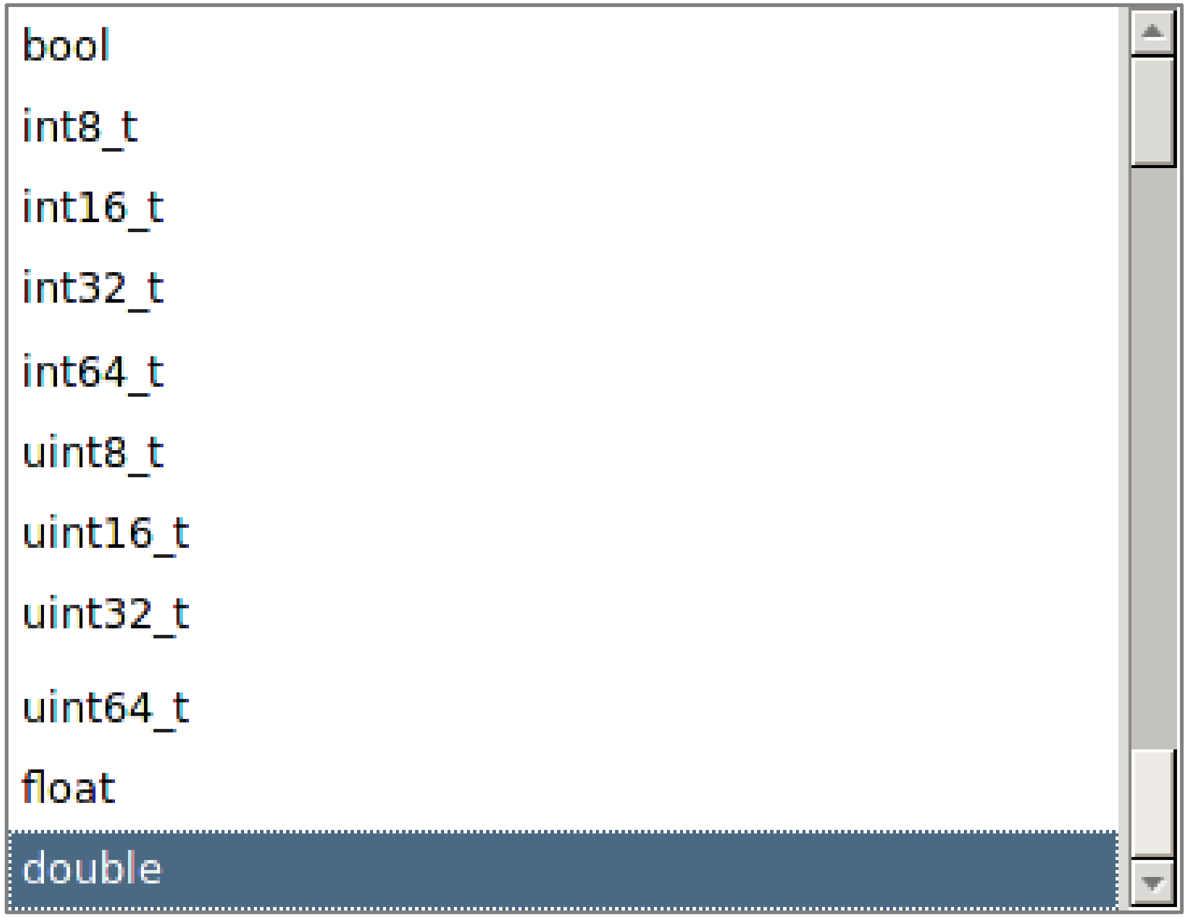
Data-Types

- With FESA3 we distinguish between default types and custom types (user defined types)
  - default
    - scalar
    - array
    - array2D
  - custom
    - custom-type-scalar
    - custom-type-array
    - custom-type-array2D

Scalar data types

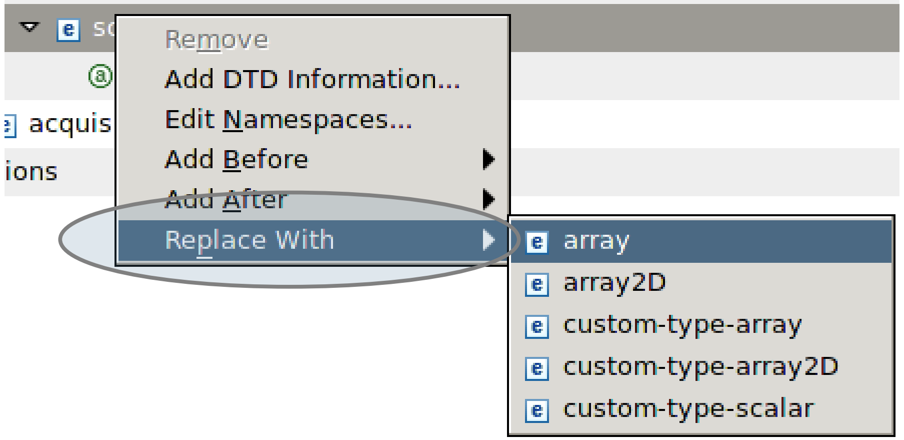


bool
int8_t
int16_t
int32_t
int64_t
uint8_t
uint16_t
uint32_t
uint64_t
float
double

- + custom types
- type “char” currently only for arrays

Array(2D) data types

- Use the context menu (rightclick on item)
- arrays can have different dimensions
  - one-dimensional (i.e. “string”)
  - two-dimensional (i.e. “string-array”)
  - a dimension can be constant or device-specific (variable)
- arrays can contain custom types



Middleware-Types

The current RDA middleware does not support the same types which are supported by FESA.

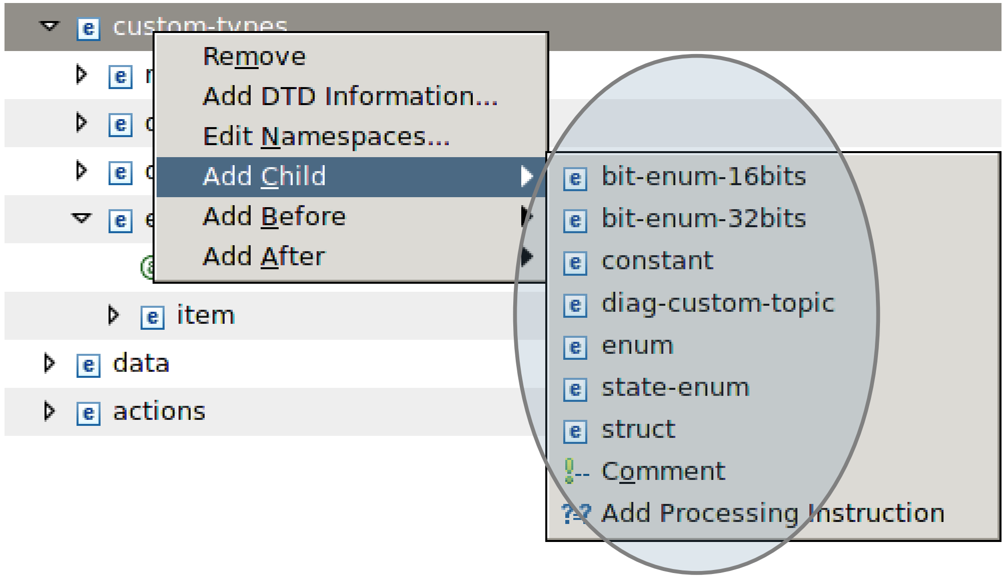
- Consequences:
  - It is not possible to use unsigned types in properties
  - It is difficult to use 2D array types in properties
  - It is not possible to use structs in properties

Custom data types

- Custom-types can be used in the design and in the C++ code
- Like in plain code, they make things more readable, prevent duplication and stop the usage of “magic-numbers”
- Use them whenever possible!

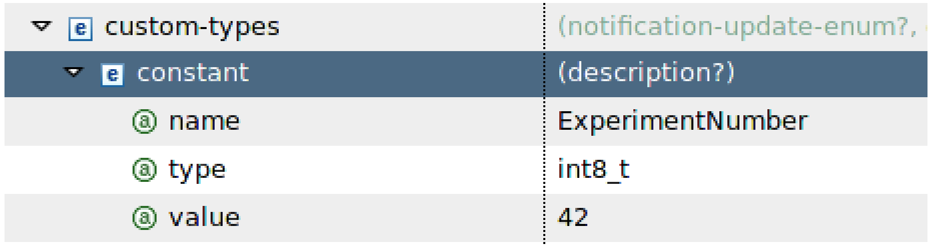
Custom data types

- **enum**
  - name to use in design and C++ code
  - items
    - Unique value
    - symbol to use in code
    - access (RW, RO, WO)
- **state-enum**
  - More or leass the same than enum
  - Some values predefined
- **constant**
- **struct**
- **bit-enum 16bits**
- **bit-enum 32bits**



Custom data types

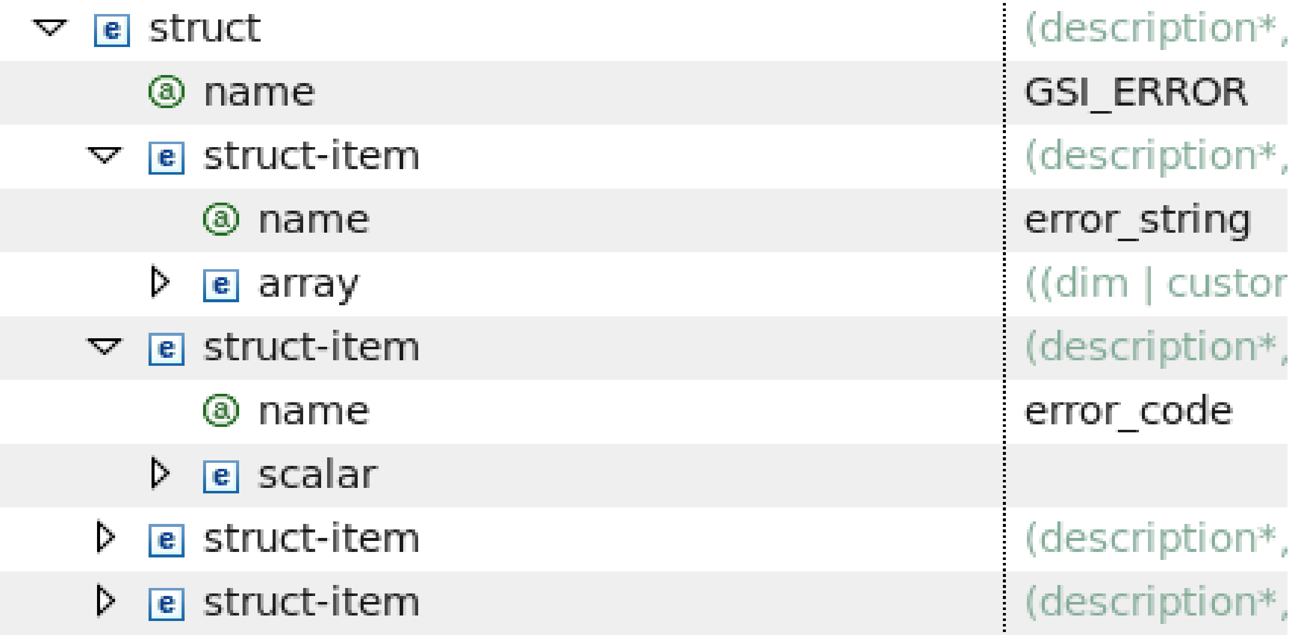
- **enum**
- **state-enum**
- **constant**
  - type can be any scalar-type
  - define the value in the design
  - due to a bug it is not yet possible to have string-constants
- **struct**
- **bit-enum 16bits**
- **bit-enum 32bits**



▽ e custom-types	(notification-update-enum?,
▽ e constant	(description?)
ⓐ name	ExperimentNumber
ⓐ type	int8_t
ⓐ value	42

Custom data types

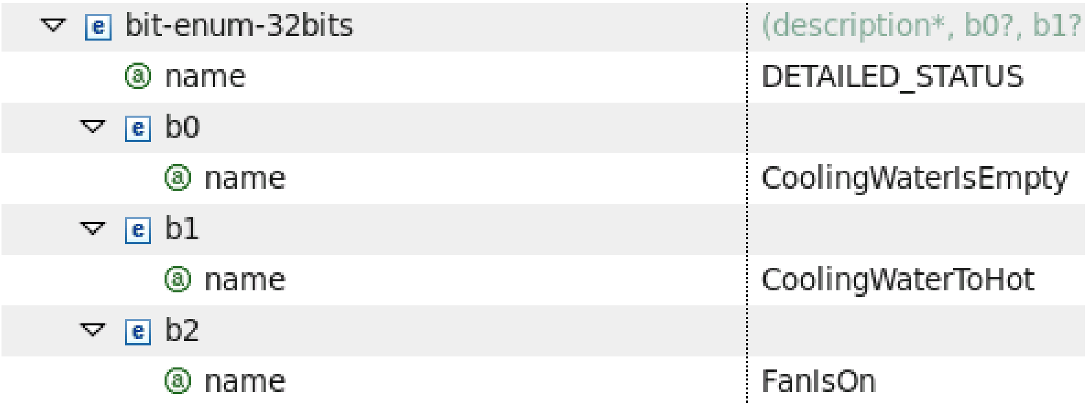
- **enum**
- **state-enum**
- **constant**
- **struct**
  - struct-item type can be any fesa-data-type
  - not possible to send structs directly via the middleware
- **bit-enum 16bits**
- **bit-enum 32bits**



▽ e struct	(description*,
ⓐ name	GSI_ERROR
▽ e struct-item	(description*,
ⓐ name	error_string
▷ e array	((dim | custor
▽ e struct-item	(description*,
ⓐ name	error_code
▷ e scalar	
▷ e struct-item	(description*,
▷ e struct-item	(description*,

Custom data types

- **enum**
- **state-enum**
- **constant**
- **struct**
- **bit-enum 16bits**
  - usage of a bitmask to address single-bit in the C++ code
  - only for scalar-types
  - often used for hardware-components
  - will get further updates in coming fesa-versions
- **bit-enum 32bits**
  - same as bit-enum 16bits, just more bits



▽ e bit-enum-32bits	(description*, b0?, b1?
ⓐ name	DETAILED_STATUS
▽ e b0	
ⓐ name	CoolingWaterIsEmpty
▽ e b1	
ⓐ name	CoolingWaterToHot
▽ e b2	
ⓐ name	FanIsOn

Mission

- **Create a class which periodically measures an array of values(use random values) in the RT-Action.**
- **The measured array has to be offered by an acquisition-property.**
- **Use a custom-type for the array-size.**
- **Use custom-types to define the minimum and maximum rand-values.**
- **The client should be able to switch the measurement ON and OFF. Use an enum-type for this.**

  (Manual Notification!)

On any problems: fesa-support@gsi.de

Thank you for your listening !!