# CertifKit workshop Project

## Concept

The repository contains all the presentations, scripts, models, report templates, configuration files required to present a ‘proposed example’ activity workflow to follow a certification standard for projects using MBD.

The idea of the demo is to show all the tools and functions in the MathWorks tool suite that can be used to achieve the certification of the customer’s project and design to a given standard.

The presentation can be used to present the standard and the required activities. The scripts and models are used to demonstrate the activity on a simple design.

The currently supported standards are:

* ISO26262
* DO178
* IEC62304
* DO254 (in progress)
* EN50128, IEC62279 (in progress)

Repository is located here: <http://svn.mi.mathworks.com/general_psp/HighIntegrityWorkshops/>

In the workshop, the following tools are demonstrated:

* SimulinkProject
* Report Generator, WebView reports
* Model Advisor
* Simulink Design Verifier
* Simulink: Requirement linking and reports
* Simulink: Simulation on Host
* Coverage tools
* Simulink Code Inspector
* Simulink: Simulation in SIL mode

# How to Prepare

### Set-up

Check out the demo from the repository above.

#### Option 1: (use when developing the workshop material)

To demonstrate a given standard, go into the corresponding /Certif/<standard> folder and startup the Simulink project in there. For example: /Certif/DO178/Autopilot\_DO178.prj.

In this mode, the project is linked to the development repository and any work you do to the project can be checked into the development workshop repository. Here, shortcuts and labels should be managed so they are setup the way that should server as a baseline for running the customer presentations.

#### Option 2: Used for customer presentations

For customer presentations the workshop should be extracted into a separate directory, this can be done with the startupWorkshop shortcut. This will copy all of the workshops into a separate folder and create a local repository to run against.

Having a separate local repository is essential because it allows you to demonstrate the workshop even if you are not connected to the MathWorks network.

# How to present and run the demo

Open the PowerPoint presentation and present each activity.

When needed (and depending on the available time), you may:

* Execute the shortcut on some or all of the model then show the newly created report
* Show the pre-canned reports (if you don’t have time to produce the report now or it you think it is unnecessary because of the nature your audience).

### Tip and Tricks / What may interest your audience:

#### WebView

Many customers are not aware of the existence of the WebView. It allows browsing in the model with a simple (SVG-enabled) web browser. Producing a WebView allows auditors to review the model even if they don’t have a Matlab/Simulink installation.

💣 Make sure you have an SVG enabled browser (I use Firefox, the SVG plugin for Explorer works too).

#### Traceability reports

Some customers don’t know about the requirements linking, the possible highlighting and reporting. It is useful to demonstrate that the tool can be used to produce a 100% bi-directional report between the high-level requirement and the design.

You may also explain and show that requirements are added in the source code.

You may demonstrate how Model Advisor can be used to checks whether the text for a requirement is still matching with the model (and therefore detect necessary rework / rereview).

#### Code Inspector

The code inspector is a relatively new product and it is quite easy to show how powerful it is by manually introduced various types of errors.

#### Coverage

To run the sX\_runEquivalenceTests script, we have created with sldv a uses BullsEyesCoverage. It is useful to show the coverage on the model (all blocks involving conditions are highlighted with the colour showing our much the conditions were exercised).

💣 You may not be able to show the coverage if you don’t have a licensed copy of BullsEye on your machine.

💣 Even though the workshop uses Bullseye, don’t forget to mention our other partner LDRA.

## Folder layout

|  |  |
| --- | --- |
| **Folder** | **What’s in it** |
| Common | All common[[1]](#footnote-1)\* scripts, scripts that may be run during a customer demo. |
| Common/private | All private scripts used by the above. |
| Common/ReportTemplates | Report Generator templates for the SDD, MRD.  Requirements.  Equivalence test reports. |
| Certif/<standard>/<Name>\_<standard>.prj | Project file to start up the workshop for <standard> |
| Certif/<standard>/ProjectManagement | Project files to use for startup and shutdown, keeps the folders to be added to path, data to be loaded etc |
| Certif/<standard>/Artefacts | A few directories where the artefacts are added when scripts are executed. |
| Certif/<standard>/MAConfig | ModelAdvisor configuration files, conventionally suffixed with the Matlab version to make it possibly to run on any version. |
| Certif/<standard>/Model | Models used to present a given standard.  Design models  Contain the actual design that will be in the controller’s software. (Must have a Top level and may have a few referenced models or libraries)  Plant Models  Used to simulate the design in a plant  Validation Models  Used to test the model in simulation. We have created harness models with SLDV.  EquivalenceTestingModels  Typically the same model as the validation model. It is used to  Supporting scripts  Any script need to set-up the workspace, callbacks,….  It may have subdirectories if needed. |
| Certif/<standard>/Model/Scripts | Scripts required for this model (e.g. Bus and Enum definitions, Workspace initialisation, |
| Certif/<standard>/Model/Tests | Test models for the design components. |
| Certif/<standard>/Model/C\_<composition>/ | Composition level (assembly of models that constitute a larger piece of functionality in the system) |
| Certif/<standard>/Model/C\_<composition>/I\_<model> | Integration level model containing the architecture of the functions (below) |
| Certif/<standard>/Model/C\_<composition>/I\_<model>/Tests | Tests for the integration level model |
| Certif/<standard>/Model/C\_<composition>/F\_<function> | Function (or module/unit…) level, part of the composition and integration |
| Certif/<standard>/Model/C\_<composition>/F\_<function>/Tests | Tests for the function. Contains validationModels and equivalence test models. These both implement test against requirements |
| Certif/<standard>/Requirements | The High Level requirements for your design |
| Certif/<standard>/Utilities | Shortcuts that link back to the common shortcuts in Common/ |
| Certif/<standard>/Presentation | PowerPoint presentation(s) for that standard. |
| Certif/work | Location where the slprj, xxx\_ert\_rtw folders are redirected. |
| Devel | Scripts used to prepare a local repository for standards (ready to demonstrate away from the MathWorks network). |

## Categories and labels identifying files

#### Impact of the “Type” Category

This is to be applied to design and verification elements to be able to filter them

|  |  |  |
| --- | --- | --- |
| **“Type” Category Label** | **Rational** | **Impact** |
| DesignModel | These are the design elements i.e. the “functions” |  |
| EquivalenceTestingModel | These are used to do equivalence testing | This label needs to be added to the F\_<function>/Tests/<function>\_harness\_req model to enable it for running equivalenceTest shortcut |
| PlantModel | Bring together the plant and the design. |  |
| TopModel | Integration level model, to be generated code from | Will be used by the shortcut generateCode |
| ValidationModel | Test models for functions | Will be used by the shortcut runModelTests |

#### Impact of the “<standard>” Category

This is to be applied to the evidence of running verification steps

|  |  |  |
| --- | --- | --- |
| **“<standard>” Category Label** | **Rational** | **Impact** |
| Data | Data Artefacts | Will be applied to the artefacts that constitute data such as coverage data etc |
| Report | Report Artefacts | Will be applied to the artefacts that are reports such as coverage reports |

# How to add support for an additional standard

|  |  |
| --- | --- |
| **Folder** | **What you need to do for a new standard** |
| Certif/<standard>/<ModelName>\_<standard>.prj | Project file for the project, add this by creating a new project connect it to SVN, then add the folders below as you go along. |
| Certif/<standard>/ProjectManagement/ | Keeps the startup and shutdown files that should be used to add paths etc. Copy from another standard. This will also automatically add the necessary “Type” tags for the standard. |
| Certif/<standard>/Utilities | Add the s<n> scripts that links back to the shortcuts in the Common folder. Add to the shortcuts in the project |
| Certif/<standard>/Artefacts | Create at least Code, CodeTest, ModelReview and ModelTest.  If your standard has extra types of artefacts, create the empty directory and choose the correct location for addWorkProduct in your added s<n>\_script.m files. |
| Certif/<standard>/MAConfig | Create at least one MA configuration, possibly a full one and a cut-down one (because the full test is likely to take too long to execute in public). |
| Certif/<standard>/Model | Create a model relevant to the industry the standard applies to.  Add as many sub-directories, .mat, .m as you like.  Set a ‘Type’ label for the Top level model (the one that is used for code generation.  Set ‘Type’ labels for test models accordingly |
| Certif/<standard>/Requirements | Your requirements, we suggest that you use word or Excel as a demo with Doors would be difficult to use outside the MathWorks network. |
| Certif/<standard>/Scripts | Create at least  getFolders.m (returns the list of folders that needs to be in the path)  SetupScript.m (enables the shortcuts for the |
| Certif/<standard>/Presentation | Add your PowerPoint presentation and optionally supporting Word documents. |

# Gotcha and to do

Works only with SVN: Although relying on Simulink Project, there are some SVN commands called directly in the report generator activities.

Utilities and Utilities/private: They are not well separated and it could be cleaned.

Doors requirements: Although the work was done by Fraser to have the DO178 project use Doors, it is not yet supported in the trunk. It may be difficult to have a demo that allows us to use sometimes Word and sometimes Doors.

Shortcut cleanup: The clean\_shortcuts script could be simplified and simply remove all shortcut (as the set-up will restore the correct ones).

Dataset vs. logsOut: The harnesses that we created for ISO26262 and DO178 use logsOut to store the simulation results. logsOut is an obsolete and to-be-retired functionality. Eventually, we have to update the models and rewrite partially the comparison scripts.

Tests: The test vectors and the scripts to verify the results were written by SLDV “Generate Tests”. That does not meet the requirement in DO178 that says that tests must be derived from requirements. Also, it makes a strict equality check between an expected output and the simulation output. In practise, we should make tests derived from requirements and that have some tolerance in them.

Directory structure: The demonstration would be easier to run if the directory was organised by activity rather than by components. See proposal in appendix.

Presentation: The presentation for DO178 (and probably all the others as well) could be improved if we added a visible mark on each slide to say which activity it relates to. (e.g. a V logo with a dot in the correct place…)

## Appendix: New folder layout by Activity

### Layout

|  |  |
| --- | --- |
| **Folder** | **What** |
| Certif/<standard>/Requirements | High level requirements |
| Certif/<standard>/Presentation | PowerPoint presentation |
| Certif/<standard>/1\_Requirements | The work document.  Supporting high-level models |
| Certif/<standard>/2\_Design | Plant models  Design models |
| Certif/<standard>/3\_Verification | Harness for model verification (host simulation)  Model Advisor configuration  Report templates |
| Certif/<standard>/4\_CodeGeneration\_and\_Verification | Code templates  Code gen configuration  Workspace data  Code Replacement scripts  Harness for model verification (SIL) |
| Certif/<standard>/5\_Certification | Process documents  Archiving Scripts, CM related scripts. |

Scripts may now be in the relevant activity.

A Utility directory will remain necessary to store all common utility scripts/

### Use tags to identify the models:

In the current model, we have relied on naming convention and directory structure to help us recognise the test model from the design models. This is underutilising the power of SimulinkProject.

We propose to use tags to earmark the top model, various test and plant models. All shortcut scripts can use a new utility scripts to locate the correct model.

1. common means common to all standards and models. [↑](#footnote-ref-1)