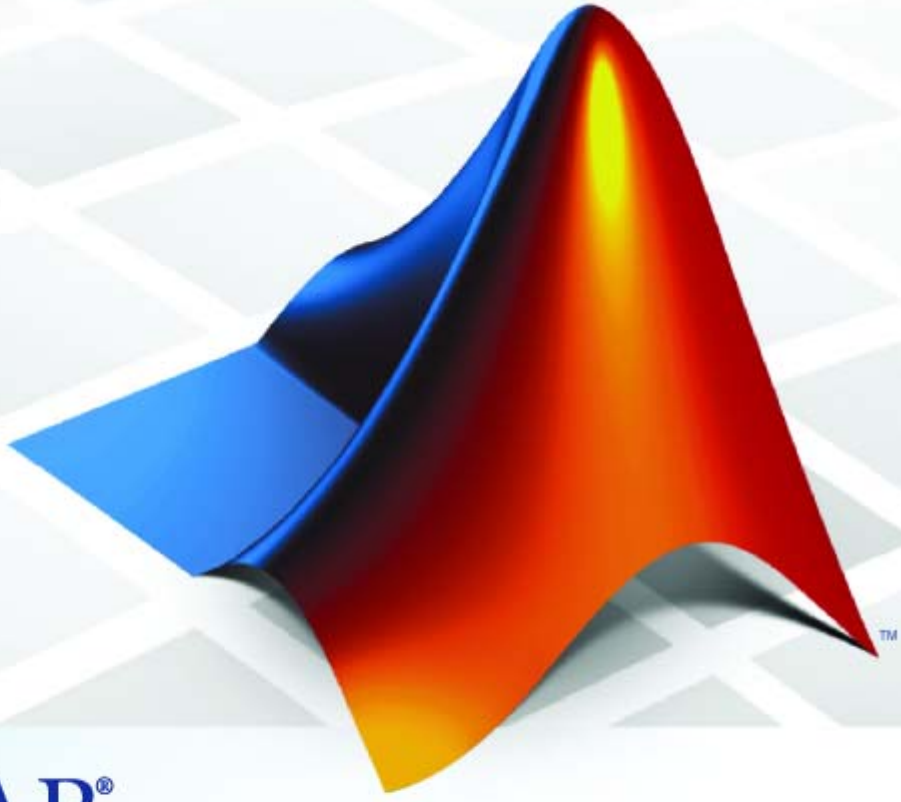
IEC Certification Kit 1
User's Guide



MATLAB®
&SIMULINK®

The MathWorks™
Accelerating the pace of engineering and science

How to Contact The 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

The 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 (For Releases 2007a+, 2008a, 2008b, 2009a)
September 2009	Online only	Revised for Version 1.1 (For Releases 2008a, 2008b, 2009a, 2009a+, 2009b)

Contents

1 Getting Started

IEC Certification Kit Product Overview 1-2
What Is the IEC Certification Kit Product? 1-2
What Is IEC 61508? 1-3
IEC 61508 Compliance Considerations 1-4
IEC 61508 Tool Certification Considerations 1-4
What Is ISO 26262? 1-5
ISO 26262 Compliance Considerations 1-6
ISO 26262 Tool Qualification 1-6
IEC Certification Kit Components 1-7
Required Knowledge 1-11

2 Certification Process

Certification Process Using the IEC Certification Kit Product 2-2
Defining Certification Objectives and Requirements 2-2
Certifying or Qualifying Software Tools 2-2

3 Supporting Certification-Related Development Activities

Generating a Traceability Matrix 3-2
About Traceability Matrices 3-2
Generating Traceability Matrices Limitations 3-3
Prerequisites for Generating a Traceability Matrix 3-3
How to Generate a Traceability Matrix 3-4

Adding Comments to a Traceability Matrix **3-5**
Requirements for Adding Comments to a Traceability Matrix **3-5**
How To Retain Comments **3-5**

4 Function Reference

Project Documentation **4-2**

5 Functions — Alphabetical List

1

Getting Started

IEC Certification Kit Product Overview

In this section...

"What Is the IEC Certification Kit Product?" on page 1-2
"What Is IEC 61508?" on page 1-3
"IEC 61508 Compliance Considerations" on page 1-4
"IEC 61508 Tool Certification Considerations" on page 1-4
"What Is ISO 26262?" on page 1-5
"ISO 26262 Compliance Considerations" on page 1-6
"ISO 26262 Tool Qualification" on page 1-6
"IEC Certification Kit Components" on page 1-7
"Required Knowledge" on page 1-11

What Is the IEC Certification Kit Product?

The IEC Certification Kit product (for IEC 61508 and ISO 26262) is a package of certification artifacts and tools. The IEC Certification Kit product supports engineers who use MathWorks™ products to develop, verify, or validate software for systems that must comply with, or be certified according to IEC 61508 or ISO 26262.

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.

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

What Is IEC 61508?

IEC 61508 is an international, industry-independent 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 from 1998 through 2000.

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 19 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 61508-3 is a generic safety publication first published in 1998. The standard does not cover advanced software development, verification, and validation technologies, such as Model-Based Design, code generation, and abstract interpretation. If developing software using such technologies, objectives and recommendations of the standard must be mapped to the processes and tools used.

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 not using a highly recommended technique or measure, 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

IEC 61508-3 highly recommends certified tools and translators for safety integrity levels SIL 2 and higher (see IEC 61508-3 clause 7.4.4.3a and table A.3). According to IEC 61508-7, clause C.4.3, wherever possible, tools should be certified. The certification of a tool is usually carried out by an independent body against independently set criteria, such as national or international standards.

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. IEC 61508 provides only limited guidance on how to satisfy the recommendation on how to certify tools. As a result, 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 workflow documentation in addition to a certificate and certificate report. The applicant must 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.

What Is ISO 26262?

ISO 26262 is an emerging international safety standard titled *Road vehicles — Functional safety*. ISO 26262 is a sector-specific standard for the automotive industry. The intention is to apply the ISO 26262 standard to safety-related systems. These systems include one or more *E/E systems*[1], and are installed in series production passenger cars with a maximum gross weight of up to 3.5 tons.

ISO® published the ISO/DIS 26262 draft international standard in July 2009. It consists of ten parts, referred to as ISO/DIS 26262-1 to ISO/DIS 26262-10.

Part 6 (ISO/DIS 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/DIS 26262-8) *Supporting processes* addresses multiple cross-functional topics, including the qualification of software tools.

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

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

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/DIS 26262 certification confirms that a product or system complies with the objectives set by the draft international standard. An applicant can certify a system in two ways:

- Certification by an independent, external certification authority.
- Self-certification

An independent, external certification authority, such as Technischer Überwachungsverein (TÜV) in Germany, can certify ISO/DIS 26262 compliance. The certification authority issues a certificate and, if applicable, a certificate report. The *certificate report* is a technical report that accompanies the certificate. The certificate report documents the details of the certification process and the constraints for the certificate.

An applicant can *self-certify* a system. Self-certification requires the applicant to demonstrate ISO/DIS 26262 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 applicable set of ISO/DIS 26262 requirements.

ISO 26262 Tool Qualification

The ISO 26262 standard acknowledges that the use of software tools simplifies or automates activities and tasks to develop safety-related software. ISO/DIS 26262-8 provides a framework for software tool 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/DIS 26262-8, Chapter 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 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, allows the selection of the appropriate qualification methods.

If the applicant can demonstrate the qualification requirements for the given software tool, no further qualification activities are needed. Otherwise, the applicant must apply 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 Certification Kit Components

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

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

- Real-Time Workshop® Embedded Coder™

- PolySpace® Client™ for C/C++; PolySpace® Server™ for C/C++

Specific versions of the preceding MathWorks products have been certified or qualified by TÜV SÜD, a German-based certification authority.

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 IEC 61508-3, ISO/DIS 26262-6, and ISO/DIS 26262-8.

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 3-2).

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

Certification Artifacts for the Real-Time Workshop Embedded Coder Product

TÜV SÜD has certified specific versions of the Real-Time Workshop Embedded Coder product for use in development processes that are required to comply with IEC 61508 or ISO/DIS 26262. These product versions are also qualified according to ISO/DIS 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 Real-Time Workshop Embedded Coder product:

- Version 5.3 (R2009a)
- Version 5.4 (R2009b)

Previous releases of the Real-Time Workshop Embedded Coder product are certified or qualified. For supporting certification artifacts, see previous releases of the IEC Certification Kit product.

Note The Real-Time Workshop Embedded Coder product was not developed using an IEC 61508 certified process.

Certification artifacts for the Real-Time Workshop Embedded Coder product are in the following folders, where *`release`* is `r2009a` or `r2009b`:

`matlabroot``\toolbox\qualkits\iec\rtwec\`*`release`*`\`

Details on the certification artifacts are in the certificate reports.

Release	Component	File
R2009a	Certificate	`Cert_Z10090667052002.pdf`
	Certificate Report	`CR_MN72051C.pdf`
	Workflow Documentation	`certkitiec_rtwec_workflow.pdf`
	Compliance Demonstration Template	`certkitiec_rtwec_cdt.rtf`
R2009b	Certificate	`Cert_Z10090667052002.pdf`
	Certificate Report	`CR_MN72051C.pdf`
	Workflow Documentation	`certkitiec_rtwec_workflow.pdf`
	Compliance Demonstration Template	`certkitiec_rtwec_cdt.rtf`
	ISO 26262 Tool Qualification Package	`certkitiec_rtwec_tqp.rtf` `certkitiec_rtwec_tqp.pdf`

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 IEC 61508, EN 50128, or ISO/DIS 26262. These product versions are also qualified according to ISO/DIS 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 5.1 (R2008a)
- Version 6.0 (R2008b)
- Version 7.0.1 (R2009a+)
- Version 7.1 (R2009b)

Previous releases of the PolySpace® products are certified or qualified. 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 folders, where *`release`* is `r2008a`, `r2008b`, `r2009a+`, or `r2009b`:

`matlabroot``\toolbox\qualkits\iec\polyspace\`*`release`*`\`

Release	Component	File
R2008a	Certificate	`Cert_Z10090767052003.pdf`
	Certificate Report	`CR_MN74651C.pdf`
	Workflow Documentation	`certkitiec_polyspace_workflow.pdf`
R2008b	Certificate	`Cert_Z10090767052003.pdf`
	Certificate Report	`CR_MN74651C.pdf`
	Workflow Documentation	`certkitiec_polyspace_workflow.pdf`
R2009a+	Certificate	`Cert_Z10090767052003.pdf`
	Certificate Report	`CR_MN74651C.pdf`
	Workflow Documentation	`certkitiec_polyspace_workflow.pdf`

Release	Component	File
R2009b	Certificate	`Cert_Z10090767052003.pdf`
	Certificate Report	`CR_MN74651C.pdf`
	Workflow Documentation	`certkitiec_polyspace_workflow.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:
  - IEC 61508 *Functional safety of electrical/electronic/programmable electronic safety-related systems*
  - ISO 26262 *Road vehicles — Functional safety*
  - EN 50128 *Railway Applications - Communications, Signalling and Processing Systems - Software for Railway Control and Protection Systems*
- Experience with MathWorks products that you use to develop, verify, or validate software for systems that are required to comply with the applicable standard.

Also, review the following information:

- Technical Solution 1-32COJP on the MathWorks Web site, which offers recommendations on how to apply Simulink®, Real-Time Workshop Embedded Coder, and other products for Model-Based Design in the context of IEC 61508.
- If you have a Real-Time Workshop Embedded Coder license,"Developing Models and Code That Comply with the IEC 61508 Standard" in the Real-Time Workshop Embedded Coder documentation.

2

Certification Process

Certification Process Using the IEC Certification Kit Product

In this section...

"Defining Certification Objectives and Requirements" on page 2-2

"Certifying or Qualifying Software Tools" on page 2-2

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 applicant and 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. For example, the tools and versions to certify or qualify.

Certifying or Qualifying Software Tools

The IEC 61508 and ISO 26262 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 qualification evidence for the following MathWorks products:

- Real-Time Workshop Embedded Coder

- 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 IEC 61508 and ISO 26262 applications. The applicant must ensure that the tools are used within the referenced workflows and constraints specified in the certificates.

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

Tool Certification Artifacts for IEC 61508 Applications

Products	Purpose	References	Artifacts and Documents[1]
Real-Time Workshop Embedded Coder	Tool Certification Evidence for code generator	• IEC 61508-3 Clause 7.4.4.3a • IEC 61508-3 Table A-3 Technique/Measure 5a "Certificated Translator"	• Certificate Z10 09 06 67052 002 • Certification report MN72051C
	Documentation of Translation validation workflow	N/A	*Application-Specific Verification and Validation of Models and Generated 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*

Tool Certification Artifacts for IEC 61508 Applications (Continued)

Products	Purpose	References	Artifacts and Documents[1]
PolySpace Client for C/C++; PolySpace Server for C/C++	Tool Certification Evidence for code verification tool	IEC 61508-3, Table A-3, Technique/Measure 4a "Certificated Tool"	• Certificate Z10 09 07 67052 003 • Certificate Report MN74651C
	Documentation of Code verification workflow	N/A	*Verification of C and C++ Code Using PolySpace Products*

[1]For file names and locations, see “IEC Certification Kit Components” on page 1-7.

Tool Qualification Artifacts for ISO 26262 Applications

Products	Purpose	References	Artifacts and Documents[1]
Real-Time Workshop Embedded Coder	Software Tool Qualification Plan	• ISO 26262-8, 11.4.2 • ISO 26262-8, 11.4.4	Customized and completed Chapter 2, *Software Tool Qualification Plan* of the *ISO 26262 Real-Time Workshop Embedded Coder Qualification Package* template
	Software Tool Documentation	ISO 26262-8, 11.4.2.2	• Customized and completed Chapter 3, *Software Tool Documentation* of the *ISO 26262 Real-Time Workshop Embedded Coder Qualification Package* template • Documentation set for the Real-Time Workshop Embedded Coder product • Installation Guide
	Software Tool Classification Analysis	ISO 26262-8, 11.4.2, 11.4.3	Customized and completed Chapter 4, *Tool Classification* of the *ISO 26262 Real-Time Workshop Embedded Coder Qualification Package* template

Tool Qualification Artifacts for ISO 26262 Applications (Continued)

Products	Purpose	References	Artifacts and Documents[1]
Real-Time Workshop Embedded Coder	Software Tool Qualification Report	ISO 26262-8, 11.4.3, 11.4.4, 11.4.5, 11.4.6, 11.4.7, 11.4.8	• Customized and completed Chapter 5, *Tool Qualification Documentation* of the *ISO 26262 Real-Time Workshop Embedded Coder Qualification Package* template • Certificate Z10 09 06 67052 002 • Certification report MN72051C
	Documentation of Translation validation workflow	N/A	*Application-Specific Verification and Validation of Models and Generated 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*

Tool Qualification Artifacts for ISO 26262 Applications (Continued)

Products	Purpose	References	Artifacts and Documents[1]
PolySpace Client for C/C++; PolySpace Server for C/C++	Software Tool Qualification Report	ISO 26262-8, 11	• Certificate Z10 09 07 67052 003 • Certificate Report MN74651C
	Documentation of Code verification workflow documentation	N/A	*Verification of C and C++ Code Using PolySpace Products*

[1]For file names and locations, see "IEC Certification Kit Components" on page 1-7.

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.

3

Supporting Certification-Related Development Activities

- "Generating a Traceability Matrix" on page 3-2
- "Adding Comments to a Traceability Matrix" on page 3-5

Generating a Traceability Matrix

In this section...

"About Traceability Matrices" on page 3-2

"Generating Traceability Matrices Limitations" on page 3-3

"Prerequisites for Generating a Traceability Matrix" on page 3-3

"How to Generate a Traceability Matrix" on page 3-4

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 Embedded MATLAB® scripts.
- 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 the `iec.ExportTraceReport` function. The function 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 Files** — File folders and names of the generated code files.

Generating Traceability Matrices Limitations

The `iec.ExportTraceReport` function that you use to generate traceability matrices has the following limitations:

- 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.
- The `iec.ExportTraceReport` function works with the Microsoft® Windows® platform only.
- 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`*

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 “Managing Model Requirements” in the Simulink Verification and Validation documentation.
2. In the Configuration Parameters dialog box, on the **Real-Time Workshop > 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 Embedded MATLAB functions"**

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 **Real-Time Workshop > General** pane, select **Generate code only**.

How to Generate a Traceability Matrix

To generate a traceability matrix:

1 Open the model.

2 Ensure that you have completed the "Prerequisites for Generating a Traceability Matrix" on page 3-3.

3 In the MATLAB® Command Window, enter the following command to generate the traceability matrix, where *model_name* is the name of the model:

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

The software generates the traceability matrix.

4 Review the traceability matrix and add comments in new columns. For more information, see "Adding Comments to a Traceability Matrix" on page 3-5.

Adding Comments to a Traceability Matrix

In this section...

"Requirements for Adding Comments to a Traceability Matrix" on page 3-5

"How To Retain Comments" on page 3-5

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:
  - Code File Name
  - Code Function
  - Requirements Source
  - Model Object SID
  - 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.

4

Function Reference

Project Documentation (p. 4-2) Document generated code

Project Documentation

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

5

Functions — Alphabetical List

iec.ExportTraceReport

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 5-3. *`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 5-3. *`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 5-3. *`path`* is an optional string that specifies the full path to the location where you want the software to save the file.

Usage

- 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 a Real-Time Workshop 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 3-3.

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 a Real-Time Workshop 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');
```

How To

- “Generating a Traceability Matrix” on page 3-2
- “Adding Comments to a Traceability Matrix” on page 3-5
- “Traceability for Production Code Generation”
- “Managing Model Requirements”