eveCSS

Marcus Michalsky

eveCSS

Marcus Michalsky

Table of Contents

Foreword

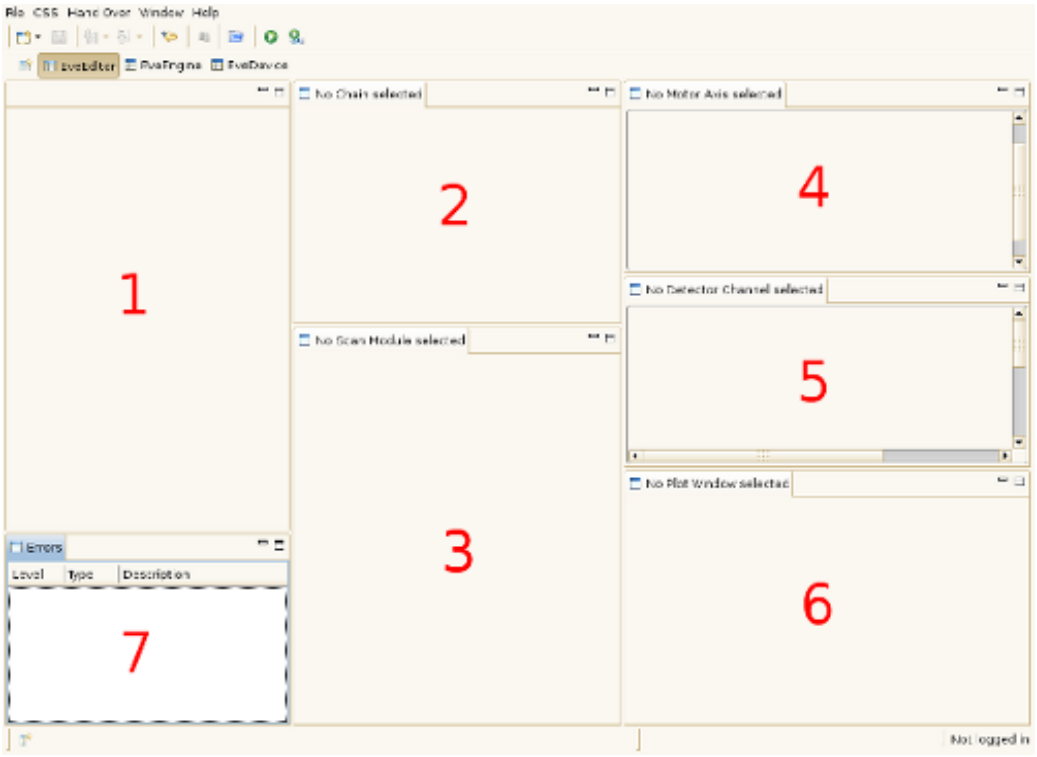
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Part I. The EVE-Editor

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Chapter 1. Getting Started

The image below shows an overview of the EVE Editor.



1	Editor Area	graphical representation of scans
2	Chain View	chain properties
3	Scan Module View	scan module properties
4	Motor Axis View	motor axis properties
5	Detector Channel View	detector channel properties
6	Plot Window View	plot window properties
7	Error View	message log

The first thing you should do is create a new scan description [tasks/createscandescription.html] .

Chapter 2. Concepts

Events

Events are a way to affect scans, e.g. by pausing it if a condition is met. Three types (describing their source) of events exist:

- **SCHEDULE**: a schedule event is triggered when a scan module is finished.
- **DETECTOR**: a detector event is triggered when a detector is ready.
- **MONITOR**: a monitor event sends `true` or `false` whether a condition is met.

The effect (or outcome) of an event is defined by the location the event is added to. Events could be added to a *chain*, a *scan module* or a *detector channel* in their respective views [../gettingstarted.html]. Each view has one or more tabs, one for each outcome:

View	Pause	Redo	Break	Stop	Trigger
Chain	+	+	+	+	
Scan Module	+	+	+		+
Detector Channel					+

Monitor conditions are evaluated at the start of a chain/scan module.

Outcomes

The effect an event has depends on its outcome which is one of *pause*, *redo*, *break*, *stop*, *trigger*.

Pause Event

The scan has a state which is either *running* or *pause*. A detector or schedule event (trigger events) can manipulate that state by its action attribute as follows:

Row	Scan State (current)	Event Action	Scan State (result)	
1	*running*	*ON*	*pause*	*
2	*running*	*OFF*	*running*	
3	*pause*	*ON*	*pause*	
4	*pause*	*OFF*	*running*	*

Note: Scenarios shown in row two and three have no effect at all since the scan state does not change.

Since monitor events do not just trigger when they occur but signal whether a specified condition is met there are more possible scenarios:

Row	Scan State (current)	Event Action	condition	Scan State (result)	
1	*running*	*ON*	*true*	*pause*	*

2	*running*	*ON*	*false*	*running*	
3	*running*	*OFF*	*true*	*running*	
4	*running*	*OFF*	*false*	*running*	
5	*running*	*ONOFF*	*true*	*pause*	*
6	*running*	*ONOFF*	*false*	*running*	
7	*pause*	*ON*	*true*	*pause*	
8	*pause*	*ON*	*false*	*pause*	
9	*pause*	*OFF*	*true*	*running*	*
10	*pause*	*OFF*	*false*	*pause*	
11	*pause*	*ONOFF*	*true*	*running*	*
12	*pause*	*ONOFF*	*false*	*pause*	

If the specified condition is not met nothing happens at all (even rows). Rows three and seven just overwrites the state with the same value. The behavior in rows one and nine is equal to the behavior of detector and schedule events. Rows five and eleven show the behavior specific to monitor events due to the additional *ONOFF* action.

Note: Each event can continue the scan (stop the pause) regardless of which event paused it (learn more on combining events).

Redo Event

If caused by trigger events (detector or schedule) the detector is read again or the scan module is executed again respectively. Monitor events behave the same way (redo if condition is `true`)except that they will pause the scan before reading/executing again until the event sends `false`.

Break/Stop/Trigger Event

Caused by a trigger event (detector or schedule) and monitor event sending `true`.

Combining *Chain* and *ScanModule* Events

Chains and scan modules have *independent* pause states, i.e. both scan module and chain states must be *running* in order to continue a scan.

When monitor events are combined (one in the chain, one in the scan module) a redo occurs if at least one of their last send values was `true`. Only when both last sent `false` the scan continues normally.

Chapter 3. Reference

Chain Properties

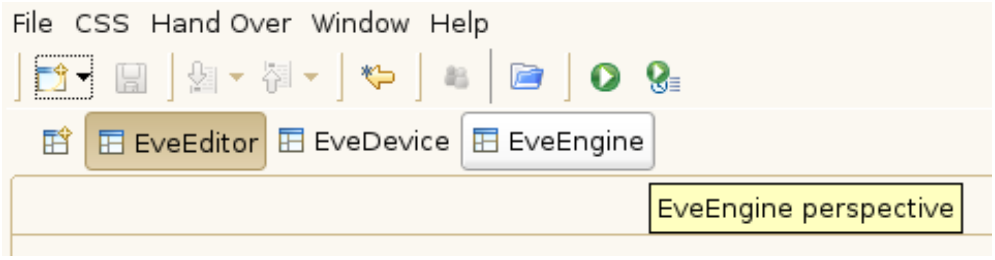
Property	Description	Domain
File Format	format of the data file	save plug ins defined in the test site XML
File Name	name of the data file	string
Save Scan-Description	Indicates whether the scan description should be saved (same location as the data file)	{true, false}
Confirm Save	Indicates whether saving should be confirmed	{true, false}
Add Autoincrementing Number to Filename	if *true* a four digit number is appended to the file name (starting at *x + 1* where *x* is the largest number found with the same filename prefix)	{true, false}
repeat count	sets how often the scan should be executed (in addition to the first execution)	[0,999999]
Comment	additional comments	string

Part II. The EVE-Viewer

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Chapter 4. Getting Started

Depending on what you want to do [overview.html] (monitor/manipulate devices or execute scans) either activate the EveDevice [concepts/evedeviceperspective.html] or EveEngine [concepts/eveengineperspective.html] Perspective by clicking on the respective button at the top of the window (see following image for illustration).



Chapter 5. Concepts

Chapter 6. Reference

Part III.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Chapter 7. Concepts

PV Update Interval

The interval between process variable updates (in the GUI) is specified via the preferences. The value shown in the user interface will be updated up to x times where x is the given value.

The default is set to 1000ms. Values from 250ms to 60000ms are valid. If the interval is set too high (250ms) and there are many active connections (e.g. multiple device inspector views with a decent amount of entries) the response time of the application could increase.

To set the interval use the menu „Edit->Preferences...” and navigate to „Display->EVE Viewer”. After entering the desired value (which must be in the range mentioned above) click „OK”.

Please Note: The value is applied only to items added to the device inspector afterwards or after a restart !

Chapter 8. Reference

Device Classes (and their influence)

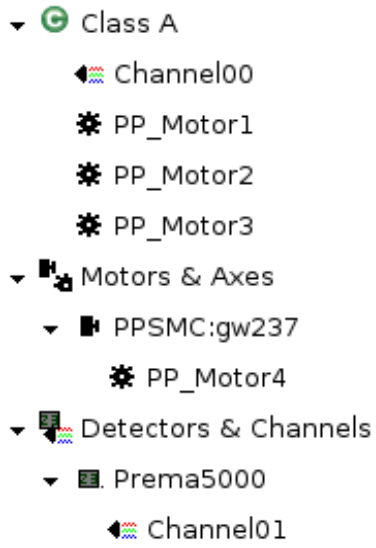
Each device (motor, motor axis, detector, detector channel and pre-/postscan device) could be assigned a class name. Depending on that class is the location in the tree hierarchy of the (Local) Devices View the device appears in. The following table shows where to expect a device with a certain class name:

Device	Class Name	Location
Motor	C	child of root element „C”
—	child of root element „Motors & Axes”	
Motor Axis	C	child of root element „C”
—	child of its parent (Motor)	
Detector	C	child of root element „C”
—	child of root element „Detectors & Channels”	
Detector Channel	C	child of root element „C”
—	child of its parent (Detector)	
(Pre-/Postscan) Device	C	child of root element „C”
—	child of root element „Devices”	

Example 1

Motor (PPSMC:gw237)	Axis 1 (PP_Motor1)	Axis 2 (PP_Motor2)	Axis 3 (PP_Motor3)	Axis 4 (PP_Motor4)
—	"Class A"	"Class A"	"Class A"	—

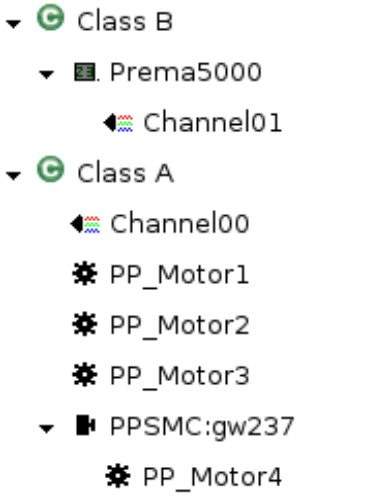
Prema5000	Channel00	Channel01
—	"Class A"	—



Example 2

Motor (PPSMC:gw237)	Axis 1 (PP_Motor1)	Axis 2 (PP_Motor2)	Axis 3 (PP_Motor3)	Axis 4 (PP_Motor4)
"Class A"	"Class A"	"Class A"	"Class A"	—

Prema5000	Channel00	Channel01
"Class B"	"Class A"	—



Appendix A. Release Notes

EVE-Editor

Version 1.2

- [Feature] Pause events now pause or continue the scan (or both)
- [Feature] Normalize Channel for Detector Channels (DetectorChannelView)
- [Feature] Names for Plot windows
- [Feature] Consistent presentation of device names (including lexicographical ordering)
- [Feature] Remove All Axes/Channels/Prescans/Postscans/Positionings/Plots/Events
- [Bug Fix] Selecting a Detector Channel in a Plot fails on first attempt

Version 1.1

- [Feature] Fill options „Save all Motor Positions” and "Save all Detector Values"