ASCET-DEVELOPER 7.9

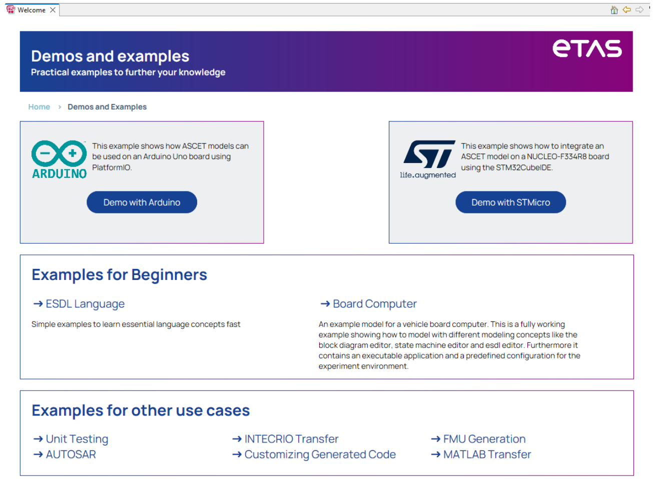
New and Noteworthy

Improve self on boarding and knowledge

New Welcome Page and Tutorials

Pain Point
☹ No easy way to self learn w/o help from ETAS Support
☹ Are there good reference examples for my use cases
☹ Where do I get help, ask questions

Benefit
☺ User friendly and good User Experience



Demos and examples
Practical examples to further your knowledge

Home › Demos and Examples

This example shows how ASCET models can be used on an Arduino Uno board using PlatformIO.
Demo with Arduino

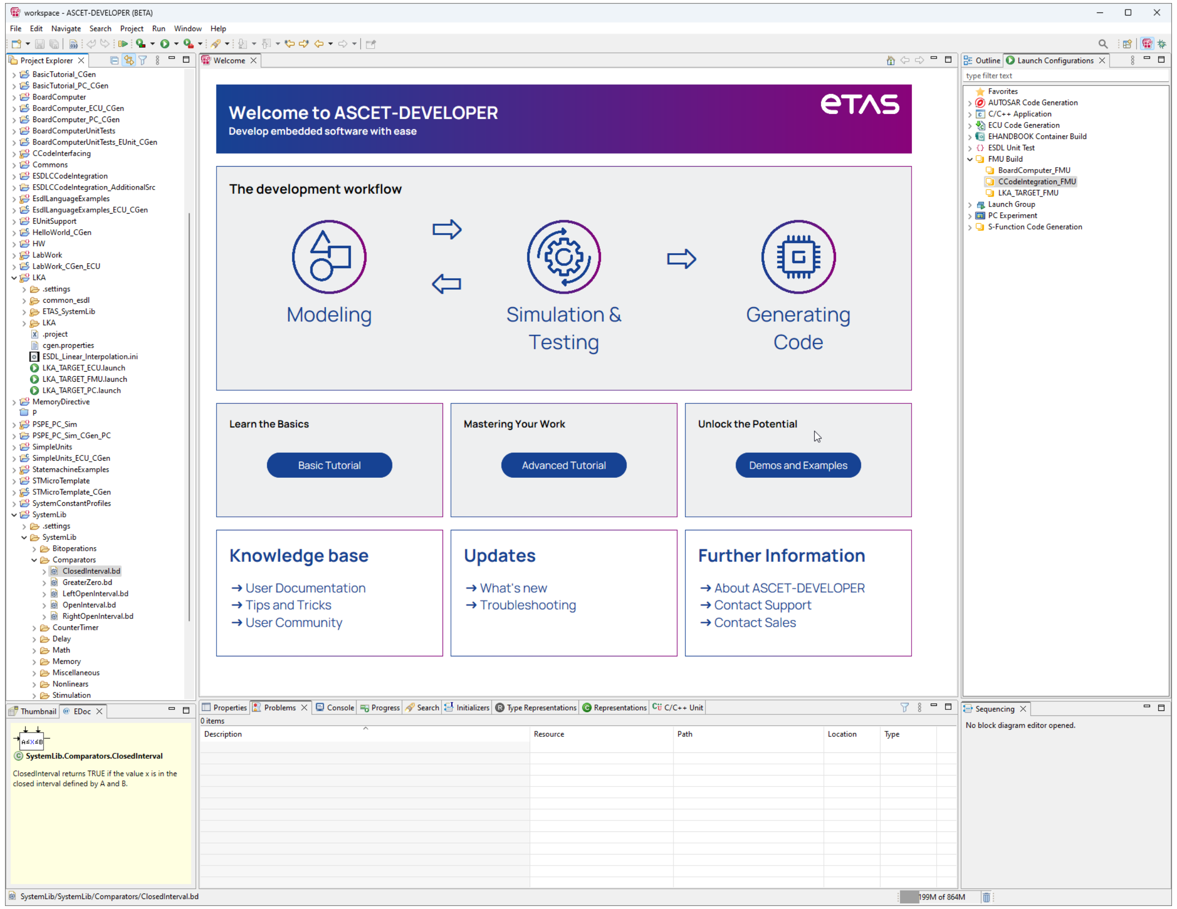
This example shows how to integrate an ASCET model on a NUCLEO-F334R8 board using the STM32CubeIDE.
Demo with STMicro

Examples for Beginners

→ ESDL Language
Simple examples to learn essential language concepts fast

→ Board Computer
An example model for a vehicle board computer. This is a fully working example showing how to model with different modeling concepts like the block diagram editor, state machine editor and esdl editor. Furthermore it contains an executable application and a predefined configuration for the experiment environment.

Examples for other use cases
→ Unit Testing
→ AUTOSAR
→ INTECRIO Transfer
→ Customizing Generated Code
→ FMU Generation
→ MATLAB Transfer



BasicTutorial_CGen
BasicTutorial_PC_CGen
BoardComputer
BoardComputer_ECU_CGen
BoardComputer_PC_CGen
BoardComputerUnitTests
BoardComputerUnitTests_EUnit_CGen
CCodeInterfacing
Commons
ESDLCCodeIntegration
ESDLCCodeIntegration_AdditionalSrc
EsdlLanguageExamples
EsdlLanguageExamples_ECU_CGen
EUnitSupport
HelloWorld_CGen
HW
LabWork
LabWork_CGen_ECU
LKA
  .settings
  common_esdl
  ETAS_SystemLib
  LKA
  .project
  cgen.properties
  ESDL_Linear_Interpolation.ini
  LKA_TARGET_ECU.launch
  LKA_TARGET_FMU.launch
  LKA_TARGET_PC.launch
MemoryDirective
P
PSPE_PC_Sim
PSPE_PC_Sim_CGen_PC
SimpleUnits
SimpleUnits_ECU_CGen
StatemachineExamples
STMicroTemplate
STMicroTemplate_CGen
SystemConstantProfiles
SystemLib
  .settings
  SystemLib
    Bitoperations
    Comparators
      ClosedInterval.bd
      GreaterZero.bd
      LeftOpenInterval.bd
      OpenInterval.bd
      RightOpenInterval.bd
    CounterTimer
    Delay
    Math
    Memory
    Miscellaneous
    Nonlinears
    Stimulation

Welcome to ASCET-DEVELOPER
Develop embedded software with ease

The development workflow
Modeling ⇄ Simulation & Testing ⇨ Generating Code

Learn the Basics
Basic Tutorial

Mastering Your Work
Advanced Tutorial

Unlock the Potential
Demos and Examples

Knowledge base
→ User Documentation
→ Tips and Tricks
→ User Community

Updates
→ What's new
→ Troubleshooting

Further Information
→ About ASCET-DEVELOPER
→ Contact Support
→ Contact Sales

Favorites
AUTOSAR Code Generation
C/C++ Application
ECU Code Generation
EHANDBOOK Container Build
ESDL Unit Test
FMU Build
  BoardComputer_FMU
  CCodeIntegration_FMU
  LKA_TARGET_FMU
Launch Group
PC Experiment
S-Function Code Generation

SystemLib.Comparators.ClosedInterval
ClosedInterval returns TRUE if the value x is in the closed interval defined by A and B.

0 items

Description	Resource	Path	Location	Type

No block diagram editor opened.

SystemLib/SystemLib/Comparators/ClosedInterval.bd
199M of 864M

Improve self on boarding and knowledge

Improved Wizard Help and new Edoc view

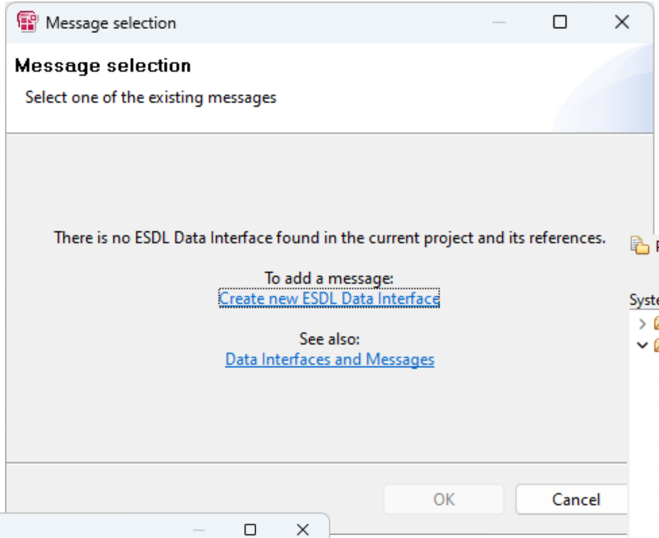
Pain Point

☹ When I click on BD Editor palette button, it is empty, I don't understand what is tool expecting

☹ If I want to understand the type description, I need to open the definition and read the comment manually.

Benefit

☺ User friendly and good User Experience


Message selection

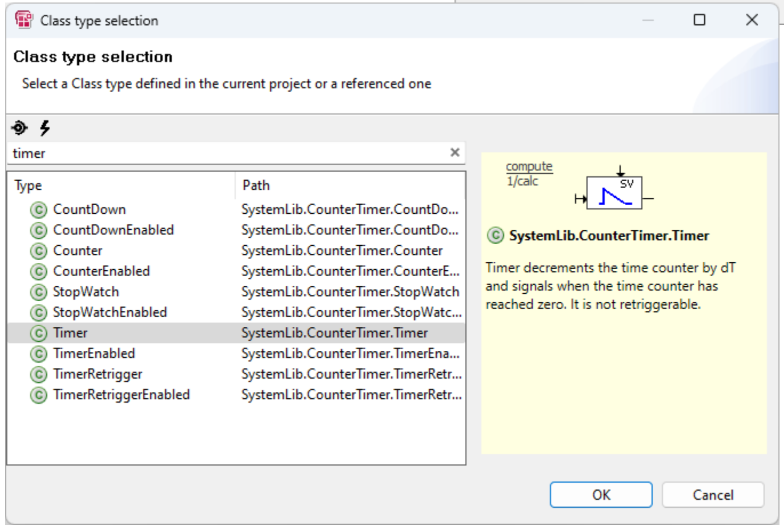
Message selection

Select one of the existing messages

There is no ESDL Data Interface found in the current project and its references.

To add a message:
Create new ESDL Data Interface

See also:
Data Interfaces and Messages


Class type selection

Class type selection

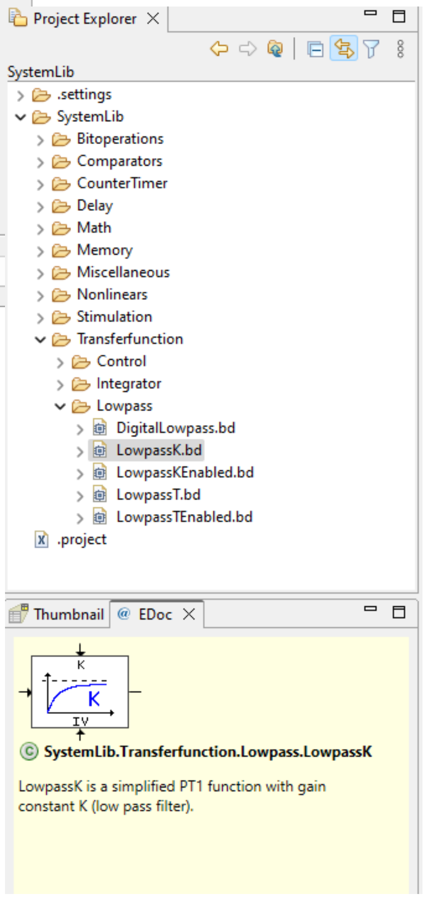
Select a Class type defined in the current project or a referenced one

timer

Type	Path
CountDown	SystemLib.CounterTimer.CountDo...
CountDownEnabled	SystemLib.CounterTimer.CountDo...
Counter	SystemLib.CounterTimer.Counter
CounterEnabled	SystemLib.CounterTimer.CounterE...
StopWatch	SystemLib.CounterTimer.StopWatch
StopWatchEnabled	SystemLib.CounterTimer.StopWatc...
Timer	SystemLib.CounterTimer.Timer
TimerEnabled	SystemLib.CounterTimer.TimerEna...
TimerRetrigger	SystemLib.CounterTimer.TimerRetr...
TimerRetriggerEnabled	SystemLib.CounterTimer.TimerRetr...

SystemLib.CounterTimer.Timer

Timer decrements the time counter by dT and signals when the time counter has reached zero. It is not retriggerable.


Project Explorer

```
SystemLib
> .settings
v SystemLib
  > Bitoperations
  > Comparators
  > CounterTimer
  > Delay
  > Math
  > Memory
  > Miscellaneous
  > Nonlinears
  > Stimulation
  v Transferfunction
    > Control
    > Integrator
    v Lowpass
      > DigitalLowpass.bd
      > LowpassK.bd
      > LowpassKEnabled.bd
      > LowpassT.bd
      > LowpassTEnabled.bd
  .project
```

SystemLib.Transferfunction.Lowpass.LowpassK

LowpassK is a simplified PT1 function with gain constant K (low pass filter).

Improve context sensitive help

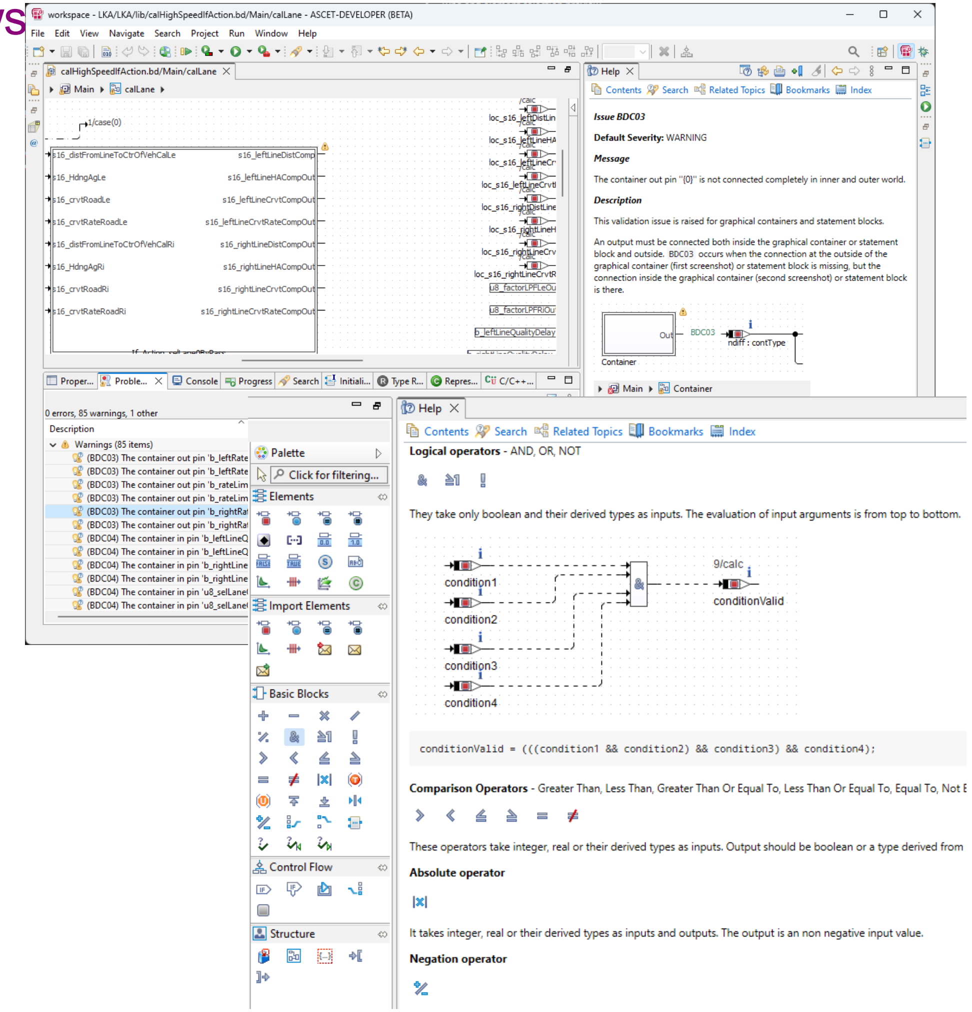
Problems View , Editor and other views

Pain Point

☹ Validation text is not easy to understand
☹ How does this view work?
☹ What is this palette button for?
☹ Time consuming to find description in user manual

Benefit

☺ Get help faster
☺ Select the issue or something in palette and press F1



workspace - LKA/LKA/lib/calHighSpeedIfAction.bd/Main/calLane - ASCET-DEVELOPER (BETA)
File Edit View Navigate Search Project Run Window Help
calHighSpeedIfAction.bd/Main/calLane
Main › calLane
Proper... Proble... Console Progress Search Initiali... Type R... Repres... C/C++...
Palette
Click for filtering...
Elements
Import Elements
Basic Blocks
Control Flow
Structure
Help
Contents Search Related Topics Bookmarks Index

0 errors, 85 warnings, 1 other

Description

Warnings (85 items)
(BDC03) The container out pin 'b_leftRate
(BDC03) The container out pin 'b_leftRate
(BDC03) The container out pin 'b_rateLim
(BDC03) The container out pin 'b_rateLim
(BDC03) The container out pin 'b_rightRa
(BDC03) The container out pin 'b_rightRa
(BDC04) The container in pin 'b_leftLineQ
(BDC04) The container in pin 'b_leftLineQ
(BDC04) The container in pin 'b_rightLine
(BDC04) The container in pin 'b_rightLine
(BDC04) The container in pin 'u8_selLane
(BDC04) The container in pin 'u8_selLane

Issue BDC03

Default Severity: WARNING

Message

The container out pin "{0}" is not connected completely in inner and outer world.

Description

This validation issue is raised for graphical containers and statement blocks.

An output must be connected both inside the graphical container or statement block and outside. BDC03 occurs when the connection at the outside of the graphical container (first screenshot) or statement block is missing, but the connection inside the graphical container (second screenshot) or statement block is there.

Logical operators - AND, OR, NOT

They take only boolean and their derived types as inputs. The evaluation of input arguments is from top to bottom.

```
conditionValid = (((condition1 && condition2) && condition3) && condition4);
```

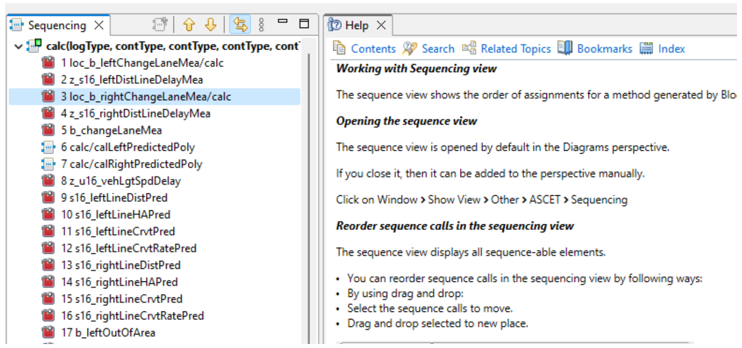
Comparison Operators - Greater Than, Less Than, Greater Than Or Equal To, Less Than Or Equal To, Equal To, Not E

These operators take integer, real or their derived types as inputs. Output should be boolean or a type derived from

Absolute operator

It takes integer, real or their derived types as inputs and outputs. The output is an non negative input value.

Negation operator



calc(logType, contType, contType, contType, cont
1 loc_b_leftChangeLaneMea/calc
2 z_s16_leftDistLineDelayMea
3 loc_b_rightChangeLaneMea/calc
4 z_s16_rightDistLineDelayMea
5 b_changeLaneMea
6 calc/calLeftPredictedPoly
7 calc/calRightPredictedPoly
8 z_u16_vehLgtSpdDelay
9 s16_leftLineDistPred
10 s16_leftLineHAPred
11 s16_leftLineCrvtPred
12 s16_leftLineCrvtRatePred
13 s16_rightLineDistPred
14 s16_rightLineHAPred
15 s16_rightLineCrvtPred
16 s16_rightLineCrvtRatePred
17 b_leftOutOfArea

Working with Sequencing view

The sequence view shows the order of assignments for a method generated by Blo

Opening the sequence view

The sequence view is opened by default in the Diagrams perspective.

If you close it, then it can be added to the perspective manually.

Click on Window › Show View › Other › ASCET › Sequencing

Reorder sequence calls in the sequencing view

The sequence view displays all sequence-able elements.

- You can reorder sequence calls in the sequencing view by following ways:
- By using drag and drop:
- Select the sequence calls to move.
- Drag and drop selected to new place.

Automatically create test harness

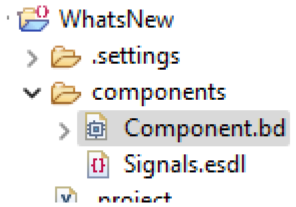
PC Simulation and Unit Testing

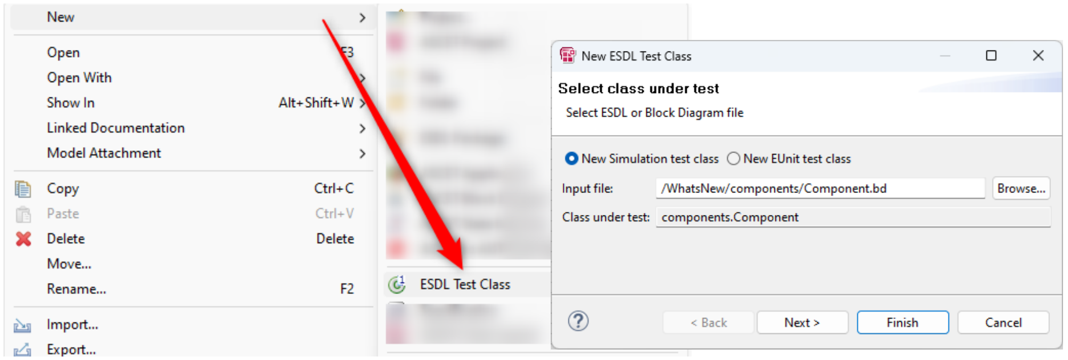
Pain Point

- ☹ Too many manual steps to test a class
- ☹ PC Simulation
  - ☹ Create wrapper Class
  - ☹ Create app and scheduling
  - ☹ Create stimuli
- ☹ Unit Test
  - ☹ Create Test Class
  - ☹ Manually create test methods

Benefit

- ☺ All test artefacts created quickly
- ☺ Start your test immediate and save time for boring work





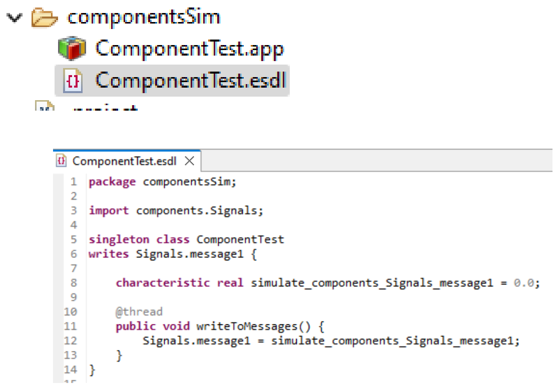
Select class under test

Select ESDL or Block Diagram file

[● New Simulation test class] ○ New EUnit test class

Input file: /WhatsNew/components/Component.bd

Class under test: components.Component



ComponentTest.esdl

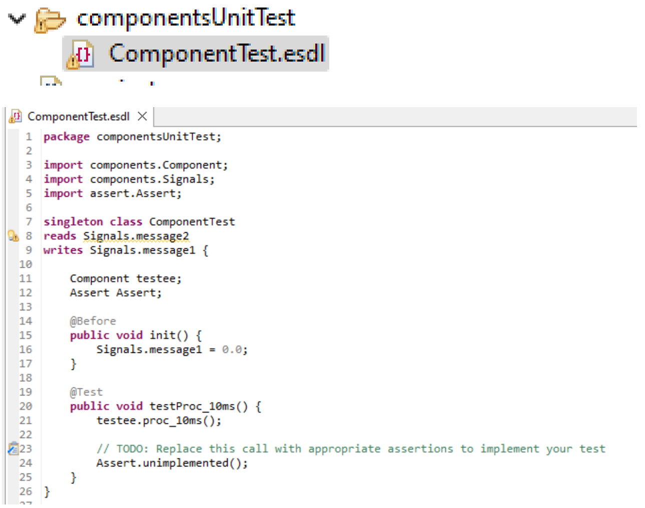
```
package componentsSim;

import components.Signals;

singleton class ComponentTest
writes Signals.message1 {

    characteristic real simulate_components_Signals_message1 = 0.0;

    @thread
    public void writeToMessages() {
        Signals.message1 = simulate_components_Signals_message1;
    }
}
```



ComponentTest.esdl

```
package componentsUnitTest;

import components.Component;
import components.Signals;
import assert.Assert;

singleton class ComponentTest
reads Signals.message2
writes Signals.message1 {

    Component testee;
    Assert Assert;

    @Before
    public void init() {
        Signals.message1 = 0.0;
    }

    @Test
    public void testProc_10ms() {
        testee.proc_10ms();

        // TODO: Replace this call with appropriate assertions to implement your test
        Assert.unimplemented();
    }
}
```

Unit Tests

Physical code generation

Pain Point

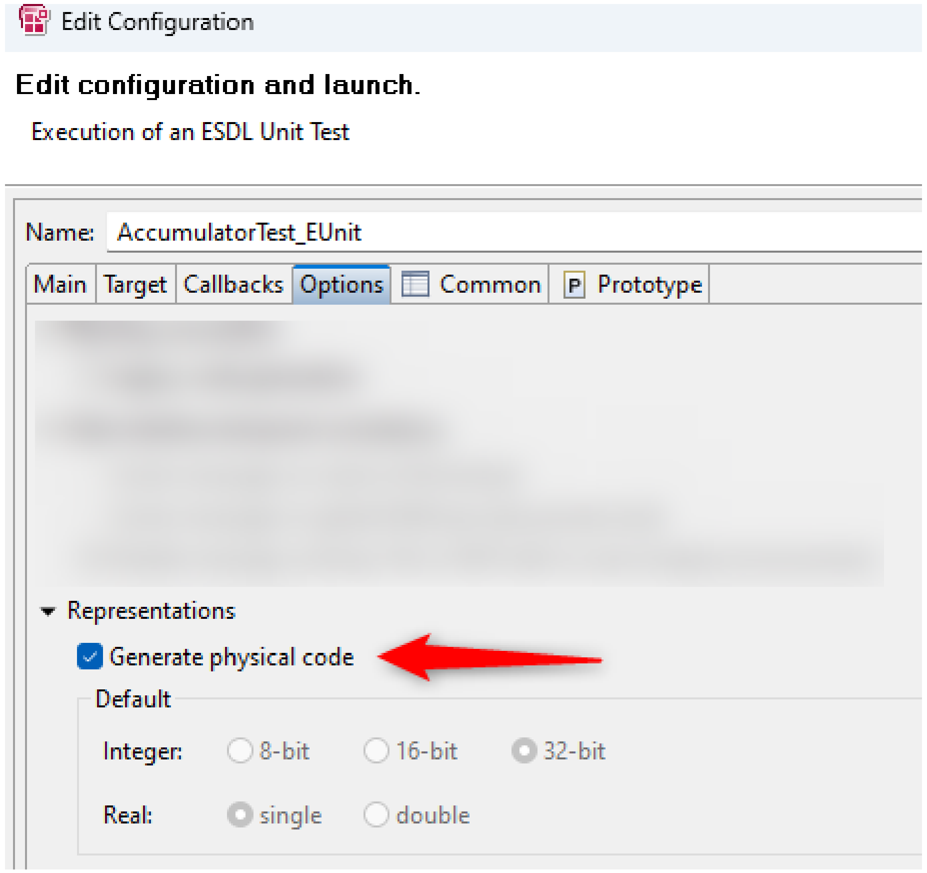
☹ Unable to unit test logic independent of target representation

☹ With failing unit test not clear if logic or representation was reason

Benefit

☺ Bringing MiL/SiL like use case also to unit testing

☺ Run back-to-back test for every change



Edit Configuration

Edit configuration and launch.

Execution of an ESDL Unit Test

Name: AccumulatorTest_EUnit

Representations

[✓] Generate physical code

Default

Integer: 8-bit 16-bit [✓ 32-bit]

Real: [✓ single] double

Group as statement block

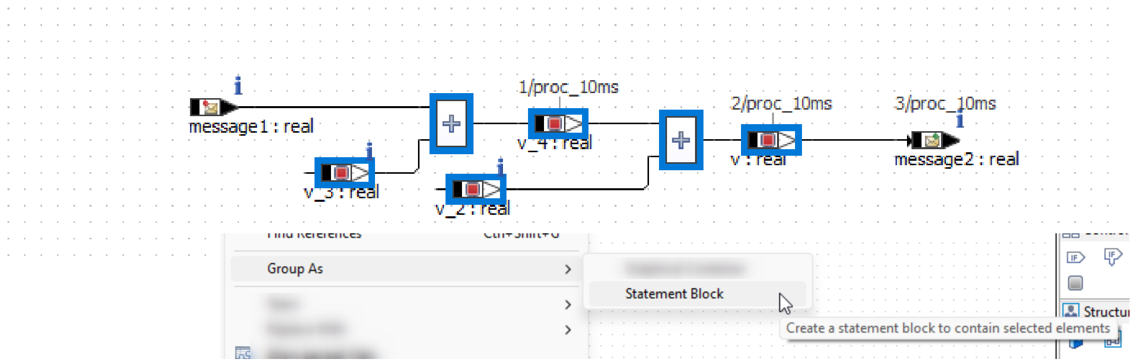
Improve refactoring

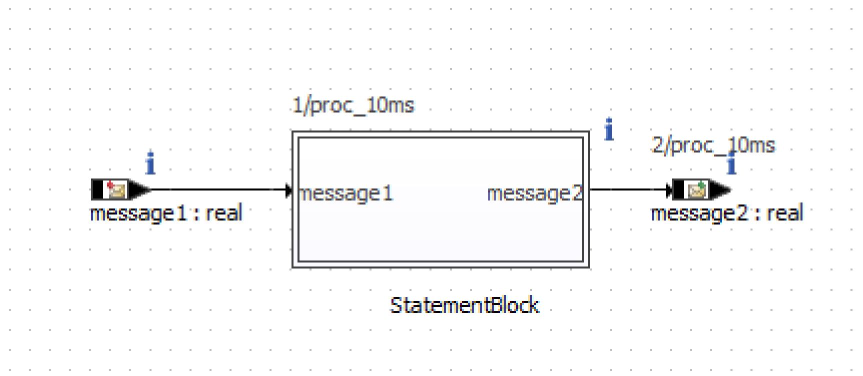
Pain Point

☹ Quickly refactoring part of diagram as statement block not possible

Benefit

☺ Refactor complex diagrams and manage control and data flow using statement block





Integrate external code

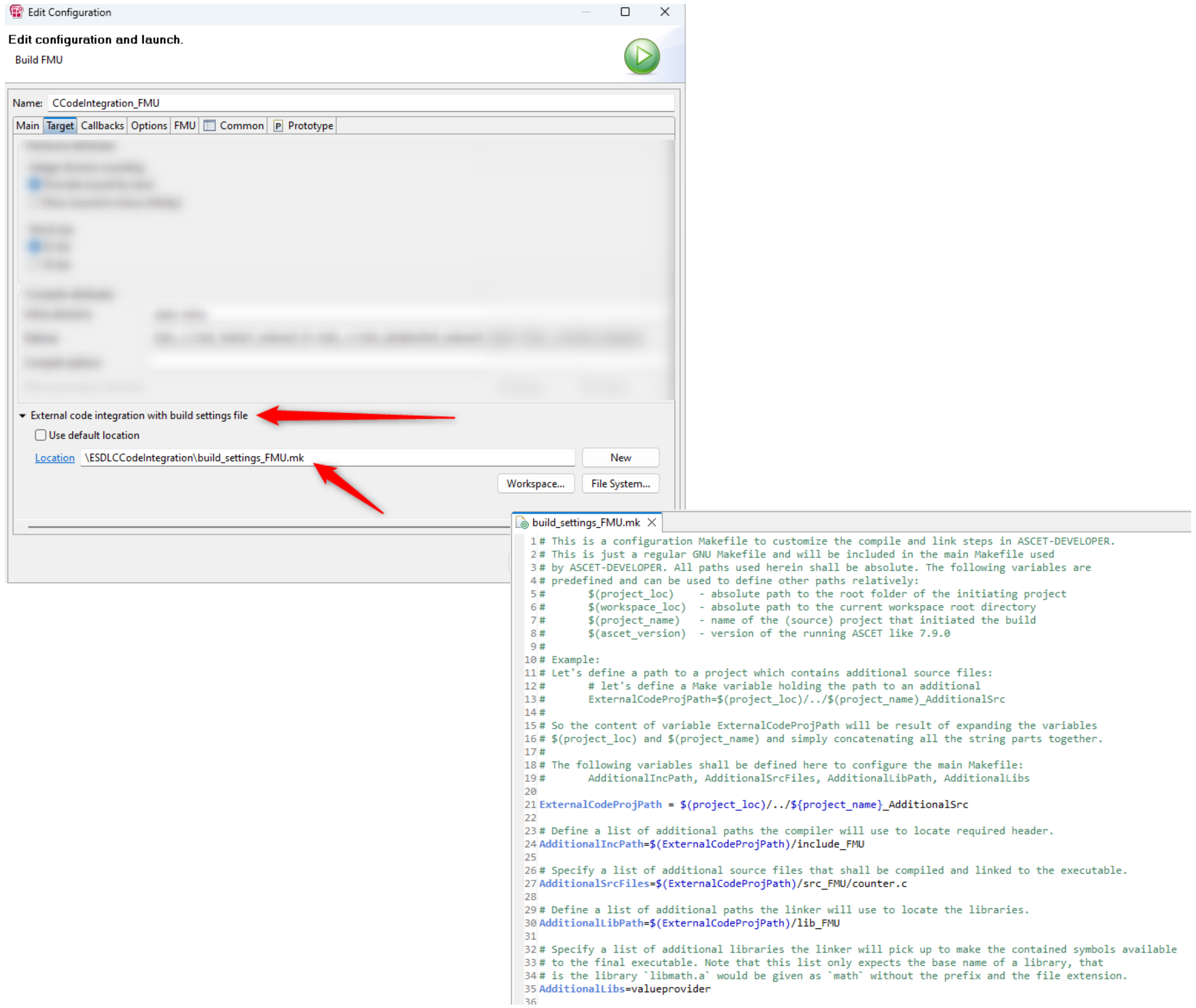
PC, EUnit, FMU Build

Pain Point

☹ Unable to include external code used by ASCET Model for the build

Benefit

☺ Easy integration and behavior dependent on external code can also be verified



Name: CCodeIntegration_FMU

External code integration with build settings file

☐ Use default location

Location \ESDLCCodeIntegration\build_settings_FMU.mk

build_settings_FMU.mk

```
# This is a configuration Makefile to customize the compile and link steps in ASCET-DEVELOPER.
# This is just a regular GNU Makefile and will be included in the main Makefile used
# by ASCET-DEVELOPER. All paths used herein shall be absolute. The following variables are
# predefined and can be used to define other paths relatively:
#       $(project_loc)    - absolute path to the root folder of the initiating project
#       $(workspace_loc)  - absolute path to the current workspace root directory
#       $(project_name)   - name of the (source) project that initiated the build
#       $(ascet_version)  - version of the running ASCET like 7.9.0
#
# Example:
# Let's define a path to a project which contains additional source files:
#       # let's define a Make variable holding the path to an additional
#       ExternalCodeProjPath=$(project_loc)/../$(project_name)_AdditionalSrc
#
# So the content of variable ExternalCodeProjPath will be result of expanding the variables
# $(project_loc) and $(project_name) and simply concatenating all the string parts together.
#
# The following variables shall be defined here to configure the main Makefile:
#       AdditionalIncPath, AdditionalSrcFiles, AdditionalLibPath, AdditionalLibs

ExternalCodeProjPath = $(project_loc)/../${project_name}_AdditionalSrc

# Define a list of additional paths the compiler will use to locate required header.
AdditionalIncPath=$(ExternalCodeProjPath)/include_FMU

# Specify a list of additional source files that shall be compiled and linked to the executable.
AdditionalSrcFiles=$(ExternalCodeProjPath)/src_FMU/counter.c

# Define a list of additional paths the linker will use to locate the libraries.
AdditionalLibPath=$(ExternalCodeProjPath)/lib_FMU

# Specify a list of additional libraries the linker will pick up to make the contained symbols available
# to the final executable. Note that this list only expects the base name of a library, that
# is the library `libmath.a` would be given as `math` without the prefix and the file extension.
AdditionalLibs=valueprovider
```

Sequence number commenting

Disable code

Pain Point

☹ No possibility in block diagram editor to disable a sequence number execution

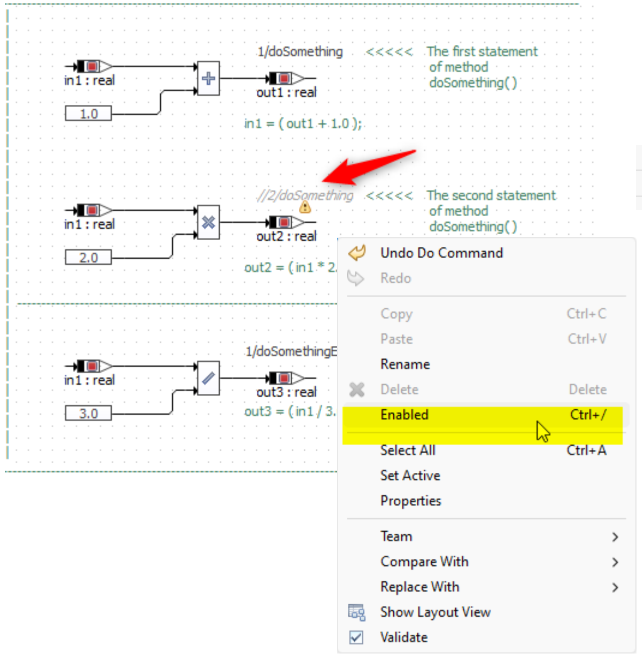
☹ In textual modeling I can comment out lines-of-code and later bring it back

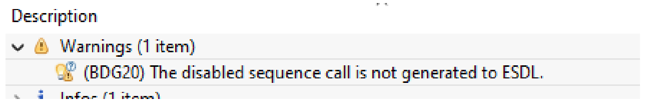
Benefit

☺ Easy way to disable sequence number and thereby its execution

☺ As natural as commenting out lines of code

☺ No need to delete BD parts as workaround





Description

Warnings (1 item)

(BDG20) The disabled sequence call is not generated to ESDL.

Improved properties view in state machine editor

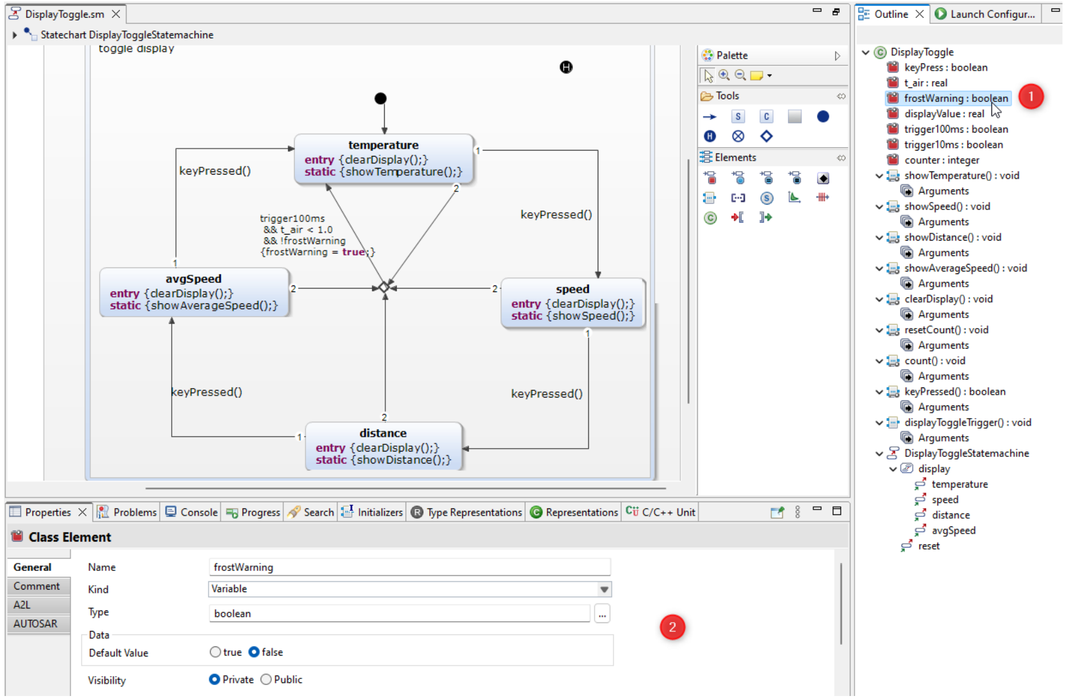
Faster operations

Pain Point

☹ To edit state machine class element properties always switch to ESDL editor

Benefit

☺ Stay in context and faster operations



- DisplayToggle
  - keyPress : boolean
  - t_air : real
  - frostWarning : boolean
  - displayValue : real
  - trigger100ms : boolean
  - trigger10ms : boolean
  - counter : integer
  - showTemperature() : void
    - Arguments
  - showSpeed() : void
    - Arguments
  - showDistance() : void
    - Arguments
  - showAverageSpeed() : void
    - Arguments
  - clearDisplay() : void
    - Arguments
  - resetCount() : void
    - Arguments
  - count() : void
    - Arguments
  - keyPressed() : boolean
    - Arguments
  - displayToggleTrigger() : void
    - Arguments
  - DisplayToggleStatemachine
    - display
      - temperature
      - speed
      - distance
      - avgSpeed
    - reset

Class Element

General | Comment | A2L | AUTOSAR

Name	frostWarning
Kind	Variable
Type	boolean
Data	
Default Value	true [✓ false]
Visibility	[✓ Private] Public

Unit conversion block

Block Diagram Editor

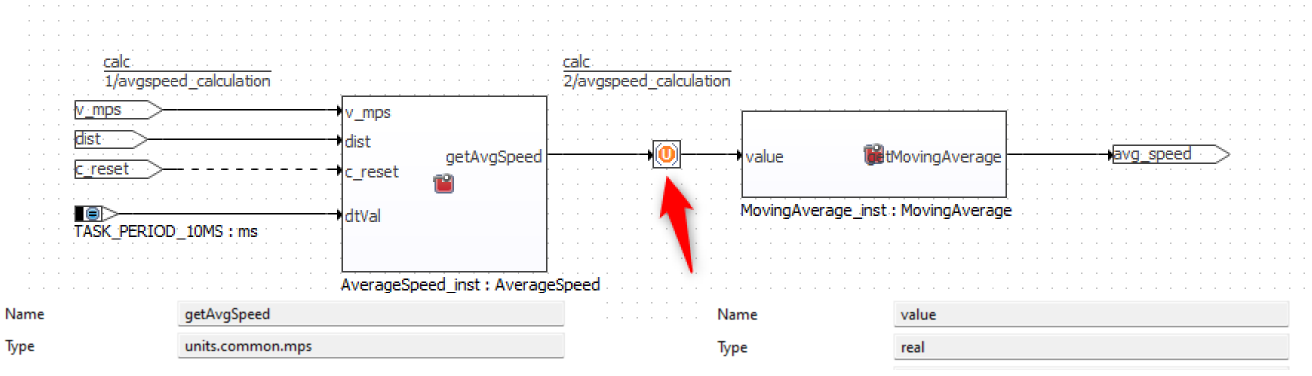
Pain Point

☹ Extra modeling effort for type conversion between unit less and unit type element

Benefit

☺ Improved user experience

☺ Cleaner block diagram



Name	getAvgSpeed
Type	units.common.mps

Name	value
Type	real

```
 * Meter per second, measuring speed or velocity.
 */
unit mps = m/s;

```

AUTOSAR

New features and other improvements

Pain Point

☹ Cannot configure for individual mapped ASCET element which RTE access type to use

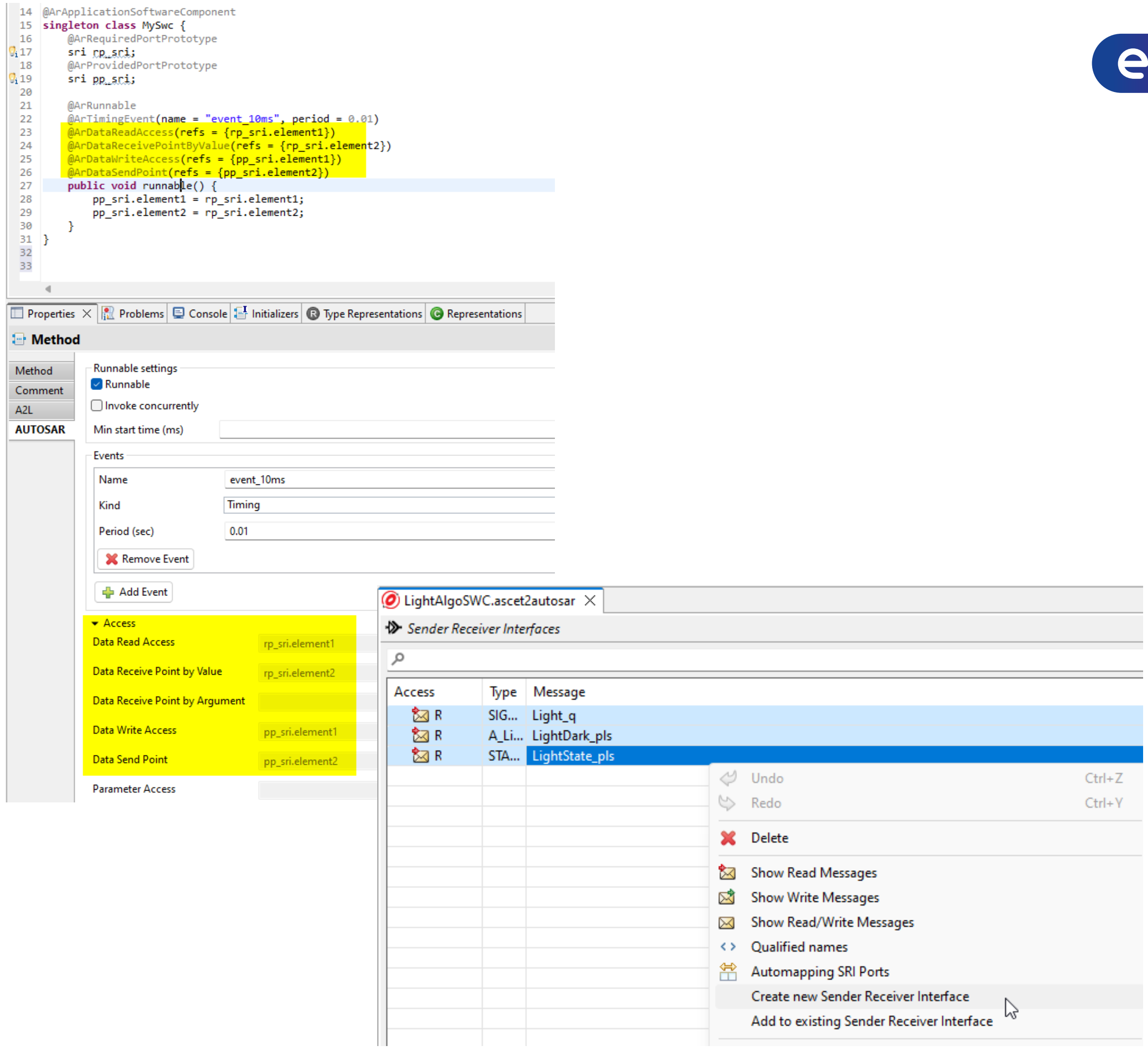
☹ Too many manual operations to create ASCET AUTOSAR structure

Benefit

☺ Customizable and easy to configure element wise access

☺ Select ESDL Messages in mapping and create AR interfaces easily

☺ Generate complete AR structure as first starting point from app



```
@ArApplicationSoftwareComponent
singleton class MySwc {
    @ArRequiredPortPrototype
    sri rp_sri;
    @ArProvidedPortPrototype
    sri pp_sri;

    @ArRunnable
    @ArTimingEvent(name = "event_10ms", period = 0.01)
    @ArDataReadAccess(refs = {rp_sri.element1})
    @ArDataReceivePointByValue(refs = {rp_sri.element2})
    @ArDataWriteAccess(refs = {pp_sri.element1})
    @ArDataSendPoint(refs = {pp_sri.element2})
    public void runnable() {
        pp_sri.element1 = rp_sri.element1;
        pp_sri.element2 = rp_sri.element2;
    }
}
```

Runnable settings
- [✓] Runnable
- [ ] Invoke concurrently

Min start time (ms):

Events

Name	event_10ms
Kind	Timing
Period (sec)	0.01

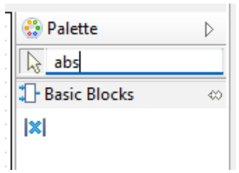
Access

Data Read Access	rp_sri.element1
Data Receive Point by Value	rp_sri.element2
Data Receive Point by Argument	
Data Write Access	pp_sri.element1
Data Send Point	pp_sri.element2
Parameter Access	

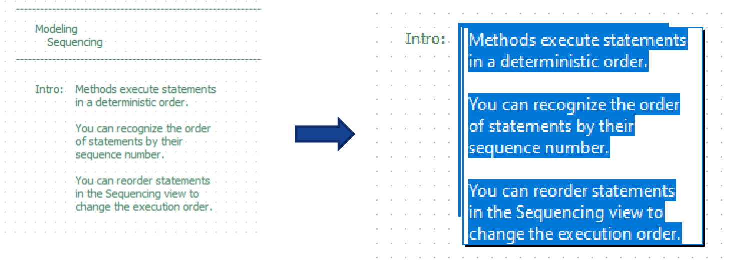
Access	Type	Message
R	SIG...	Light_q
R	A_Li...	LightDark_pls
R	STA...	LightState_pls

Small things

– Search in BD Editor Palette
  ☺*Find block faster*



– Edit graphical comment directly in BD Editor canvas
  ☺*Faster operation avoid shift focus to properties view*


Modeling
Sequencing

Intro: Methods execute statements in a deterministic order.

You can recognize the order of statements by their sequence number.

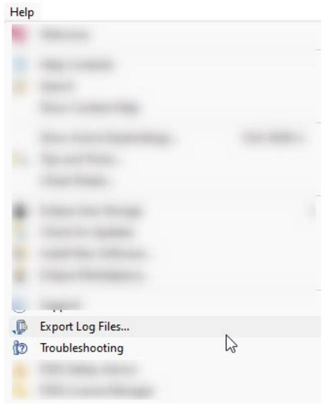
You can reorder statements in the Sequencing view to change the execution order.

Intro: Methods execute statements in a deterministic order.

You can recognize the order of statements by their sequence number.

You can reorder statements in the Sequencing view to change the execution order.

– Problem Report Export from Help Menu
  ☺*Provide debug relevant info to etas support easier*
  ☺*Direct link to troubleshooting chapter*



Thank you