Software Tools

About Software Tool Validation

Some safety standards recommend the validation of software tools, using application-independent test cases to:

- Demonstrate that a software tool complies with its specified requirements.
- Examine the reaction of the software tool to anomalous operating conditions.

The IEC Certification Kit product provides exemplary test cases and test procedures that you can use to automate tool validation tests for the following products:

- Embedded Coder
- Simulink Verification and Validation (Model Coverage Analysis, Model Compliance Checking)

The exemplary test cases provided with the IEC Certification Kit product are templates that you can modify and extend to create test suites that cover the requirements that are relevant for your application, your specific tool configuration, operating environment, and so on.

Note MathWorks acknowledges the Automotive Code Validation Suite (AVS) as the initial test suite used with Embedded Coder, as described in the following article:

```
http://www.mathworks.com/company/pressroom/article31185.html
```

Running Test Cases and Procedures for Embedded Coder

To execute the test cases and procedures for Embedded Coder:

1. Copy the *matlabroot*`/toolbox/qualkits/iec/ecoder` folder and its subfolders to a location to which you have write access. Use that location to run the test cases and procedures.

   Note

   - To execute the test procedure, you must have an Embedded Coder license.
   - Some test models require Stateflow® and Simulink® licenses.

2. Open the file `certkitiec_ecoder_modelList.m` in the relocated folder.

3. Edit the file to specify the test cases (that is, test models and supporting files) that you want to execute. Check that the models and files that you specify exist in their specified locations in the `/tests` subfolder.

4. Optionally, edit the file to specify baselines corresponding to the tests. Check that the baselines that you specify exist in the `/baselines` subfolder.

5. Save the file.

6. To run the tests and generate a validation report, execute the file `certkitiec_ecoder_tests.m`. You can invoke it from the MATLAB® command line or in the Certification Artifacts Explorer. Test reports are generated in HTML format and are placed in the `/outputs` subfolder.

7. Review the generated test reports for correct results.

Running Test Cases and Procedures for Simulink Verification and Validation

To execute the test cases and procedures for Simulink Verification and Validation (Model Coverage Analysis, Model Compliance Checking):

1. Copy the *matlabroot*`/toolbox/qualkits/iec/slvnv` folder and its subfolders to a location to which you have write access. Use that location to run the test cases and procedures.

   Note

   - To run the tests and generate reports, you must have MATLAB® Report Generator™ and Simulink® Report Generator™ licenses.
   - Some model coverage RPT files require Simulink® Fixed Point™, Stateflow, and Simulink Design Verifier licenses.

2. Open the files `certkitiec_slvnv_tests*.xls` in the relocated folder.

3. Edit the files to specify the test cases (that is, test models and supporting files) that you want to execute, the expected results, and additional information. Check that the models and files that you specify exist in their specified locations in the `/tests` subfolder.

4. Save the files.

5. To run the tests and generate reports, execute the files `certkitiec_slvnv_tests*.rpt`. You can invoke them in the Certification Artifacts Explorer, from the MATLAB command line, or from the Report Explorer, as follows:

   - In the Certification Artifacts Explorer, right-click an RPT file and select **Execute Tests**.
   - At the MATLAB command line, enter the command

     ```
     report ('rpt_file')
     ```

     where *`rpt_file`* is the name of the test procedure.

- To open Report Explorer, double-click an RPT file, or in Certification Artifacts Explorer, right-click an RPT file and select **Open File**. In Report Explorer, select **File > Report**.

Simulink Report Generator creates the test reports and places them in the `/outputs` subfolder.

Note

- Before you execute model coverage RPT files, set the Java™ heap size for your MATLAB session to at least 512 MB. To check the Java heap size, open the MATLAB Preferences dialog box and select **General > Java Heap Memory**. If the **Java Heap Size** value is less than 512 MB, change it to 512 MB, click **OK**, and restart MATLAB. (If the maximum available heap size value is less than 512 MB, select the maximum value.) This may help you avoid `java.lang.OutOfMemoryError` messages.
- Before you execute each model coverage RPT file, start a new MATLAB session.

---

6 Review the generated test reports for correct results.

4

Accessing and Managing Certification Artifacts

Accessing Certification Artifacts Using the Certification Artifacts Explorer

In this section...

Certification Artifacts in the IEC Certification Kit Product

The IEC Certification Kit product includes the following certification artifacts:

- Certification and qualification evidence
- Documents and templates

For more information about the certification artifacts that are part of the IEC Certification Kit product, see "IEC Certification Kit Components" on page 1-8.

For more information about certifying or qualifying software tools, see "Certification Process Using the IEC Certification Kit Product" on page 2-2.

What Is a Certification Package?

A certification package is a group of certification artifacts that you use to certify your project. The Certification Artifacts Explorer displays:

- The certification artifacts that are part of the IEC Certification Kit product.
- Certification packages that you create.

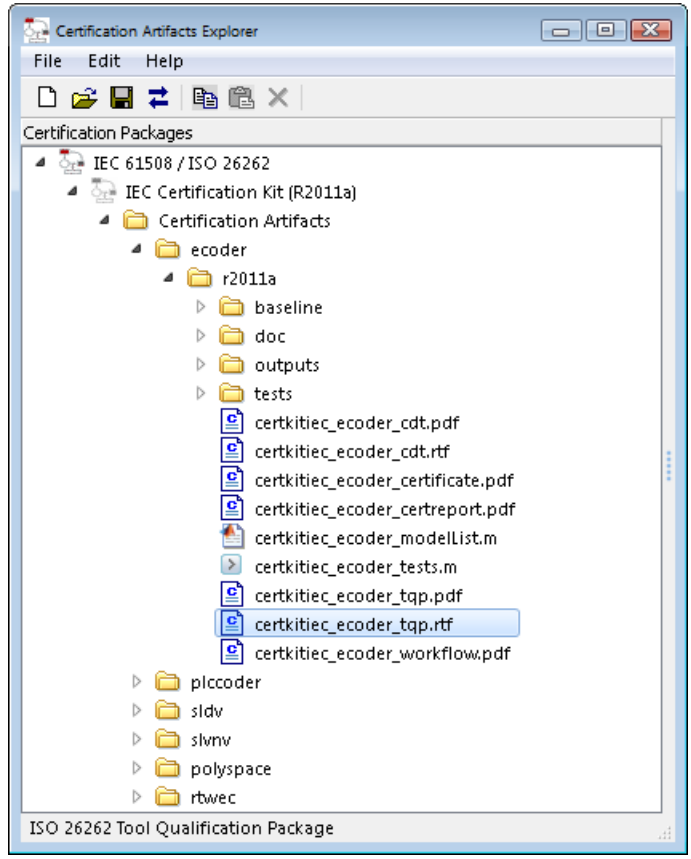
How To Access Certification Artifacts

You can use the Certification Artifacts Explorer to access certification artifacts. To start the Certification Artifacts Explorer, use one of the following methods.

To start the Certification Artifacts Explorer...	Do this:
From the MATLAB **Start** menu	Select **Start > Simulink > IEC Certification Kit > Certification Artifacts Explorer**.
From the MATLAB command line	Enter `certkitiec`.

The Certification Artifacts Explorer window displays the certification artifacts that are available with the IEC Certification Kit product. If the IEC Certification Kit product contains artifacts for more than one release, the Certification Artifacts Explorer lists the artifacts for each release. As you select folders and files, relevant information about the current selection is dynamically displayed in the status bar. Additionally,

- To display the properties of a certification package, right-click the package name and select **Properties**.
- To open an artifact, right-click the artifact and select **Open File**.



Managing Certification Artifacts Using the Certification Artifacts Explorer

In this section...

Managing Certification Artifacts Overview

To manage certification artifacts using the Certification Artifacts Explorer:

1. Create a new certification package.
2. Name the certification package.
3. Define the location where the Certification Artifacts Explorer stores the certification package.
4. Save the certification package. The saved package has a KIT extension.
5. Copy the certification artifacts for the product of interest into the certification package.
6. Delete certification artifacts that are not required for your project.
7. Optionally, add related files to the certification package using a file browser such as Microsoft® Windows® Explorer.

   Tip When you add files, to refresh the file list, use **File > Refresh**.

8. Use the Certification Artifacts Explorer to access certification artifacts. For a list of artifacts that you might need to access and modify, see "Certifying or Qualifying Software Tools" on page 2-2

When you create and save new certification packages, the Certification Artifacts Explorer displays them. The certification packages that are listed

remain visible unless you delete them from the Certification Artifacts Explorer.

Deleting Certification Packages

The Certification Artifacts Explorer displays all certification packages that you create or open. If you delete a certification package from the Certification Artifacts Explorer, the files associated with the package are still available on your computer. To delete the files, use a file browser such as Windows Explorer.

Limitations of the Certification Artifacts Explorer

The Certification Artifacts Explorer has the following limitations:

- The Certification Artifacts Explorer works on Microsoft Windows platforms only.
- For optimal performance, ensure that Microsoft Internet Explorer® is available on your machine. Internet Explorer does not have to be your default web browser.

5

Supporting Certification-Related Development Activities

Generating a Traceability Matrix

In this section...

About Traceability Matrices

When you use Model-Based Design and production code generation to develop application software components, you can generate a *traceability matrix*. The traceability matrix provides traceability among model objects, generated code, and model requirements. You can add comments to the generated traceability matrix. If you change the model and regenerate the traceability matrix, the software retains your comments.

For a given model, the generated traceability matrix can provide information about:

- Model objects that are traceable between the model and generated code, such as Simulink blocks, Stateflow objects, and MATLAB functions.
- Model objects that are untraceable between the model and generated code, such as eliminated and virtual blocks.
- Requirements documents that you link to model objects using the Simulink Verification and Validation Requirements Management Interface (RMI).

Generate the traceability matrix using either the `iec.ExportTraceReport` function from the MATLAB Command Window or the **Generate Traceability Matrix** button in the generated HTML code generation report for your model. Either method creates an XLS file that contains the following worksheets:

- **Report** — Traceability information for each model object, including model, generated code, and requirements. Each row in the worksheet pertains to a single occurrence of a model object. The information for a model object is in more than one row if the object:

  - Appears more than once in the generated code.
  - Links to more than one requirement.
- **Model Information** — Summary of the model configuration and checksum. The summary includes the model name, version, author, creation date, last saved by, last updated date, checksum, and the selection of **Traceability Report Contents** parameters.
- **Code Interface** — Information about the generated code interface, such as function prototype and timing information for the model initialize and step functions.
- **Code Files** — File folders and names of the generated code files.

Prerequisites for Generating a Traceability Matrix

Before generating a traceability matrix for model objects, generated code, and model requirements, perform the following steps:

1. Optionally, attach requirements documents. For more information, see "Requirements Traceability" in the Simulink Verification and Validation documentation.
2. In the Configuration Parameters dialog box, on the **Code Generation > Report** pane, select:
   a. **"Create code generation report"**
   b. At least one of the following **Traceability Report Contents** parameters:
      - **"Eliminated / virtual blocks"**
      - **"Traceable Simulink blocks"**
      - **"Traceable Stateflow objects"**
      - **"Traceable MATLAB functions"**

   Tip If you want to generate the traceability matrix directly from the code generation report, select **"Launch report automatically"**.
3. Generate code for the model.

Tip You do not have to build an executable to generate a traceability matrix. To generate code only, on the **Code Generation > General** pane, select **Generate code only**.

How to Generate a Traceability Matrix

To generate a traceability matrix:

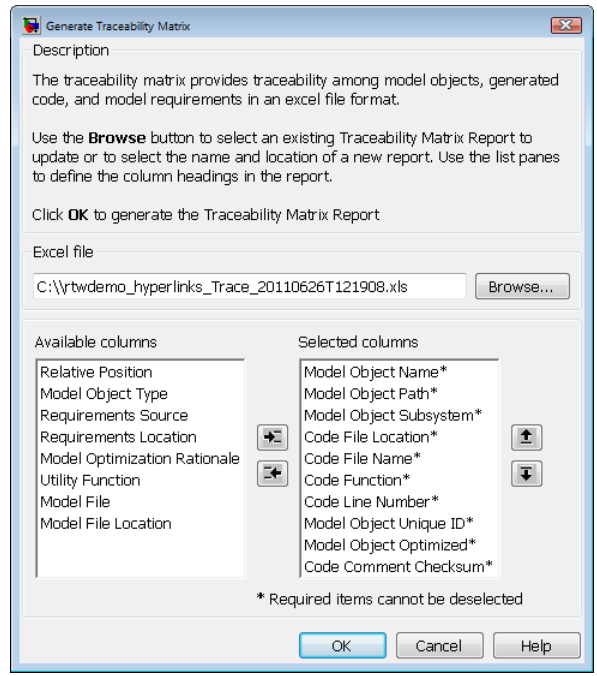
1. Open the model if it is not already open.
2. Ensure that you have completed the "Prerequisites for Generating a Traceability Matrix" on page 5-3.
3. Generate the traceability matrix using one of the following methods:
   - In the MATLAB Command Window, enter the following command, where *model_name* is the name of the model:

     ```
     iec.ExportTraceReport('model_name')
     ```

     The software generates the traceability matrix.
   - Open the code generation report for the model if it is not already open. Go to the **Traceability Report** section and click the **Generate Traceability Matrix** button. For example:

     When you click the button, the Generate Traceability Matrix dialog box appears.



You can use this dialog box to browse to an existing matrix file to update or specify a new matrix file to create. Optionally, you can also use this dialog box to select and order the columns that appear in the generated matrix. Click **OK** to update or create the specified report.

4 Review the traceability matrix and add comments in new columns. For more information, see "Adding Comments to a Traceability Matrix" on page 5-6.

Adding Comments to a Traceability Matrix

In this section...

Requirements for Adding Comments to a Traceability Matrix

You can add comments to the traceability matrix that you generated using the `iec.ExportTraceReport` function.

To add comments to the traceability matrix, you must:

- Create new columns for your comments.
- Use unique column headings. All columns that you add must have headings.
- Add at least one entry to the column, other than the column heading.
- Retain the following columns:
  - Model Object Name
  - Model Object Path
  - Model Object Subsystem
  - Code File Location
  - Code File Name
  - Code Function
  - Code Line Number
  - Model Object Unique ID
  - Model Object Optimized
  - Code Comment Checksum

Note All comments must resolve to a text string. For example, a link to an image resolves to a text string, but a copy of the image does not.

How To Retain Comments

To regenerate a traceability matrix and retain your comments:

1. Navigate to the working folder of the model.
2. Optionally, regenerate code for your model. Regenerating code before generating the traceability matrix ensures that you have the latest model-to-code traceability information.
3. In the MATLAB Command Window, enter the following command. *`file_name`* is the name of the existing traceability matrix that you are regenerating. If the existing traceability matrix is in a different folder, include the full path to that folder in *`path`*.

   ```
   iec.ExportTraceReport('model_name', 'file_name', 'path')
   ```

   The traceability matrix regenerates.

Traceability Matrix Limitations

The traceability matrix generation capability has the following limitations:

- Does not support generating a traceability matrix for referenced models. When you generate a traceability matrix for a model that contains referenced models, the traceability matrix contains information about the Model block only. The traceability matrix does not contain information about the contents of the referenced model. If your model contains referenced models, generate a traceability matrix for the top-level model and each referenced model separately.
- Works with the Microsoft Windows platform only.
- In most cases, identifies comments that you add to the traceability matrix, but when comments cannot be identified, the traceability matrix includes the text:

  Row is not unique: *`comment`*

- Does not support information stored in external `.req` files. For example, when you generate a traceability matrix for a model with externally stored requirements information, the traceability matrix does not include the requirements information.

6

Function Reference

Certification Artifacts Management

`certkitiec`	Open Certification Artifacts Explorer

Certification-Related Development Activities

`iec.ExportTraceReport`	Generate XLS file that contains traceability matrix

7

Functions — Alphabetical List

certkitiec

Purpose

Open Certification Artifacts Explorer

Syntax

```
certkitiec
```

Description

`certkitiec` opens the Certification Artifacts Explorer and displays certification artifacts.

Tips

- The `certkitiec` function works in Microsoft Windows platforms only.
- For optimal performance, ensure that Microsoft Internet Explorer is available on your machine. Internet Explorer does not have to be your default web browser.

Alternatives

Open the Certification Artifacts Explorer by selecting **Start > Simulink > IEC Certification Kit > Certification Artifacts Explorer**.

How To

- Chapter 4, "Accessing and Managing Certification Artifacts"
- "Certification Process Using the IEC Certification Kit Product" on page 2-2

Purpose

Generate XLS file that contains traceability matrix

Syntax

```
iec.ExportTraceReport('model_name')
iec.ExportTraceReport('model_name', 'file_name')
iec.ExportTraceReport('model_name', 'file_name', 'path')
```

Description

`iec.ExportTraceReport('model_name')` generates an XLS file that contains a "Traceability Matrix" on page 7-4. *`model_name`* is the name of the model.

`iec.ExportTraceReport('model_name', 'file_name')` generates an XLS file that contains a "Traceability Matrix" on page 7-4. *`file_name`* is a string that specifies the name of the file. The first time that you call `iec.ExportTraceReport`, *`file_name`* is optional. If you do not provide *`file_name`*, the function names the file using the following convention. *`modelUpdate`* is the date and time that you last updated the model:

`model_name``_Trace_`*`modelUpdate`*`.xls`

To regenerate the traceability matrix, you must specify *`file_name`*.

`iec.ExportTraceReport('model_name', 'file_name', 'path')` generates an XLS file that contains a "Traceability Matrix" on page 7-4. *`path`* is an optional string that specifies the full path to the location where you want the software to save the file.

Tips

- The `iec.ExportTraceReport` function works in Microsoft Windows platforms only.
- To include requirements documentation in the traceability matrix, attach requirements documents to the model before using `iec.ExportTraceReport`.
- You must generate a code generation traceability report (requires an Embedded Coder license) for your model before using `iec.ExportTraceReport`.
- The `iec.ExportTraceReport` function does not support generating a traceability matrix for referenced models. When you generate a traceability matrix for a model that contains referenced models,

the traceability matrix contains information about the Model block only. The traceability matrix does not contain information about the contents of the referenced model. If your model contains referenced models, generate a traceability matrix for the top-level model and each referenced model separately.

- In most cases, the `iec.ExportTraceReport` function identifies comments that you add to the traceability matrix. When the function cannot identify comments, the traceability matrix includes the text:

  Row is not unique: *`comment`*

For more information, see "Prerequisites for Generating a Traceability Matrix" on page 5-3.

Definitions

Traceability Matrix

A traceability matrix provides traceability among model objects, generated code, and model requirements. You can add comments to the generated traceability matrix. If you change the model and regenerate the traceability matrix, the software retains your comments.

Examples

Generate a traceability matrix with traceability between model objects and generated code for the `rtwdemo_hyperlinks` model:

Note This example requires an Embedded Coder license.

```
% Open the model.
open_system('rtwdemo_hyperlinks');
% Generate code only.
set_param('rtwdemo_hyperlinks', 'GenCodeOnly', 'on');
% Initiate the build process.
rtwbuild('rtwdemo_hyperlinks');
% Generate a traceability matrix.
iec.ExportTraceReport('rtwdemo_hyperlinks');
```

Generate a traceability matrix with traceability among model objects, generated code, and model requirements for the `slvnvdemo_fuelsys_docreq` model:

Note This example requires a Simulink Verification and Validation license.

```
% Open the model.
open_system('slvnvdemo_fuelsys_docreq');
% Select the code generation report and traceability report parameters.
set_param('slvnvdemo_fuelsys_docreq', 'GenerateReport', 'on');
set_param('slvnvdemo_fuelsys_docreq', 'GenerateTraceReport', 'on');
set_param('slvnvdemo_fuelsys_docreq', 'GenerateTraceReportSl', 'on');
set_param('slvnvdemo_fuelsys_docreq', 'GenerateTraceReportSf', 'on');
set_param('slvnvdemo_fuelsys_docreq', 'GenerateTraceReportEml', 'on');
% Generate code only.
set_param('slvnvdemo_fuelsys_docreq', 'GenCodeOnly', 'on');
% Initiate the build process.
rtwbuild('slvnvdemo_fuelsys_docreq');
% Generate a traceability matrix.
iec.ExportTraceReport('slvnvdemo_fuelsys_docreq');
```

Alternatives

You can generate a traceability matrix directly from the code generation report for your model. Go to the **Traceability Report** section and click the **Generate Traceability Matrix** button.

How To

- "Generating a Traceability Matrix" on page 5-2
- "Adding Comments to a Traceability Matrix" on page 5-6
- "Code Tracing"
- "Requirements Traceability"