

Model-Based Design for DO-178C/DO-331 Compliance

OEM: MathWorks • Code: MW-MODEL-BASED-DESIGN-FOR-DO-178C-DO-331

COURSE DESCRIPTION

This three-day course is intended for software engineers that are using MathWorks tools to generate production code intended for DO-178C certification. The course assumes prior knowledge of Simulink modeling principles and verification workflows in Simulink and Polyspace. The course focuses on generating artifacts from those workflows that can be used in the DO certification process. Note that this training course does not cover the DO-178C or DO-331 standards themselves, but rather how to use MathWorks tools to demonstrate DO-178C and DO-331 compliance.

AUDIENCE PROFILE

- IT professionals and technical staff seeking hands-on training in this course's subject area.

PREREQUISITES

- Simulink Fundamentals (or Simulink Fundamentals for Aerospace Applications), Simulink Model Management and Architecture, and Simulation-Based Testing with Simulink. Knowledge of the DO-178C standard and its DO-331 supplement for model-based software development. As needed: Stateflow for Logic-Driven System Modeling, Design Verification with Simulink, Embedded Coder for Production Code Generation, and Polyspace for C/C++ Code Verification.

COURSE OBJECTIVES

By the end of this course, participants will be able to:

- Understand how MathWorks workflows can satisfy and support DO-178C and DO-331 objectives.
- Trace a model to requirements according to DO-178/DO-331 compliance.
- Understand conformance to modeling standards for DO-178C/DO-331.
- Learn to verify software requirements through simulation.
- Utilize Simulink Coverage to collect model coverage data.
- Learn how to utilize Simulink Design Verifier for low level requirement verification.
- Use Simulink Code Inspector to demonstrate structural and algorithmic equivalence between a model and its generated code.
- Show that generated code is compliant with coding standards and prove that it is robust against various run-time errors.
- Use Simulink Test to demonstrate complete code coverage via software requirements test cases, and verify software requirements on the target hardware.

- Use DO Qualification Kit to qualify MathWorks products and features for DO-178C certification.

COURSE OUTLINE

Module 1: Day 1 of 3

Model-Based Design for DO-178C/DO-331 Overview

- Overview of system, software, and hardware life cycles
- DO-178C/DO-331 workflow with Model-Based Design
- Satisfying DO-178C/DO-331 objectives using Model-Based Design

Requirements Traceability

- Requirements overview in the DO-178C process
- Using Requirements Toolbox for requirement traceability
- Generating a System Design Description (SDD)
- Understanding compatibility with external requirement tools

Modeling Standards for DO-178C/DO-331

- Modeling standards for DO-178C/DO-331 applications
- Modeling standards in the Model-Based Design (MBD) process
- Using Simulink Check pre-packaged rules for high-integrity systems

Module 2: Day 2 of 3

Model Verification

- Test cases authoring in Simulink Test
- Test case inputs and data collection options
- Tracing test cases to requirements
- Requirements-based testing with Simulink Test

Model Coverage

- Model coverage metrics
- Capturing model coverage with Simulink Coverage
- Missing model coverage resolution

Derived Requirements and Design Verification

- Simulink Design Verifier (SLDV) to identify unreachable elements
- Developing low-level tests for missing Model coverage
- Model coverage reporting

Module 3: Day 3 of 3

Model-Code Verification and Traceability

- Code inspection overview
- Checking model compatibility for Simulink Code Inspector
- Verifying model-to-code traceability

Static Code Analysis and Formal Code Verification

- Polyspace tools overview
- Running Polyspace analyses from Simulink
- Enforcing coding standards with Polyspace Bug Finder
- Proving code robustness with Polyspace Code Prover
- Demonstrating control and data coupling coverage with Polyspace Code Prover

Code Verification Against Software Requirements

- Software-in-the-loop (SIL) and processor-in-the-loop (PIL) overview
- Testing executable object code against high-level requirements
- Collecting code coverage

Tool Qualification

- Tool qualification requirements
- DO Qualification Kit overview
- Tool qualification example
- Qualifiable tools and credits taken
- DO Qualification Kit customization