

Association for standardisation of automation and measuring systems



# **Release Presentation**

### ASAM AE XIL 2.1

**Generic Simulator Interface** 

2017-06-26



### Agenda

- Introduction and General Concepts
- What's New?
- Deliverables
- Compatibility
- History and Outlook



# **Motivation of the Standard**



Separation of Test HW and Test SW by means of standardized APIs



Association for standardisation of automation and measuring systems

## **Concept of Ports**

#### MAPort Model Access port provides access to the simulation model read and write parameters, capture and generate signals.

The ECUM port allows capturing

variables. The ECUC port is used

and reading of measurement

#### **NetworkPort**

provides access to field bus systems such as CAN. E. g. Allows measurement (monitoring) and transmission (single transmit or replay) of bus data.

#### **EESPort**

Electrical Error Simulation port controls electrical error simulation hardware. It allows the setup of different types of errors (e. g. short cuts).

#### DiagPort

Diagnostic port communicates with a diagnostic system, reads data via diagnostic services from an ECU.



**ECUPort** 

for calibration.

Association for standardisation of automation and measuring systems

## Port-based Access of a Test Automation Tool



www.asam.net



## **Framework-based Access to Testbench**

#### **Major Benefits:**

- · Port independence of testcases by using an object-oriented access to variables
- · Framework starts and shuts down ports in a configured order
- Test Developer can use both: Testbench Port access and Variable-based access
- · Framework Variables provide access to the underlying Testbench Port



www.asam.net



Association for standardisation of automation and measuring systems

## **More Details**

https://wiki.asam.net/display/STANDARDS/ASAM+XIL



## What's New?

- Pause Simulation
- Support of Real-Time Scripts
- Parameterized SignalDescriptions
- Relation between Simulation and Capturing / Stimulation
- Capture Client Events
- Read/Write Simultaneously
- Check Variable Names
- Download Parameter Sets
- New Version of General Expression Syntax (GES) 1.0.1
- Relation to FMI: Support of the variability tunable, fixed and const
- DiagPort





eSIMULATION\_PAUSED state can be entered by either a method (PauseSimulation) or a **MAPortBreakpoint** 

www.asam.net



# Support of Real-Time Scripts

Users can administrate scrips (e. g. for real-time based tests) analogue to the well known SignalGenerator for stimulation, such as LoadToTarget, Start, Stop.



+

setSignalDescriptionSet(SignalDescriptionSet) :void



## **Parameterized SignalDescriptions**

SignalDescriptions support expressions with placeholders to be assigned to SignalSegment parameters.

#### **Benefits:**

This significantly increases reusability of SignalDescriptions.

With expressions and placeholders properly applied a client does not need to know about the segment structure and the interrelations when using and adapting a predefined SignalDescription.





### **Relation between Simulation and Capturing / Stimulation**



The synchronous behavior of start/stop and pause a simulation in relation to multiple captures and to stimulations has been defined precisely. For example, if a simulation pauses at the MAPort, then the simulation time freezes and so does the time in the corresponding Capture and SignalGenerator objects.

Association for standardisation of automation and measuring systems

### **Capture Client Events**

Users can add events at the Capture object, e. g. to bookmark or label interesting situations (e. g. due to electrical error simulation or feedback of the diagnostic system).

The CaptureResult object provides a getter getEvents() to access the captured events.

	Capture
+ + + + +	ClearConfiguration() :void Fetch(whenFinished :A_BOOLEAN) :CaptureResult SetStartTrigger(watcher :Watcher, delay :Duration) :void SetStopTrigger(watcher :Watcher, delay :Duration) :void Start(writer :CaptureResultWriter) :void Stop() :void TriggerClientEvent(eventid :A_UINT64, description :A_UNICODE2STRING) :void
"	
* + + + + + + + + + + + + + + + + + + +	getCaptureResult() :CaptureResult getDownsampling() :A_UINT64 getMinBufferSize() :A_INT64 getPort() :Port getRetriggering() :A_INT64 getStartTriggerCount() :A_INT64 getStartTriggerDelay() :Duration getStartTriggerWatcher() :Watcher getState() :CaptureState getStopTriggerDelay() :Duration getStopTriggerDelay() :Duration getStopTriggerWatcher() :Watcher getVariables() :A_UNICODE2STRING[]
«g	etter, deprecated»
+	
«Si + + + +	etter» setDownsampling(downSampling:A_UINT64) :void setMinBufferSize(minBufferSize:A_INT64) :void setRetriggering(Retriggering:A_INT64) :void setVariables(variableNames:A_UNICODE2STRING[)) :void
«Se	etter, deprecated»
+	setDurationUnit(durationUnit :EDurationUnit) :void
«d	eprecated»
+ +	SetStartTriggerCondition(triggerDefinition :Watcher, delay :A_FLOAT64) :void SetStopTriggerCondition(triggerDefinition :Watcher, delay :A_FLOAT64) :void

CaptureResult				
<ul> <li>ExtractSignalValue(signalGroupName :A_UNICODE2STRING, variableName :A_UNICODE2STRING) :SignalValue</li> <li>GetSignalGroupValue(signalGroupName :A_UNICODE2STRING) :SignalGroupValue</li> <li>GetVariableNames(signalGroupName :A_UNICODE2STRING) :A_UNICODE2STRING[]</li> <li>Open(reader : CaptureResultReader) :void</li> <li>Save(writer : CaptureResultWriter) :void</li> </ul>				
<pre>«getter» + getCaptureStartTime():A_EL_OAT64 + getEvents():CaptureEvent[] + getM etaData():StringNamedCollection + getSignalGroupNames():A_UNICODE2STRING[]</pre>				
«ætter» + setM etaData(metaData :StringNamedCollection) :void				



### **Read/Write Simultaneously**

Ensure that reading and writing of multiple model variables takes place in the same simulation step or in the same cycle of a given task.

MAPort				
+	CheckVariableNames(variableNames:A_UNICODE2STRING[]):A_UNICODE2STRING[]			
+	Configure(config:MAPortConfig, forceConfig:A BOOLEAN) :void			
+	CreateCapture(taskName :A_UNICODE2STRING) :Capture			
+	CreateSignalGenerator() :SignalGenerator			
+	CreateTargetScript():TargetScript			
+	DownloadParameterSets(filepaths:A_UNICODE2STRING[]):void			
+	GetDataType(variableName:A_UNICODE2STRING):DataType			
+	GetVariableInfo(variableName :A_UNICODE2STRING) :MAPortVariableInfo			
+	IsReadable(variableName : A_UNICODE2STRING) :A_BOOLEAN			
+	IsWritable(variableName : A UNICODE2STRING) : A BOOLEAN			
+	LoadConfiguration(filepath:A_UNICODE2STRING):MAPortConfig			
+	PauseSimulation():void			
+	Read(variableName A_UNICODE2STRING) :BaseValue			
+	ReadSimultaneously(variableNames:A_UNICODE2STRING[], taskName:A_UNICODE2STRING):BaseValue[]			
+ '	StartSimulation() :void			
+	StopSimulation() :void			
+	WaitForBreakpoint(timeout :A FLOAT64) :void			
+	Write(variableName : A_UNICODE2STRING, value :BaseValue) :void			
+	WriteSimultaneously(variableNames: A UNICODE2STRING[], values:BaseValue[], taskName: A UNICODE2STRING):void			
«a	etter»			
+	getBreakpoint():MAPortBreakpoint			
+	aetConfiguration():MAPortConfig			
+	getDAQClock) :A FLOAT64			
+	aetSimulationStepSize() :A FLOAT64			
a defSimultaneous evel () Simultaneous evel				
+	detState() MAPortState			
+	detTaskInfos() TaskInfo[]			
+	detTasklames) : A UNICODE2STRINGI			
+	getVariableNames) : A UNICODE2STRINGI			
«setter»				
+	selbreakpoint(breakpoint .mAPontBreakpoint) :void			



### **Check Variable Names**

Allows the user to check, whether variable names exist or not. The method checks if the given variable names are valid variable names with the current port configuration. Invalid names are returned. If all names are valid the returned list is empty.

MAPort					
F	Check/ariableNames(variableNames:A_UNICODE2STRING[]):A_UNICODE2STRING[]				
ł	Configure(config :MAPortConfig, forceConfig :A_BOOLEAN) :void				
ł	CreateCapture(taskName :A_UNICODE2STRING) :Capture				
ł	CreateSignalGenerator():SignalGenerator				
+	CreateTargetScript() :TargetScript				
+	DownloadParameterSets(filepaths:A_UNICODE2STRING[]) :void				
ł	GetDataType(variableName :A_UNICODE2STRING) :DataType				
ł	GetVariableInfo(variableName :A_UNICODE2STRING) :MAPortVariableInfo				
ł	IsReadable(variableName :A_UNICODE2STRING) :A_BOOLEAN				
ł	IsWritable(variableName :A_UNICODE2STRING) :A_BOOLEAN				
ł	LoadConfiguration(filepath :A_UNICODE2STRING) :MAPortConfig				
F	PauseSimulation() :void				
+ Read(variableName:A_UNICODE2STRING):BaseValue					
ł	ReadSimultaneously(variableNames:A_UNICODE2STRING[], taskName:A_UNICODE2STRING):BaseValue[]				
+ StartSimulation() :void					
+	stopsimulation() word				
+	Water water and the second sec				
	WiteSimilaneeud/waideblohames 4. UNICODE2STRING, value basevalue), void				
«g	elle»				
+	getbeakpoint() .MAP orbitakpoint				
•	getConfiguration() .mAppliconing				
-	getDAQCIOCK) .A_FLOAT64				
•	getSimulationStepSiZe() .A_TELOAT64				
-	getomutaneous.every.comutaneous.ever				
	get askinosytakinoj				
+	get/agiteleName() -A_UNICODE25TRING[]				
«setter»					
t	setereakpoint(breakpoint .MAPORBReakpoint) :void				



### **Download Parameter Sets**

Loads the specified parameter set files and writes the contained parameter values to the corresponding variables of the simulation tool or hardware

	MAPort			
+	CheckVariableNames(variableNames: A UNICODE2STRING[]) : A UNICODE2STRING[]			
+	Configure(config:MAPortConfig; forceConfig:A BOOLEAN) :void			
۲	CreateCapture(taskName : A UNICODE2STRING) :Capture			
+	CreateSignalGenerator():SignalGenerator			
ł	CreateTargetScript():TargetScript			
ł	DownloadParameterSets(filepaths:A_UNICODE2STRING[]):void			
F	GetDataType(variableName:A_UNICODE2STRING):DataType			
F	GetVariableInfo(variableName :A_UNICODE2STRING) :MAPortVariableInfo			
F	IsReadable(variableName : A_UNICODE2STRING) : A_BOOLEAN			
F	IsWritable(variableName : A_UNICODE2STRING) : A_BOOLEAN			
ŀ	LoadConfiguration(filepath :A_UNICODE2STRING) :MAPortConfig			
F	PauseSimulation() :void			
F	Read(variableName :A_UNICODE2STRING) :BaseValue			
ReadSimultaneously(variableNames:A_UNICODE2STRING[], taskName:A_UNICODE2STRING) :BaseValue[]				
F	StartSimulation() :void			
F	StopSimulation() :void			
F	WaitForBreakpoint(timeout :A_FLOAT64) :void			
+	Write(variableName :A_UNICODE2STRING, value :BaseValue) :void			
ŀ	WriteSimultaneously(variableNames:A_UNICODE2STRING[], values:BaseValue[], taskName:A_UNICODE2STRING):voi			
«g	etter»			
F	getBreakpoint() :MAPortBreakpoint			
ł	getConfiguration() :MAPortConfig			
-	getDAQClock() :A_FLOAT64			
-	getSimulationStepSize() :A_FLOAT64			
ł	getSimultaneousLevel() :SimultaneousLevel			
	getState() :MAPortState			
F	get Laskintos() : Laskinto[]			
F	getTaskNames() :A_UNICODE2STRING[]			
F	getVanableNames() :A_UNICODE2STRING[]			
«S	etter»			
+	setBreakpoint(breakPoint:MAPortBreakpoint):void			



### New Version of General Expression Syntax (GES) 1.0.1

New additional methods to detect value changes

Support of a new constant

	Semantic	Syntax and Arguments			
Neth	Detection of a positive value change: returns true when the value of <i>expr1</i> is increased with respect to the previous evaluation step and the increase is greater than or equal to <i>expr2De1ta</i>	changedpos(expr1, expr2Delta)			
	Detection of a negative value change: returns true when the value of <i>expr1</i> is decreased with respect to the previous evaluation step and the decrease is greater than or equal to <i>expr2De1ta</i>	changedneg(expr1, expr2Delta)			

₹<sup>37</sup> INF

Infinity, to configure the borders of the data file segment within stimulations.

ASAM Association f automation a

Association for standardisation of automation and measuring systems

### Relation to FMI: Support of the variability tunable, fixed and const

The dependency of the writeability on simulation state is useful for the following reasons:

- Some variables influence a model only during initialization, e. g. initial values for discrete or continuous state variables.
- Variables can be used for computation of other variables, e. g. the vehicle's total mass m as the sum of the vehicle mass  $m_v$  and the additional load  $m_i$ . Thus, m needs to be calculated only once during initialization, and not again and again in every simulation step.

XIL already defines a MAPortVariableInfo object to check if a variable is readable or is writeable.	is in the second	<ul> <li>MAPortVariableInfo</li> <li>«getter» getAvailableTasks()</li> <li>IsReadable(MAPortState)</li> <li>IsWriteable(MAPortState)</li> </ul>				
The interpretation of the IsReadable/IsWriteable method results has been defined as described in the table according to the variability of <u>FMI</u> .	Hoh	Variability according to FMI	edisconnected	esimulation_stopped	esimulation_paused	esimulation_running
		constant	False	False	False	False
		fixed	False	True	False	False
		tunable	False	True	True	True



### **DiagPort (1)**

DiagPort classes have been re-designed to bring the DiagPort into operation.



### **DiagPort (2)**

Macro classes have been removed due to missing relevance.

XIL 2.0.2



### XIL 2.1





# Compatibility

The API elements listed in the second column of the table are deprecated and might be removed in a future version of the standard.

So it is recommended to use the replacement depicted in the third column.

Note: Changes and extensions in DiagPort are not compatible to XIL 2.0.2.

Affected Component	Deprecated Element	Replacement			
Capture	Method SetStartTriggerCondition	Method SetStartTrigger			
(Package Testhonsh Common Conturi	Method SetStopTriggerCondition	Method SetStopTrigger			
ng)	Property <b>DurationUnit</b>	Implicitly set by passing a <b>TimeSpanDuration</b> or <b>CycleNumberDuration</b> object (from package Testbench.Common.Duration) to SetStartTrigger, SetStopTrigger and the DurationWatcher factory methods.			
ConstSymbol (Testbench.Common.Symb ol)	Property Value	Property Expression			
DurationUnit (Testbench.Common.Captu ring.Enum)	Enumeration <b>DurationUnit</b>	Implicit representation by TimeSpanDuration and CycleNumberDuration in package Testbench.Common.Duration			
<b>DurationWatcher</b> (Package Testbench.Common.Watch erHandling)	Property <b>Duration</b>	Property <b>Duration2</b> . The property's value cannot be changed. Create a new DurationWatcher for a different duration value.			
SignalGenerator (Package Testbench.Common.Signal Generator)	Property <b>State</b>	Property <b>ScriptState</b> inherited from the base interface Script in package Testbench.Common (SignalGenerator is derived from Script)			
<b>SignalGeneratorState</b> (Package Testbench.Common.Signal Generator.Enum)	Enumeration SignalGeneratorState	Enumeration <b>ScriptState</b> in package Testbench.Common.Script.Enum			
SymbolFactory (Testbench.Common.Symb ol)	Method CreateConstSymbolByValue	Method CreateConstSymbolByExpression			
WatcherFactory (Package Testbench.Common.Watch erHandling)	Method CreateDurationWatcher	Methods CreateDurationWatcherByTimeSpa n und CreateDurationWatcherByCycleNu mber			



### **Deliverables**

- Package Standard
- Package XilSupportLibrary (framework software parts)
- Package Example Framework
- Package Prototype and test environment
- Package MSI Setup

## **History & Outlook**

2009 (July) **2010 (December): 2012 (January)** 2012 (June) 2013 (February) **2013 (October) 2014 (October) 2015 (October)** 2016 (July) 2017 (June) **2017 (October)** 2018 2019

**HIL API 1.0.0 HIL API 1.0.1 HIL API 1.0.2 Crosstest No. 1** (among 4 vendors) (among 5 vendors) **Crosstest No. 2 XIL API 2.0.0 XIL API 2.0.1 XIL API 2.0.2 Crosstest No. 3** (among 7 vendors) XII API 2.1 **Crosstest No. 4** (Planned) XIL API 2.1.1 (Planned) **XIL API 2.1.2** (Planned)