

# AUTOSAR AP 예제

## - COM 01 (Local IPC) -

Junho Kwak

Architecture and Compiler for Embedded System LAB.

School of Electronics Engineering, KNU, KOREA

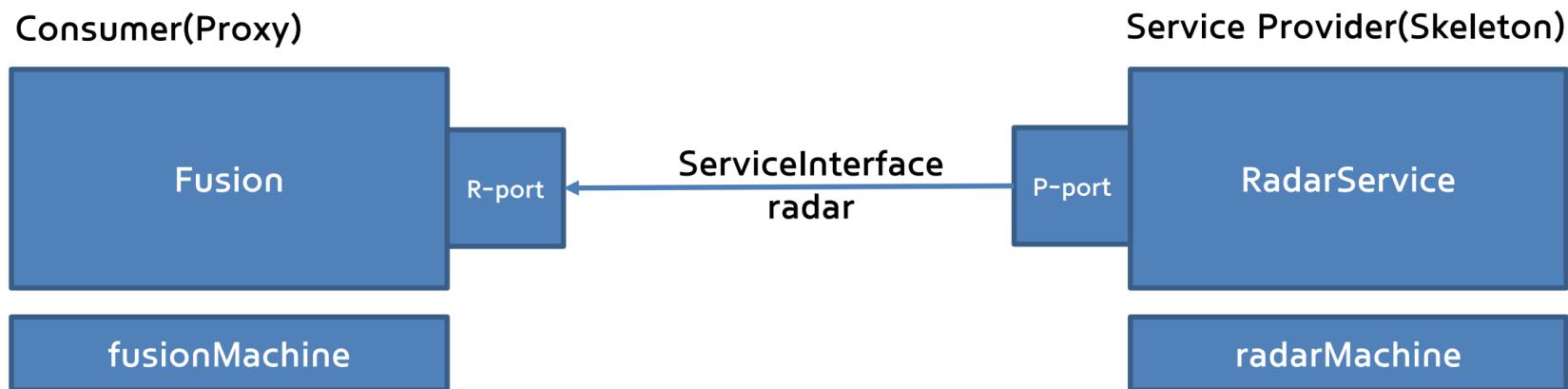
2024-07-07



# COM 01 (Local IPC) 예제 시스템

## ▪ Fusion – Radar 시스템

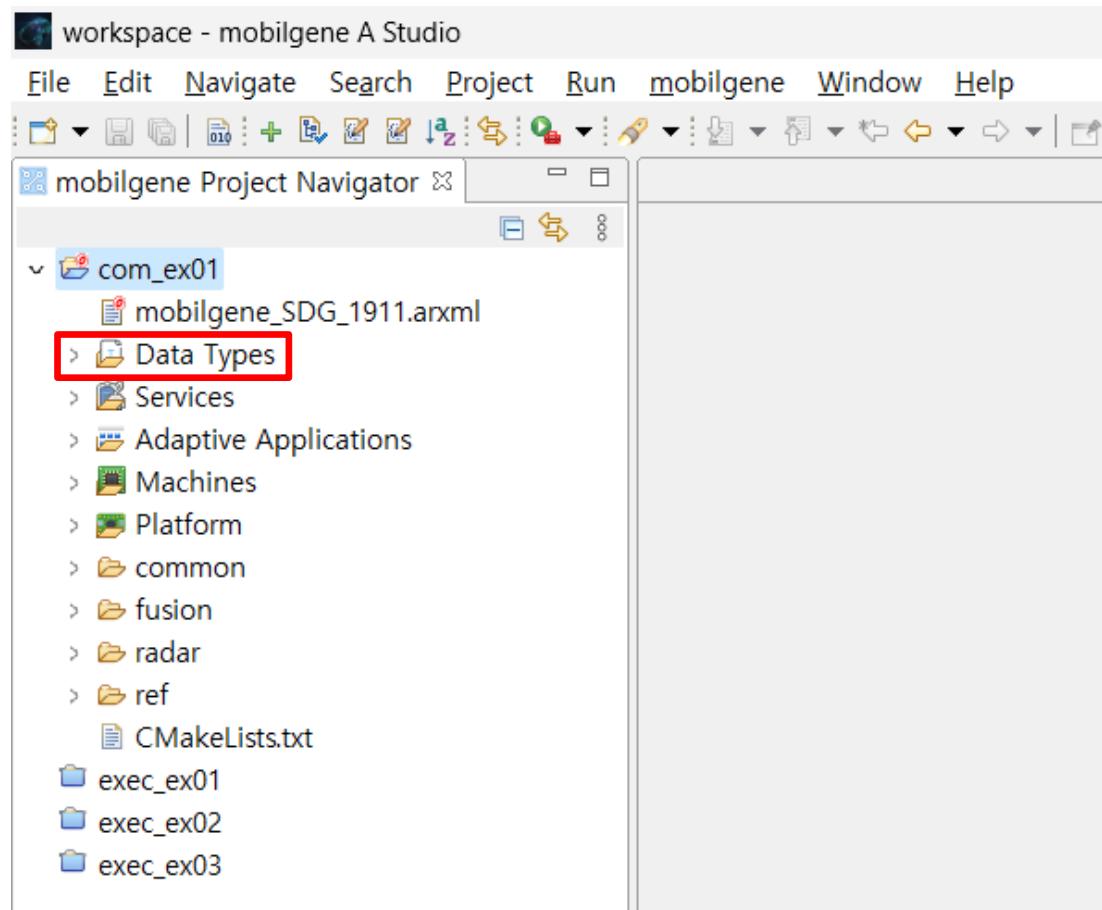
- ✓ Radar 서비스에 대해 두 어플리케이션이 통신하는 시스템임
  - ✓ Fusion : 서비스를 요청하는 Proxy로 동작함
  - ✓ Radar : 서비스를 제공하는 Skeleton으로 동작함
- ✓ Fusion과 Radar는 Local IPC를 통해 통신하여 서비스 동작 요청/동작을 수행함



# RadarService: Data Types 설정

- **Data Type Editor 활성화**

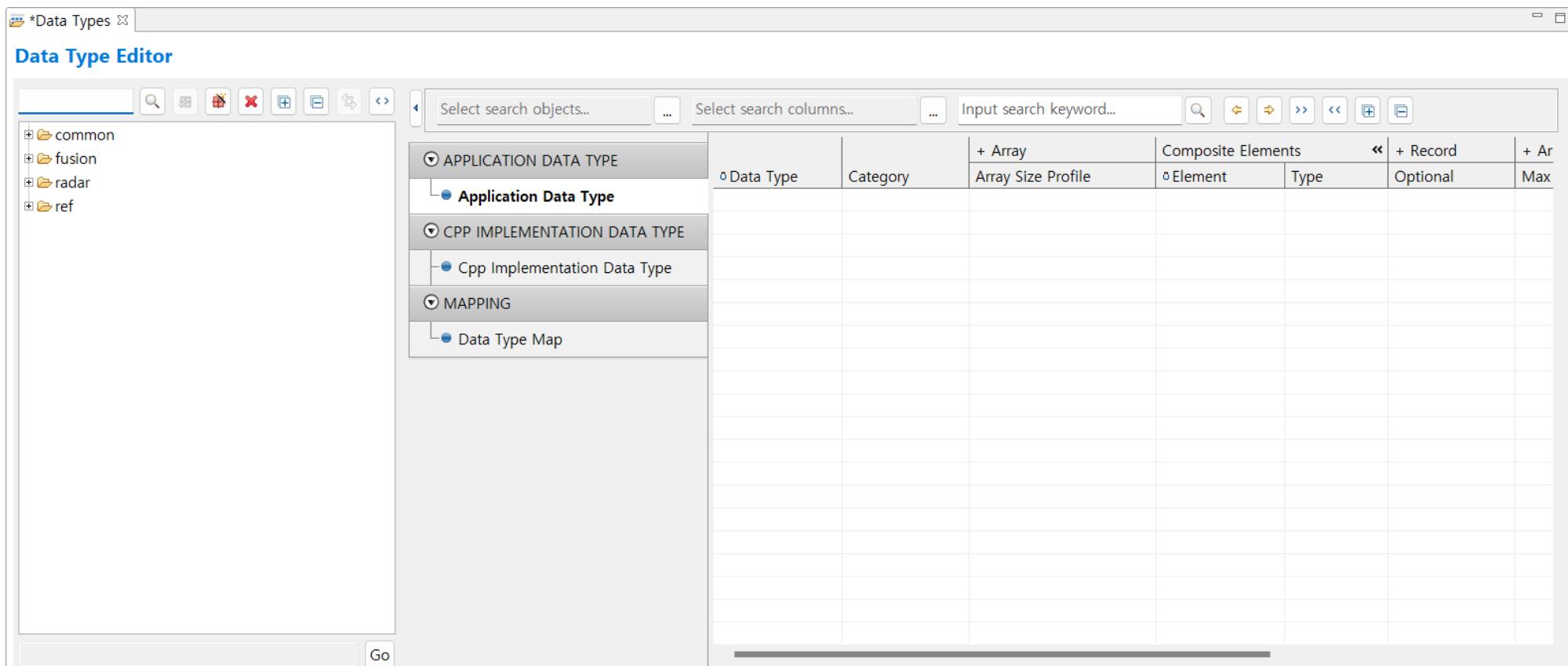
- ✓ 왼쪽의 'mobilgene Project Navigator' 창에서 해당 Project의 'Data Types'를 더블 클릭함



# RadarService: Data Types 설정

- **Data Type Editor 활성화 확인**

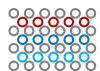
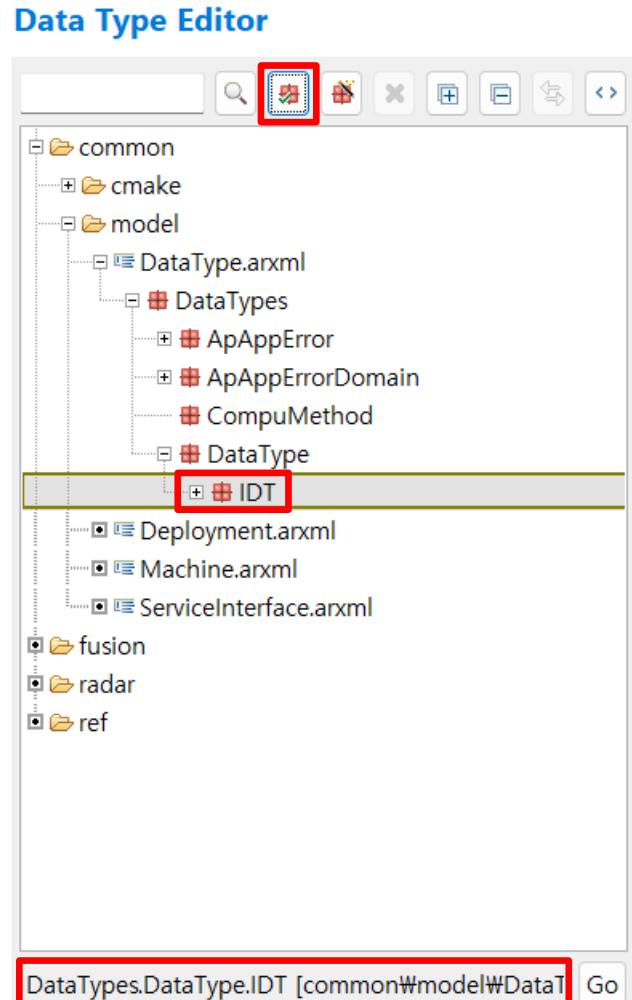
- ✓ 활성화 된 Data Type Editor를 확인함



# RadarService: Data Types 설정

## ▪ Default Package 설정

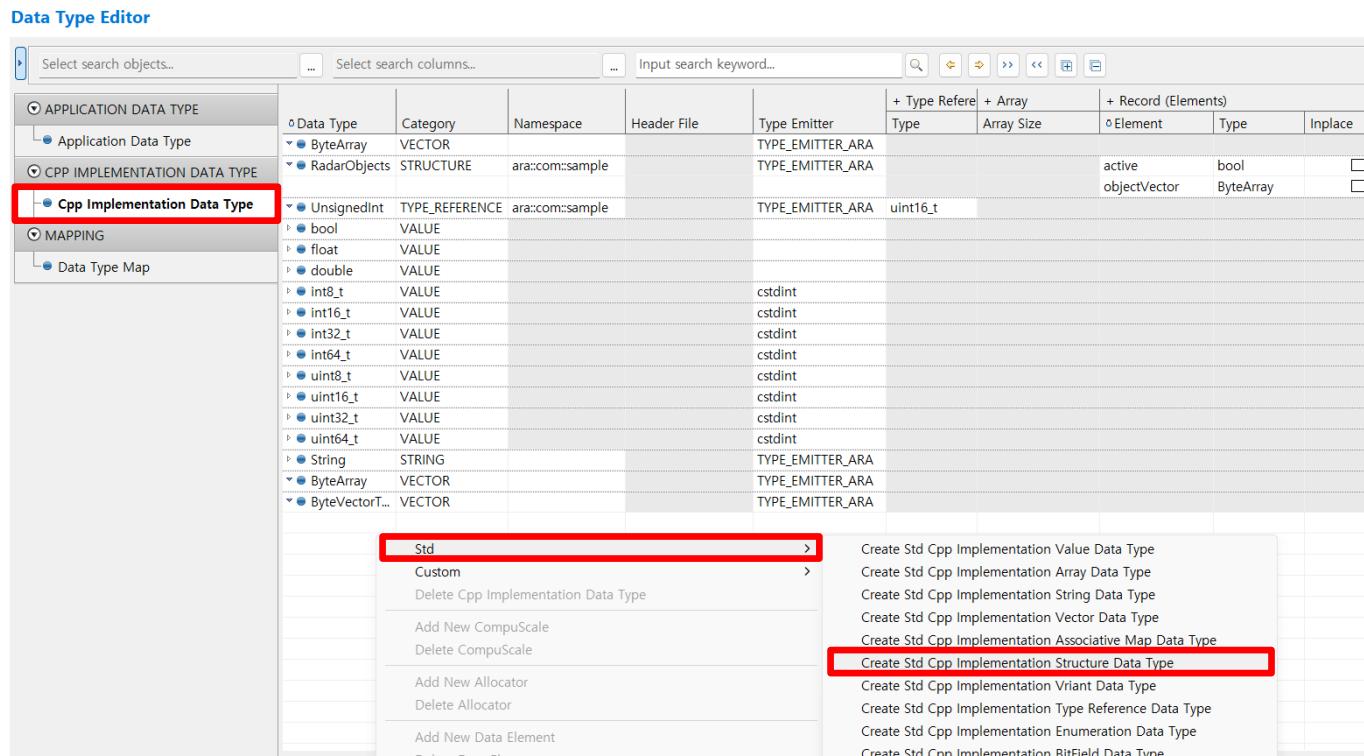
- ✓ 설정이 저장될 Default Package를 설정함
  - ✓ Default Package로 설정할 'IDT' Package를 선택함
  - ✓ 우측 상단의 'Set Default Package' 버튼을 클릭함
  - ✓ 하단에 설정된 Default Package를 확인함



# RadarService: Data Types 설정

## ▪ 'Position' 데이터 타입 추가 (1)

- ✓ 데이터 타입 추가를 위해 좌측의 'Cpp Implementation Data Type' 탭으로 이동함
- ✓ 빈 곳에서 우클릭을 하여 새로운 데이터 타입을 추가함
- ✓ 'Position' 데이터 타입은 구조체이기 때문에 'Std' - 'Create Std Cpp Implementation Structure Data Type'을 클릭함

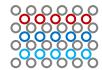


# RadarService: Data Types 설정

- ‘Position’ 데이터 타입 추가 (2)
  - ✓ 생성된 새로운 데이터 타입을 확인함

Data Type Editor

Select search objects...		Select search columns...		Input search keyword...		+ Type Refere		+ Array		+ Record (Elements)	
DATA TYPE	Category	Namespace	Header File	Type Emitter	Type	Array Size	Element	Type	Inplace		
APPLICATION DATA TYPE	Data Type	VECTOR		TYPE_EMITTER_ARA							
Application Data Type	ByteArray	VECTOR		TYPE_EMITTER_ARA			active	bool			
CPP IMPLEMENTATION DATA TYPE	RadarObjects	STRUCTURE	ara::com::sample	TYPE_EMITTER_ARA			objectVector	ByteArray			
Cpp Implementation Data Type	UnsignedInt	TYPE_REFERENCE	ara::com::sample	TYPE_EMITTER_ARA	uint16_t						
MAPPING	bool	VALUE									
Data Type Map	float	VALUE									
	double	VALUE									
	int8_t	VALUE		cstdint							
	int16_t	VALUE		cstdint							
	int32_t	VALUE		cstdint							
	int64_t	VALUE		cstdint							
	uint8_t	VALUE		cstdint							
	uint16_t	VALUE		cstdint							
	uint32_t	VALUE		cstdint							
	uint64_t	VALUE		cstdint							
	String	STRING		TYPE_EMITTER_ARA							
	ByteArray	VECTOR		TYPE_EMITTER_ARA							
	ByteVectorT	VECTOR		TYPE_EMITTER_ARA							
	StdCpp_1	STRUCTURE									

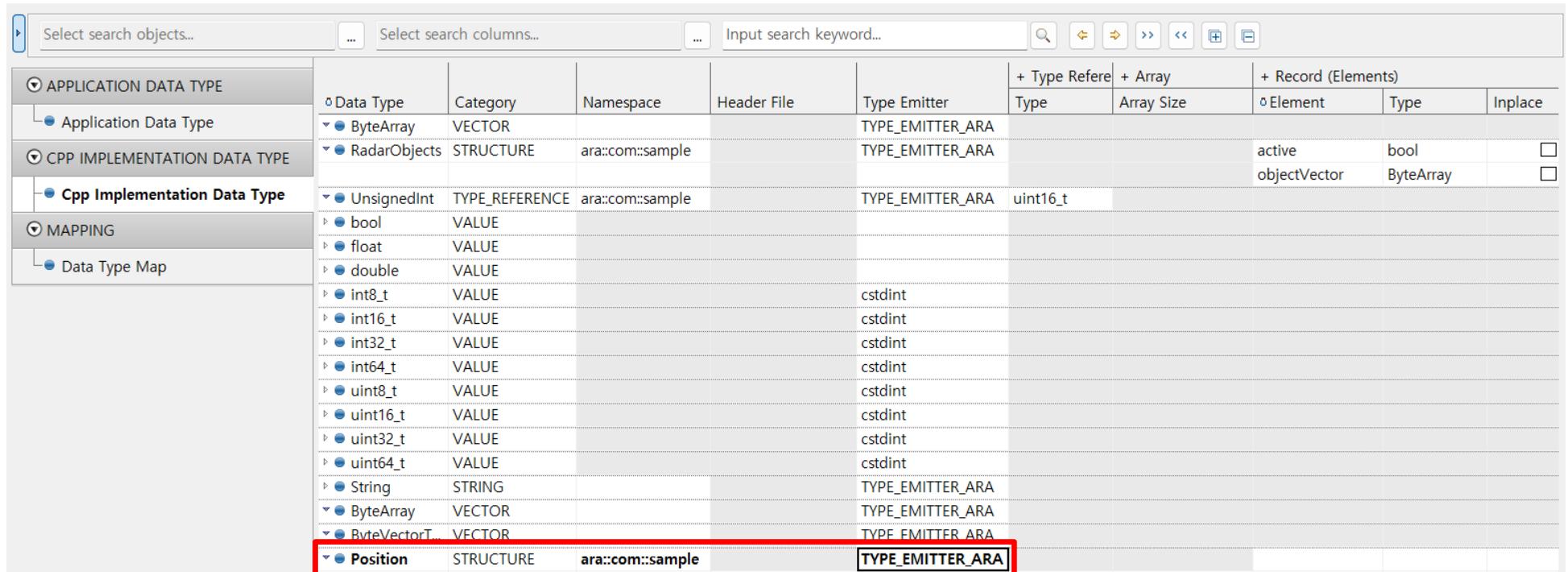


# RadarService: Data Types 설정

## ▪ 'Position' 데이터 타입 추가 (3)

- ✓ 수정할 부분을 더블 클릭하여 내용을 다음과 같이 작성함
  - ✓ Data Type : 'Position'
  - ✓ Namespace : 'ara::com::sample'
  - ✓ Type Emitter : 'TYPE\_EMITTER\_ARA'

Data Type Editor

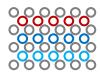
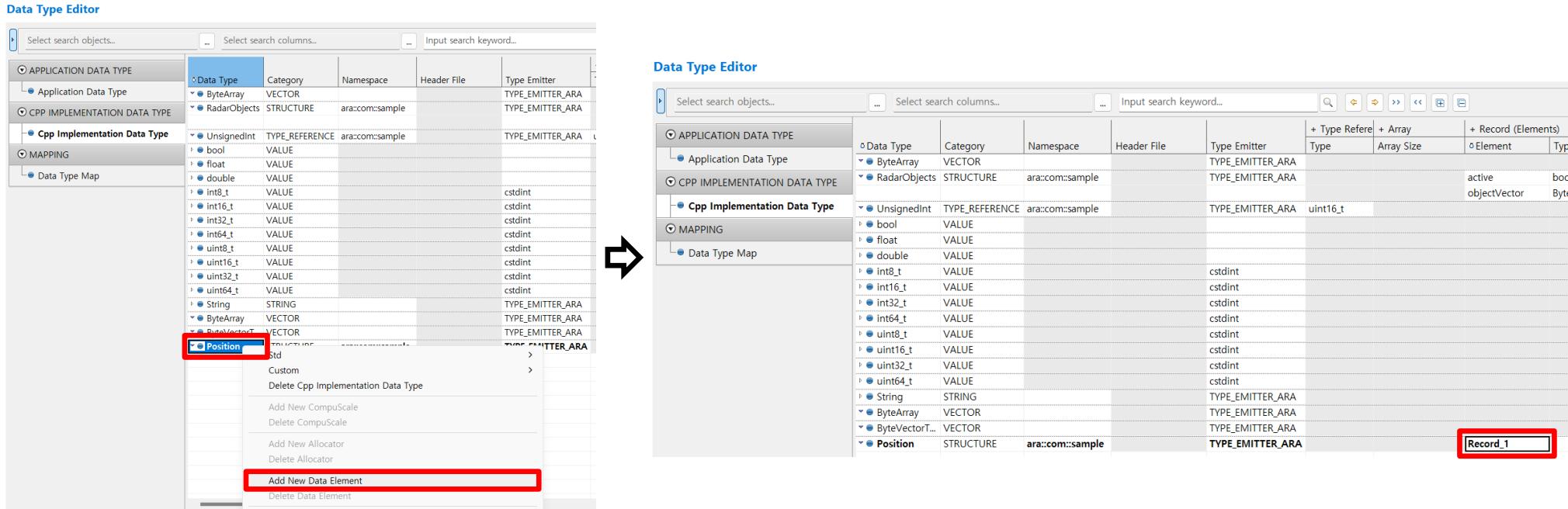


Select search objects...		Select search columns...		Input search keyword...		+ Type Reference		+ Array		+ Record (Elements)		
Category	Type	Category	Namespace	Header File	Type Emitter	Type	Array Size	Element	Type	Inplace		
APPLICATION DATA TYPE												
Application Data Type	ByteArray	VECTOR			TYPE_EMITTER_ARA							
CPP IMPLEMENTATION DATA TYPE	RadarObjects	STRUCTURE	ara::com::sample		TYPE_EMITTER_ARA			active	bool			
Cpp Implementation Data Type	UnsignedInt	TYPE_REFERENCE	ara::com::sample		TYPE_EMITTER_ARA	uint16_t		objectVector	ByteArray			
MAPPING	bool	VALUE										
Data Type Map	float	VALUE										
	double	VALUE										
	int8_t	VALUE			cstdint							
	int16_t	VALUE			cstdint							
	int32_t	VALUE			cstdint							
	int64_t	VALUE			cstdint							
	uint8_t	VALUE			cstdint							
	uint16_t	VALUE			cstdint							
	uint32_t	VALUE			cstdint							
	uint64_t	VALUE			cstdint							
	String	STRING			TYPE_EMITTER_ARA							
	ByteArray	VECTOR			TYPE_EMITTER_ARA							
	BvteVectorT	VECTOR			TYPE_EMITTER_ARA							
	Position	STRUCTURE	ara::com::sample		TYPE_EMITTER_ARA							

# RadarService: Data Types 설정

## ▪ 'Position' 데이터 타입 추가 (4)

- ✓ 생성한 'Position' 데이터 타입에 구조체 내부 Element를 추가함
- ✓ 'Position' 위에서 우클릭을 하여 'Add New Data Element'를 클릭함
- ✓ 생성된 새로운 Element를 확인함



# RadarService: Data Types 설정

#### ▪ ‘Position’ 데이터 탑입 추가 (5)

- ✓ 생성된 Element의 이름과 Type을 다음과 같이 수정함
    - ✓ Element : 'x'
    - ✓ Type : 'int32\_t'

## Data Type Editor

Select search objects...		Select search columns...		Input search keyword...								
APPLICATION DATA TYPE	Data Type	Category	Namespace	Header File	Type Emitter	+ Type Reference		+ Array		+ Record (Elements)		
						Type	Array Size	Element	Type	Inplace		
Application Data Type	ByteArray	VECTOR			TYPE_EMITTER_ARA							
CPP IMPLEMENTATION DATA TYPE	RadarObjects	STRUCTURE	ara::com::sample		TYPE_EMITTER_ARA			active	bool			
Cpp Implementation Data Type	UnsignedInt	TYPE_REFERENCE	ara::com::sample		TYPE_EMITTER_ARA	uint16_t		objectVector	ByteArray			
MAPPING	bool	VALUE			cstdint							
Data Type Map	float	VALUE			cstdint							
	double	VALUE			cstdint							
	int8_t	VALUE			cstdint							
	int16_t	VALUE			cstdint							
	int32_t	VALUE			cstdint							
	int64_t	VALUE			cstdint							
	uint8_t	VALUE			cstdint							
	uint16_t	VALUE			cstdint							
	uint32_t	VALUE			cstdint							
	uint64_t	VALUE			cstdint							
	String	STRING			TYPE_EMITTER_ARA							
	ByteArray	VECTOR			TYPE_EMITTER_ARA							
	ByteVectorT...	VECTOR			TYPE_EMITTER_ARA							
	Position	STRUCTURE	ara::com::sample		TYPE_EMITTER_ARA			x	int32_t			

# RadarService: Data Types 설정

## ▪ ‘Position’ 데이터 타입 추가 (6)

- ✓ 동일한 방법으로 2개의 Element를 더 추가하고 다음 그림과 같이 수정함

Data Type Editor

The screenshot shows the Data Type Editor interface with the following details:

Category	Namespace	Header File	Type Emitter	Type	Array Size	Element	Type	Inplace
VECTOR			TYPE_EMITTER_ARA			active	bool	
STRUCTURE	ara::com::sample		TYPE_EMITTER_ARA			objectVector	ByteArray	
TYPE_REFERENCE	ara::com::sample		TYPE_EMITTER_ARA	uint16_t				
VALUE				cstdint				
VALUE				cstdint				
VALUE				cstdint				
VALUE				cstdint				
VALUE				cstdint				
VALUE				cstdint				
STRING			TYPE_EMITTER_ARA					
VECTOR			TYPE_EMITTER_ARA					
STRUCTURE	ara::com::sample		TYPE_EMITTER_ARA			x	int32_t	
						y	int32_t	
						z	int32_t	

The 'Position' structure is defined with three elements: x, y, and z, each of type int32\_t. The 'z' element is highlighted with a red box.

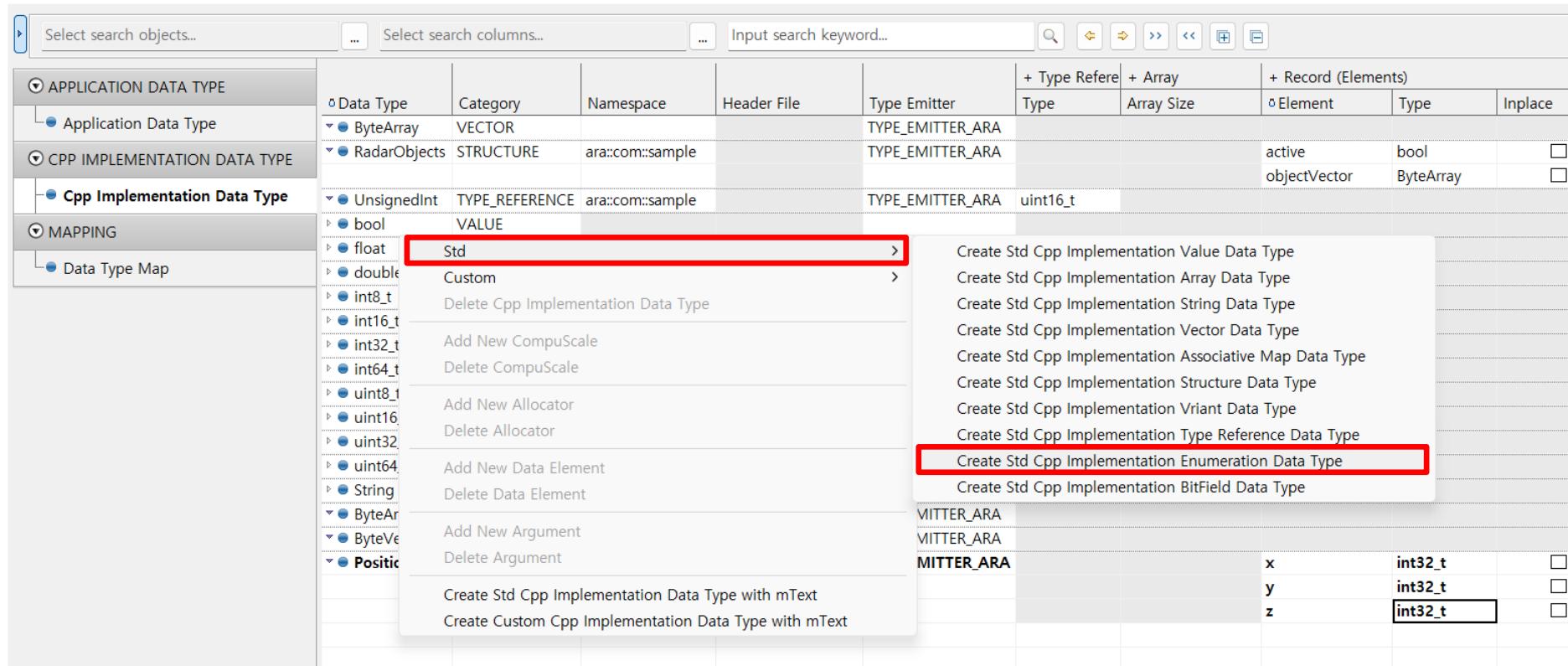


# RadarService: Data Types 설정

## ■ 'FusionVariant' 데이터 타입 추가 (1)

- ✓ 빈 곳에서 우클릭을 하여 새로운 데이터 타입을 추가함
- ✓ 'FusionVariant' 데이터 타입은 Enum이기 때문에 'Std' - 'Create Std Cpp Implementation Enumeration Data Type'을 클릭함

Data Type Editor



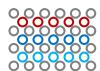
# RadarService: Data Types 설정

- ‘FusionVariant’ 데이터 타입 추가 (2)
  - ✓ 생성된 새로운 데이터 타입을 확인함

Data Type Editor

The screenshot shows the Data Type Editor interface with the following details:

Category	Namespace	Type Emitter	Type	Array Size	Element	Type	Inplace
APPLICATION DATA TYPE							
Application Data Type							
CPP IMPLEMENTATION DATA TYPE							
Cpp Implementation Data Type							
MAPPING							
Data Type Map							
DATA TYPES							
ByteArray	VECTOR						
RadarObjects	STRUCTURE	ara::com::sample					
UnsignedInt	TYPE_REFERENCE	ara::com::sample					
bool	VALUE						
float	VALUE						
double	VALUE						
int8_t	VALUE		cstdint				
int16_t	VALUE		cstdint				
int32_t	VALUE		cstdint				
int64_t	VALUE		cstdint				
uint8_t	VALUE		cstdint				
uint16_t	VALUE		cstdint				
uint32_t	VALUE		cstdint				
uint64_t	VALUE		cstdint				
String	STRING		TYPE_EMITTER_ARA				
ByteArray	VECTOR		TYPE_EMITTER_ARA				
ByteVectorT...	VECTOR		TYPE_EMITTER_ARA				
Position	STRUCTURE	ara::com::sample	TYPE_EMITTER_ARA		x	int32_t	
					y	int32_t	
					z	int32_t	
StdCpp_1	ENUMERATION						

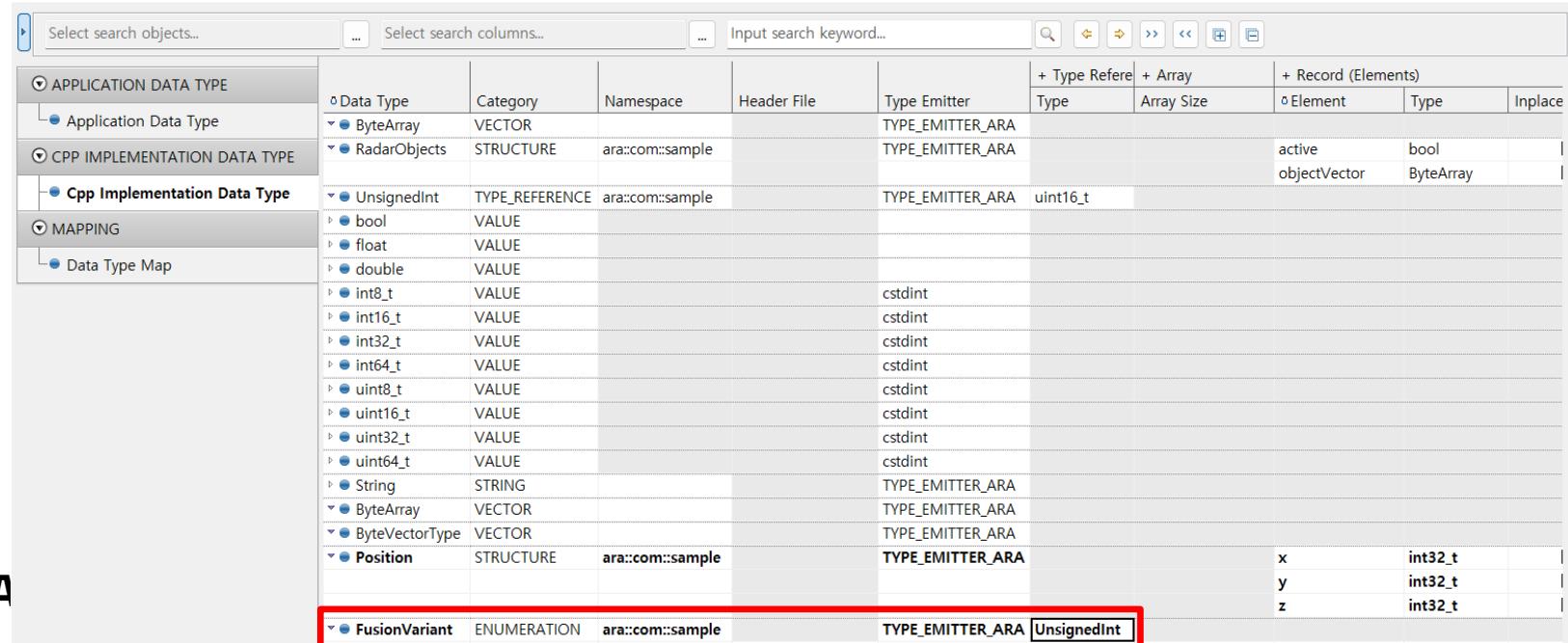


# RadarService: Data Types 설정

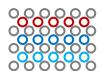
## ■ 'FusionVariant' 데이터 타입 추가 (3)

- ✓ 수정할 부분을 더블 클릭하여 내용을 다음과 같이 작성함
  - ✓ Data Type : 'FusionVariant'
  - ✓ Namespace : 'ara::com::sample'
  - ✓ Type Emitter : 'TYPE\_EMITTER\_ARA'
  - ✓ Type : 'UnsignedInt'

Data Type Editor



Select search objects...		Select search columns...		Input search keyword...												
APPLICATION DATA TYPE		Data Type	Category	Namespace	Header File	Type Emitter	+ Type Reference		+ Array		+ Record (Elements)					
CPP IMPLEMENTATION DATA TYPE		ByteArray	VECTOR			TYPE_EMITTER_ARA	Type		Array Size	Element	Type	Inplace				
Application Data Type		ByteString	STRUCTURE	ara::com::sample		TYPE_EMITTER_ARA				active	bool					
Cpp Implementation Data Type		UnsignedInt	TYPE_REFERENCE	ara::com::sample		TYPE_EMITTER_ARA	uint16_t			objectVector	ByteArray					
Mapping		bool	VALUE													
Data Type Map		float	VALUE													
		double	VALUE													
		int8_t	VALUE			cstdint										
		int16_t	VALUE			cstdint										
		int32_t	VALUE			cstdint										
		int64_t	VALUE			cstdint										
		uint8_t	VALUE			cstdint										
		uint16_t	VALUE			cstdint										
		uint32_t	VALUE			cstdint										
		uint64_t	VALUE			cstdint										
		String	STRING			TYPE_EMITTER_ARA										
		ByteArray	VECTOR			TYPE_EMITTER_ARA										
		ByteVectorType	VECTOR			TYPE_EMITTER_ARA										
		Position	STRUCTURE	ara::com::sample		TYPE_EMITTER_ARA		x		int32_t						
		FusionVariant	ENUMERATION	ara::com::sample		TYPE_EMITTER_ARA	UnsignedInt	y		int32_t						
								z		int32_t						



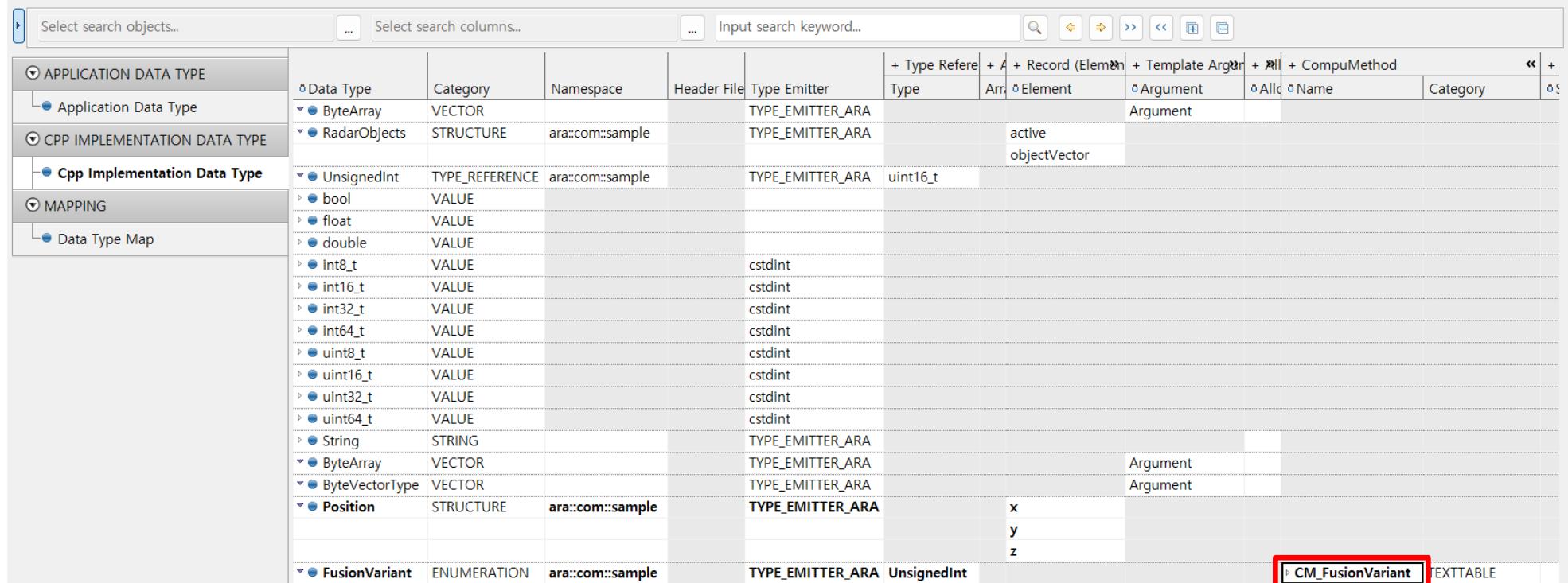
A

# RadarService: Data Types 설정

## ■ ‘FusionVariant’ 데이터 타입 추가 (4)

- ✓ 우측에 생성되어 있는 CompuMethod 확인
- ✓ CompuMethod의 이름을 ‘CM\_FusionVariant’로 수정함

Data Type Editor



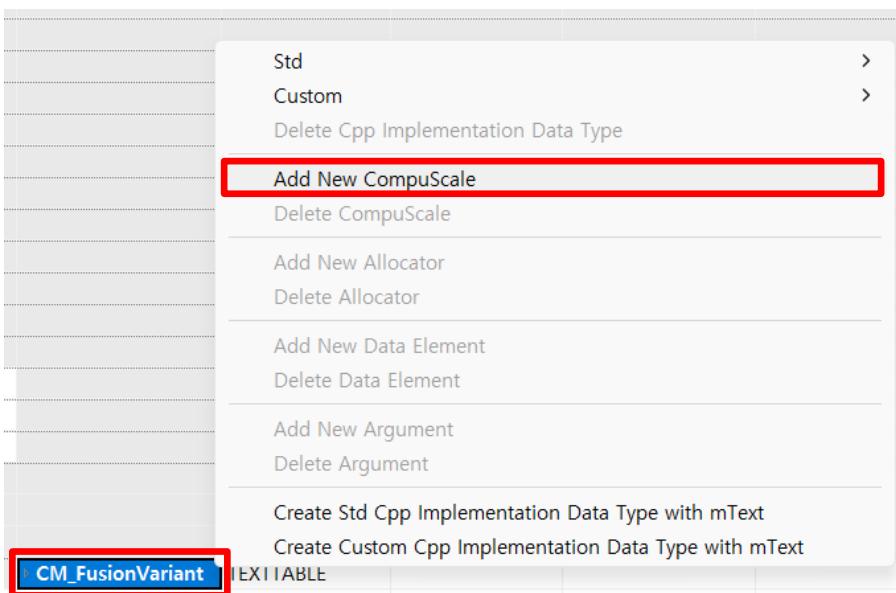
Select search objects...		Select search columns...		Input search keyword...								«	»
APPLICATION DATA TYPE		Data Type	Category	Namespace	Header File	Type Emitter	+ Type Reference	+ Record (Element)	+ Template Argument	+ All	+ CompuMethod	«	»
CPP IMPLEMENTATION DATA TYPE							Type	Arr.	Element	Argument	All	Name	Category
Application Data Type	ByteArray	VECTOR				TYPE_EMITTER_ARA				Argument			
Cpp Implementation Data Type	RadarObjects	STRUCTURE	ara::com::sample			TYPE_EMITTER_ARA			active				
	UnsignedInt	TYPE_REFERENCE	ara::com::sample			TYPE_EMITTER_ARA	uint16_t		objectVector				
	bool	VALUE											
	float	VALUE											
	double	VALUE											
	int8_t	VALUE				cstdint							
	int16_t	VALUE				cstdint							
	int32_t	VALUE				cstdint							
	int64_t	VALUE				cstdint							
	uint8_t	VALUE				cstdint							
	uint16_t	VALUE				cstdint							
	uint32_t	VALUE				cstdint							
	uint64_t	VALUE				cstdint							
	String	STRING				TYPE_EMITTER_ARA							
	ByteArray	VECTOR				TYPE_EMITTER_ARA			Argument				
	ByteVectorType	VECTOR				TYPE_EMITTER_ARA			Argument				
	Position	STRUCTURE	ara::com::sample			TYPE_EMITTER_ARA	x						
							y						
							z						
	FusionVariant	ENUMERATION	ara::com::sample			TYPE_EMITTER_ARA	UnsignedInt			CM_FusionVariant			



# RadarService: Data Types 설정

## ▪ 'FusionVariant' 데이터 타입 추가 (5)

- ✓ Enum 설정을 하기 위해 CompuScale을 추가함
- ✓ 'CM\_FusionVariant'에서 우클릭하여 'Add New CompuScale'을 클릭함
- ✓ 생성된 CompuScale을 확인함



+ CompuMethod		+ CompuMethod:: CompuScales			
Name	Category	Symbol	Short Label	Limit	Mask
Literal_1					

The screenshot shows a table titled '+ CompuMethod' with a sub-table '+ CompuMethod:: CompuScales'. The columns are: Name, Category, Symbol, Short Label, Limit, and Mask. A single row is present with the value 'Literal\_1' in the 'Name' column. A red box highlights the 'Literal\_1' entry.

# RadarService: Data Types 설정

- ‘FusionVariant’ 데이터 타입 추가 (6)
    - ✓ 생성된 CompuScale을 다음과 같이 수정함
      - ✓ Symbol : ‘FV\_CHINA’
      - ✓ Limit : ‘0x01’

+ CompuMethod		<< + CompuMethod: CompuScales			
Name	Category	Symbol	Short Label	Limit	Mask
CM_FusionVariant	TEXTTABLE	FV_CHINA		0x01	



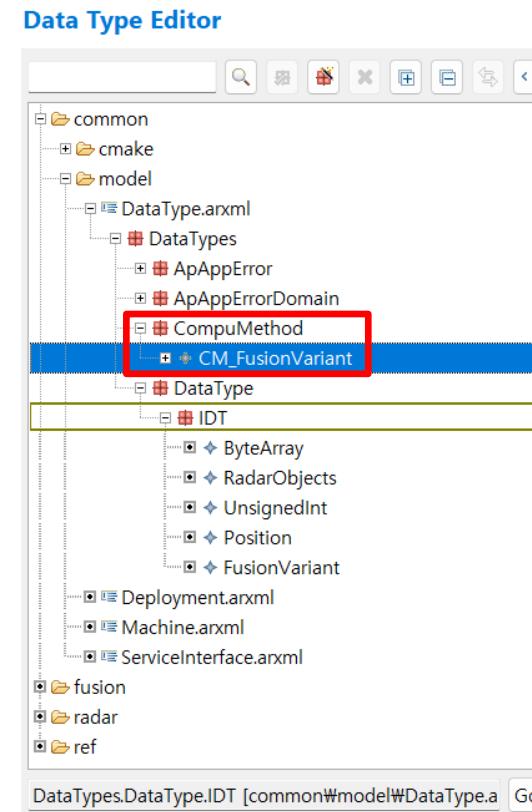
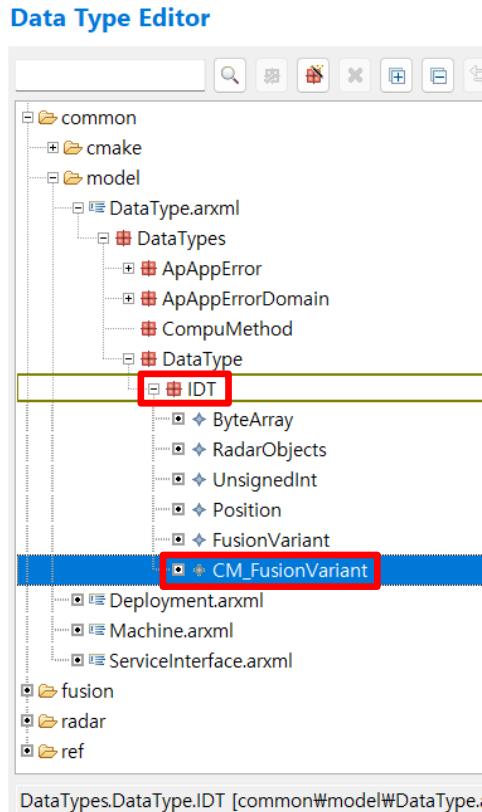
# RadarService: Data Types 설정

- ‘FusionVariant’ 데이터 타입 추가 (7)
    - ✓ 동일한 방법으로 3개의 CompuScale을 더 추가하고 다음 그림과 같이 수정함

# RadarService: Data Types 설정

## ▪ 'CM\_FusionVariant' 설정 Package 이동

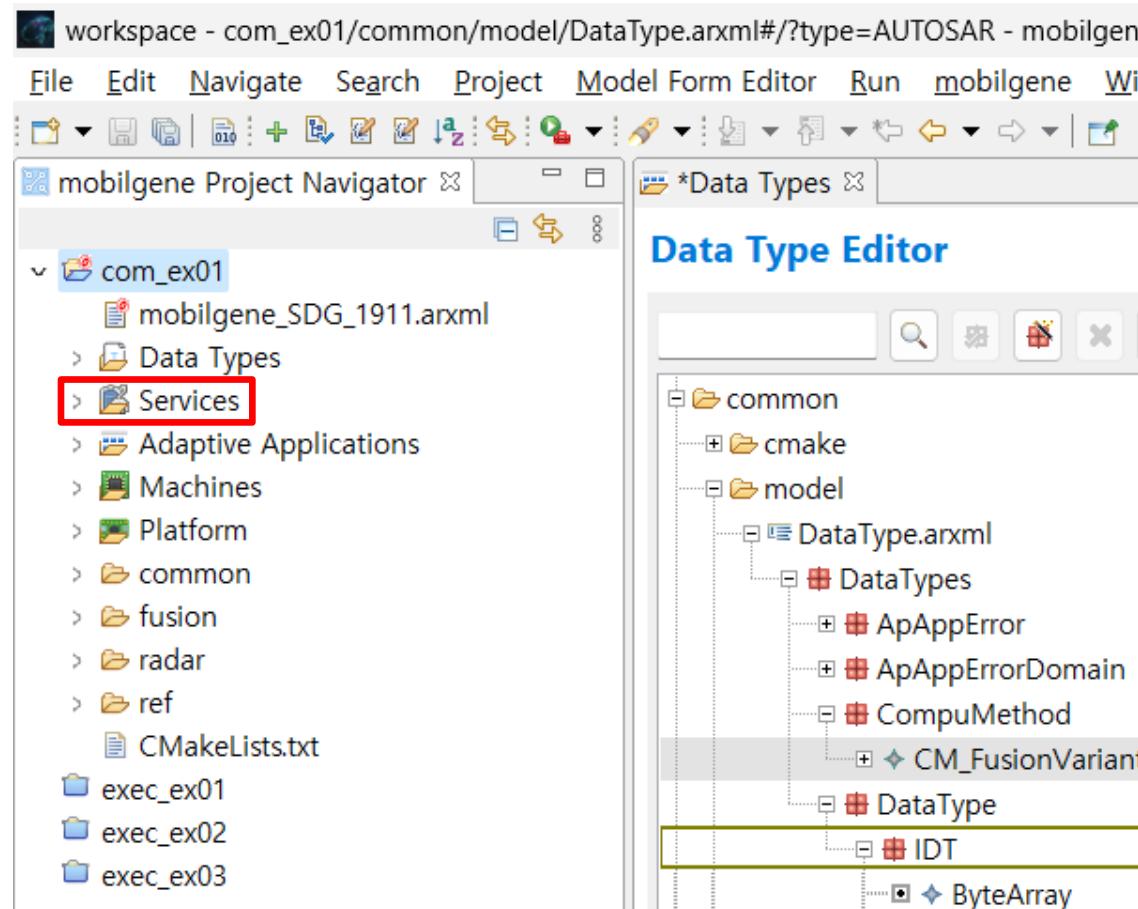
- ✓ Package 별 관리를 위해 'CM\_FusionVariant'를 'CompuMethod' Package로 이동함
- ✓ 드래그 앤 드롭을 통해 설정 내용의 이동을 수행할 수 있음



# RadarService: Services 설정

- **Service Editor 활성화**

- ✓ 왼쪽의 'mobilgene Project Navigator' 창에서 해당 Project의 'Services'를 더블 클릭함



# RadarService: Services 설정

- **Service Editor 활성화 확인**

- ✓ 활성화 된 Service Editor를 확인함

The screenshot shows the Service Editor application window. At the top, there are tabs for "Data Types" and "Services". The "Services" tab is active, indicated by a blue background. Below the tabs is a toolbar with various icons for search, filter, and navigation. The main area is divided into two sections: a tree view on the left and a list view on the right.

**Left Panel (Tree View):**

- common
- fusion
- radar
- ref

**Right Panel (List View):**

The list view displays a hierarchical structure of service components:

- Service Interface
  - ApApplication Error
  - Port Connection
- SERIALIZATION CONFIG.
  - Transformation Props.
  - Someip Data Prototype Transformation Props.
  - TLV Data ID Definition
  - Transformation Props. - Service Interface Element Map
- SOME/IP BINDING
  - Interface Deployment
  - Provided Instance
  - P-Instance Evt/Method/Evt Group
  - Required Instance
  - R-Instance Method/Evt Group

Below the list view is a table with the following columns:

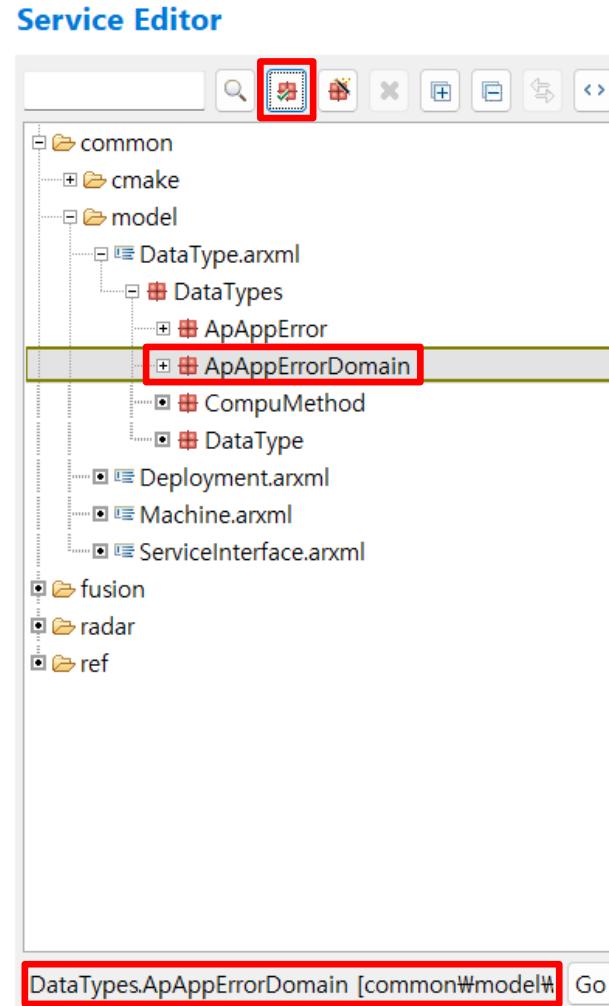
Service Interface	Namespace	Major Ver.	Minor Ver.	Events	Event	Data Type

At the bottom of the window, there is a status bar with the text "DataTypes.DataType.IDT [common#model#DataT Go]" and a progress bar.

# RadarService: Services 설정

## ▪ Default Package 설정

- ✓ 'ApAppErrorHandler' Package를 Default Package로 설정하고 아래 창을 통해 확인함

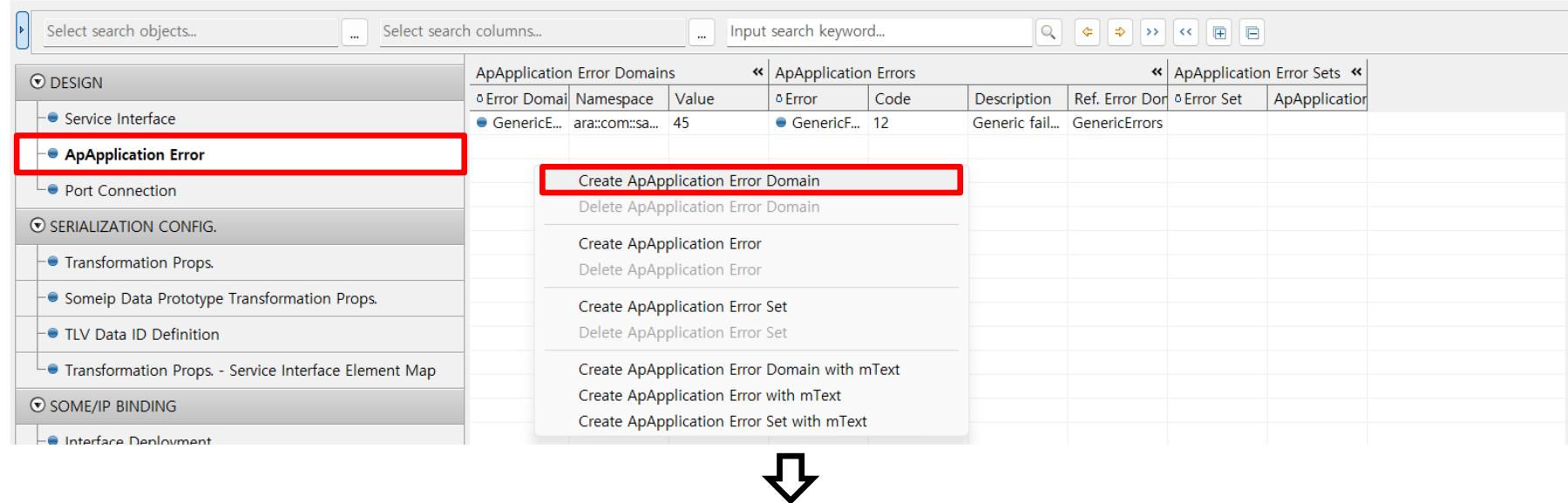


# RadarService: Services 설정

## ■ ApApplication Error Domain 설정 (1)

- ✓ 좌측의 'ApApplication Error' 탭으로 이동함
- ✓ 빈 곳에서 우클릭하여 'Create ApApplication Error Domain'을 클릭함

Service Editor



Service Editor

This screenshot shows the Service Editor after the creation of a new ApApplication Error Domain. The tree view on the left now includes 'ApApplication Error Domain.1' under the 'ApApplication Error' node. In the central table, a new row has been added for this domain: 'ApAppErrorDomain.1' with 'ara::com::sa...' as the Namespace, '0' as the Value, and 'GenericErrors' as the Ref. Error Domains. The rest of the table structure remains the same as in the previous screenshot.

Error Domain	Namespace	Value	Error	Code	Description	Ref. Error Domains	Error Set	ApApplications
GenericErrors	ara::com::sa...	45	GenericF...	12	Generic fail...	GenericErrors		
ApAppErrorDomain.1	ara::com::sa...	0						

# RadarService: Services 설정

## ▪ ApApplication Error Domain 설정 (2)

- ✓ 생성한 ApApplication Error Domain를 다음과 같이 수정함
  - ✓ Error Domain : 'SpecificErrors'
  - ✓ Namespace : 'ara::com::sample'
  - ✓ Value : '44'

Service Editor

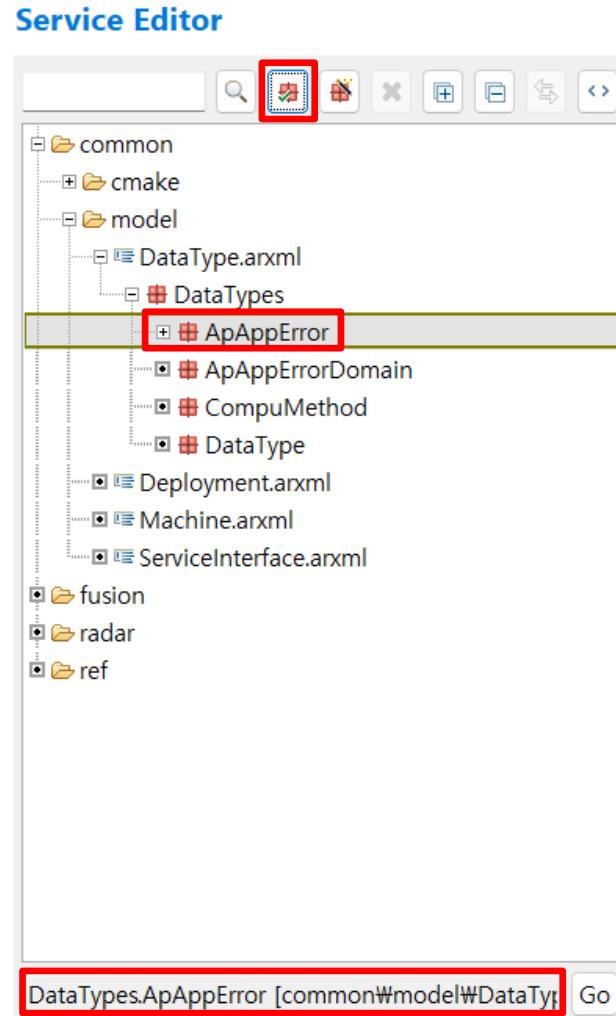
The screenshot shows the Service Editor interface with the 'ApApplication Error Domains' table highlighted. The table has columns: Error Domain, Namespace, Value, Error, Code, Description, Ref. Error Dor, Error Set, and ApApplication. A row for 'SpecificErrors' is selected and highlighted with a red box. The 'Value' column for this row contains '44'. The left sidebar shows navigation categories like DESIGN, SERIALIZATION CONFIG., and SOME/IP BINDING.

ApApplication Error Domains			ApApplication Errors			ApApplication Error Sets		
Error Domain	Namespace	Value	Error	Code	Description	Ref. Error Dor	Error Set	ApApplication
GenericErrors	ara::com::sample	45	GenericF...	12	Generic fail...	GenericErrors		
SpecificErrors	ara::com::sample	44						

# RadarService: Services 설정

- **Default Package** 변경

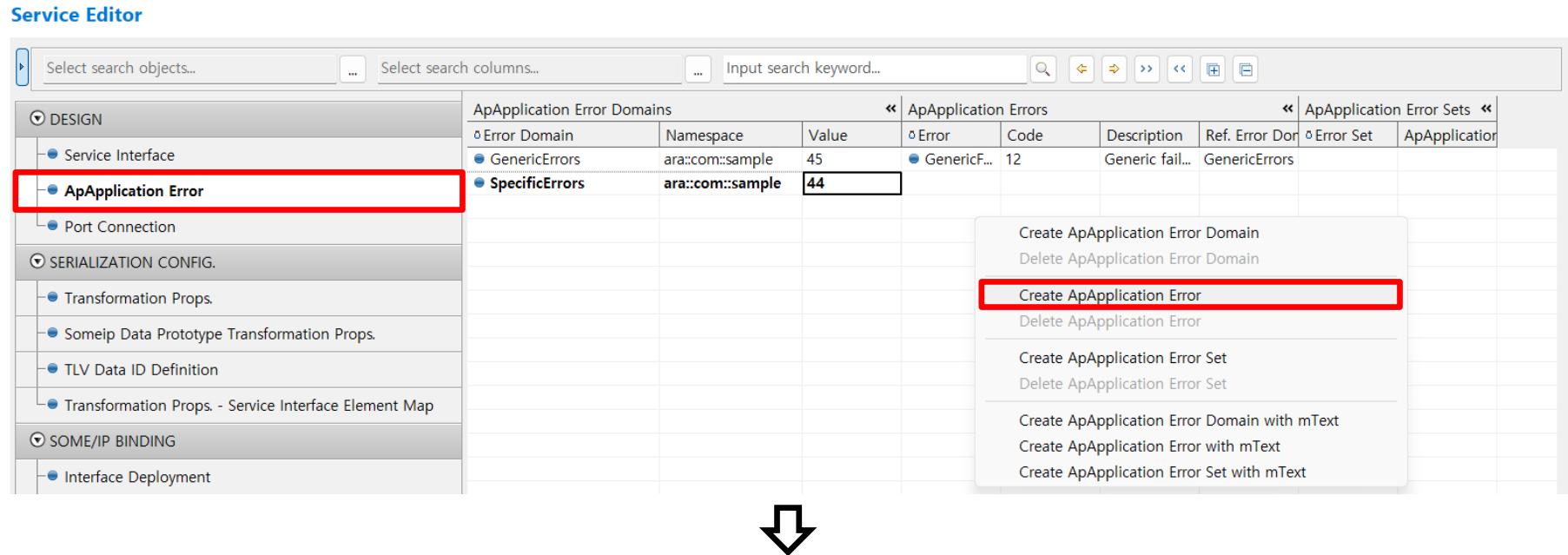
- ✓ 'ApAppError' Package를 Default Package로 설정하고 아래 창을 통해 확인함



# RadarService: Services 설정

## ■ ApApplication Error 설정 (1)

- ✓ 'ApApplication Error' 탭의 빈 곳에서 우클릭하여 'Create ApApplication Error'를 클릭함



ApApplication Error Domains			ApApplication Errors			ApApplication Error Sets		
Error Domain	Namespace	Value	Error	Code	Description	Ref. Error Dom	Error Set	ApApplication
GenericErrors	ara::com::sample	45	GenericFailure	12	Generic fail...	GenericErrors		
SpecificErrors	ara::com::sample	44						

# RadarService: Services 설정

## ▪ ApApplication Error 설정 (2)

- ✓ 생성된 ApApplication Error를 다음 그림과 같이 수정함

Service Editor

The screenshot shows the Service Editor interface with the 'DESIGN' tab selected. On the left, there is a tree view with nodes like 'Service Interface', 'ApApplication Error' (which is expanded), and 'Port Connection'. The main area displays two tables: 'ApApplication Error Domains' and 'ApApplication Errors'.

**ApApplication Error Domains:**

Error Domain	Namespace	Value
GenericErrors	ara::com::sample	45
SpecificErrors	ara::com::sample	44

**ApApplication Errors:**

Error	Code	Description	Ref. Error Domain
GenericFailure	12	Generic failure	GenericErrors
InvalidConfigString	5	Configuration string is invalid	SpecificErrors

The row for 'InvalidConfigString' is highlighted with a red box.



# RadarService: Services 설정

## ▪ ApApplication Error 설정 (3)

- ✓ ApApplication Error를 1개 더 추가하고, 다음 그림과 같이 수정함

Service Editor

The screenshot shows the Service Editor interface with the 'DESIGN' tab selected. On the left, there is a tree view with nodes like 'Service Interface', 'ApApplication Error' (which is expanded), and 'Port Connection'. The main area displays two tables: 'ApApplication Error Domains' and 'ApApplication Errors'.

**ApApplication Error Domains:**

Error Domain	Namespace	Value
GenericErrors	ara::com::sample	45
SpecificErrors	ara::com::sample	44

**ApApplication Errors:**

Error	Code	Description	Ref. Error Domain
GenericFailure	12	Generic failure	GenericErrors
InvalidConfigString	5	Configuration string is invalid	SpecificErrors
InvalidParameter	6	Parameter is invalid	SpecificErrors

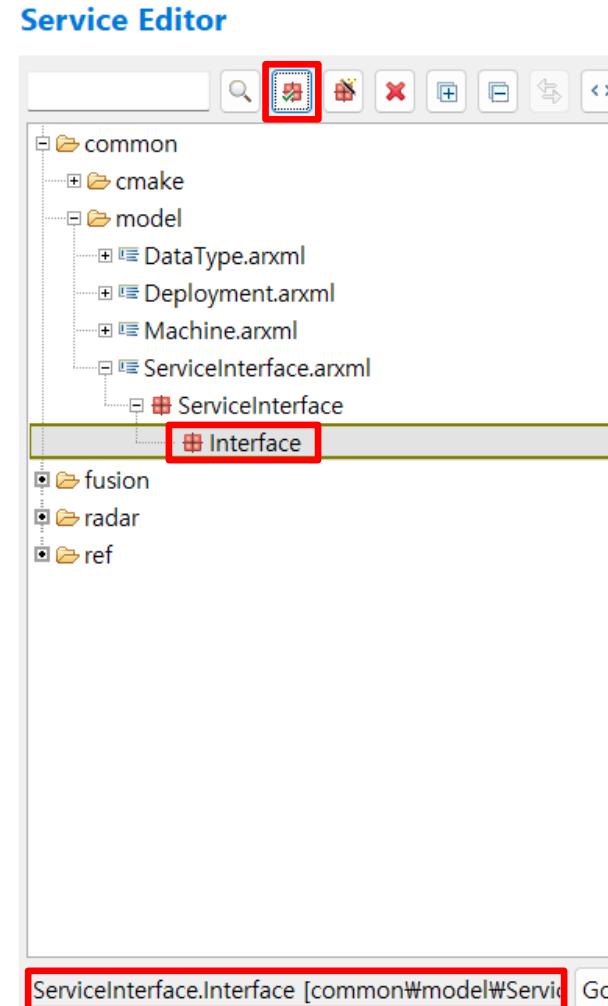
The last two rows of the 'ApApplication Errors' table (specifically 'InvalidConfigString' and 'InvalidParameter') are highlighted with a red box.



# RadarService: Services 설정

- **Default Package** 변경

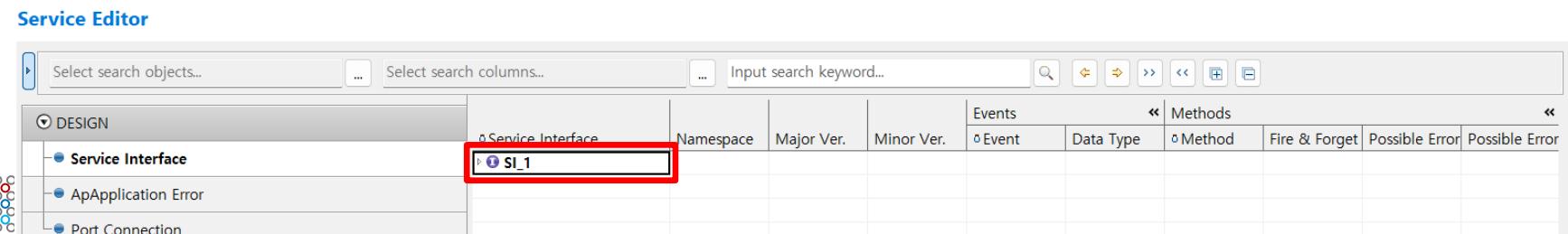
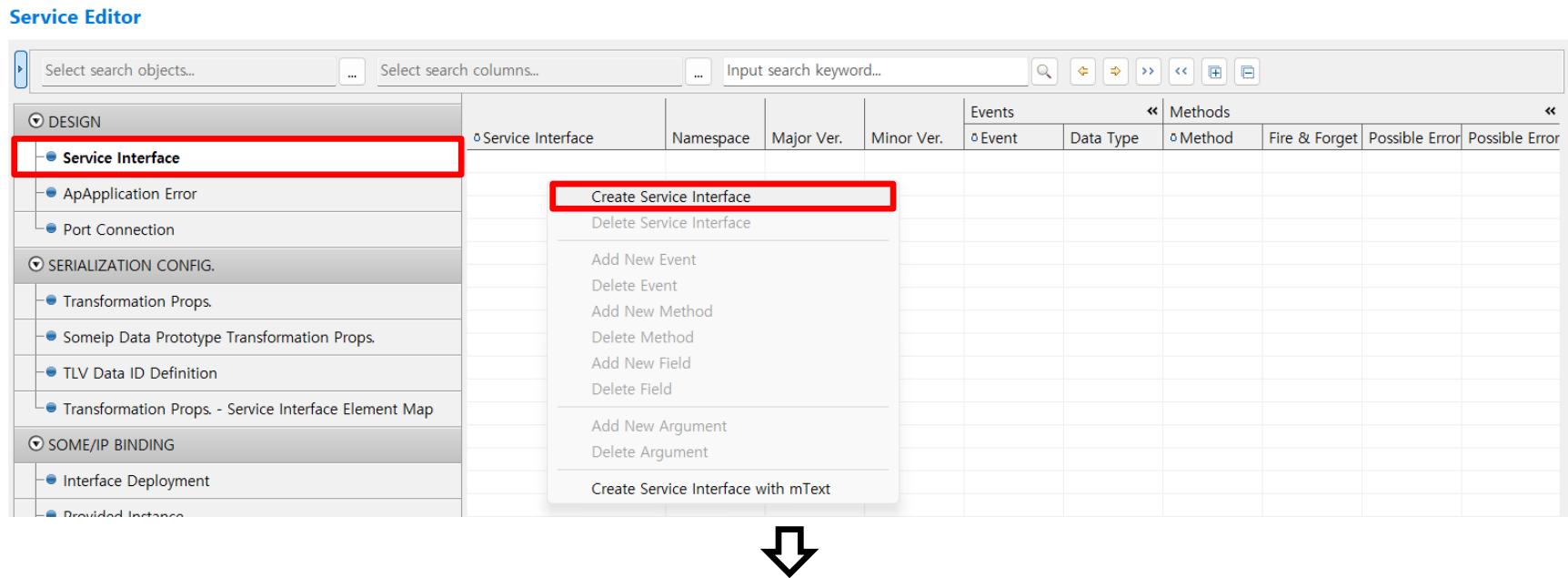
- ✓ 'Interface' Package를 Default Package로 설정하고 아래 창을 통해 확인함



# RadarService: Services 설정

## ■ Service Interface 설정 (1)

- ✓ 좌측의 'Service Interface' 탭으로 이동함
- ✓ 빈 곳에서 우클릭하여 'Create Service Interface'를 클릭함



# RadarService: Services 설정

## ▪ Service Interface 설정 (2)

- ✓ 생성한 Service Interface를 다음과 같이 수정함
  - ✓ Service Interface : 'radar'
  - ✓ Namespace : 'ara::com::sample'

Service Editor

The screenshot shows the Service Editor interface with the following details:

- DESIGN Tab:** Contains a table with columns: Service Interface, Namespace, Major Ver., Minor Ver., Events, and Methods.
- Service Interface Row:** The first row in the table is highlighted with a red box. It contains:
  - Service Interface: radar
  - Namespace: ara::com::sample
  - Events: Event
  - Methods: Method
- SERIALIZATION CONFIG. Tab:** Contains a list of items:
  - Transformation Props.
  - Someip Data Prototype Transformation Props.
  - TLV Data ID Definition
  - Transformation Props. - Service Interface Element Map
- SOME/IP BINDING Tab:** Contains a list of items:
  - Interface Deployment
  - Provided Instance
  - P-Instance Evt/Method/Evt Group
  - Required Instance

At the bottom left, there is a logo with the text "ACE Lab." and a page number "51/98" at the bottom right.

# RadarService: Services 설정

## ▪ Service Interface 설정 (3)

- ✓ 'radar'에서 우클릭하여 Event를 추가함

Service Editor

The screenshot shows the Service Editor interface. On the left, there's a tree view under 'DESIGN' with nodes like 'Service Interface', 'ApApplication Error', and 'Port Connection'. The 'Service Interface' node is selected. In the main area, a table lists 'Service Interface' entries. One entry for 'radar' is selected and highlighted with a red box. A context menu is open over this entry, also highlighted with a red box. The menu items include: Create Service Interface, Delete Service Interface, Add New Event (which is also highlighted with a red box), Delete Event, Add New Method, Delete Method, Add New Field, Delete Field, Add New Argument, Delete Argument, and Create Service Interface with mText. A large black arrow points downwards from the context menu towards the second screenshot.

Service Editor

This screenshot shows the same Service Editor interface after the new event has been added. The 'radar' service interface entry is now listed in the table, and its 'Events' column contains a single row with the value 'Event\_1', which is highlighted with a red box. The rest of the table structure remains the same as in the previous screenshot.



# RadarService: Services 설정

## ■ Service Interface 설정 (4)

- ✓ Event의 이름과 Data Type을 다음과 같이 수정함
  - ✓ Event : 'brakeEvent'
  - ✓ Data Type : 'RadarObjects'

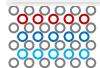
Service Editor

The screenshot shows the Service Editor interface with the following details:

- Left Sidebar (DESIGN):** Lists categories: Service Interface, Application Error, Port Connection, Serialization Config., Transformation Props., Someip Data Prototype Transformation Props., TLV Data ID Definition, Transformation Props. - Service Interface Element Map, and Some/IP Binding.
- Main View:** A table view showing Service Interface configurations.
  - Columns:** Service Interface, Namespace, Major Ver., Minor Ver., Events, Methods.
  - Rows:** One row is visible for the service interface "radar" located in the namespace "ara::com::sample".
  - Events Column:** Contains "brakeEvent".
  - Methods Column:** Contains "RadarObjects".

A red box highlights the "brakeEvent" and "RadarObjects" cells in the Events and Methods columns respectively, indicating they have been modified.

Service Interface	Namespace	Major Ver.	Minor Ver.	Events		Methods	
				Event	Data Type	Method	Fire & Forget
radar	ara::com::sample			brakeEvent	RadarObjects		



# RadarService: Services 설정

## ■ Service Interface 설정 (5)

- ✓ 동일한 방법으로 Event 한 개를 더 추가하고, 다음 그림과 같이 수정함

Service Editor

The screenshot shows the Service Editor interface. On the left, there is a sidebar with sections: DESIGN, SERIALIZATION CONFIG., and SOME/IP BINDING. Under DESIGN, 'Service Interface' is selected. In the main area, a table lists a service interface named 'radar' with namespace 'ara::com::sample'. The 'Events' column contains two entries: 'brakeEvent' and 'parkingBrakeEvent', both associated with the 'RadarObjects' data type. A red box highlights the 'parkingBrakeEvent' row. The table has columns for Service Interface, Namespace, Major Ver., Minor Ver., Events, Methods, Event, Data Type, Method, Fire & Forget, and Possible F.

Service Interface	Namespace	Major Ver.	Minor Ver.	Events	Methods	Event	Data Type	Method	Fire & Forget	Possible F
radar	ara::com::sample			brakeEvent	RadarObjects	parkingBrakeEvent	RadarObjects			

# RadarService: Services 설정

## ▪ Service Interface 설정 (6)

- ✓ 'radar'에서 우클릭하여 Method를 추가함

Service Editor

The screenshot shows the Service Editor interface. On the left, there's a tree view under 'DESIGN' with nodes like 'Service Interface', 'ApApplication Error', and 'Port Connection'. A context menu is open over the 'radar' node in the tree. The menu items are: 'Create Service Interface', 'Delete Service Interface', 'Add New Event', 'Delete Event', 'Add New Method' (which is highlighted with a red box), 'Delete Method', 'Add New Field', 'Delete Field', 'Add New Argument', 'Delete Argument', and 'Create Service Interface with mText'. A large black arrow points downwards from the menu towards the second screenshot.

Service Editor

This screenshot shows the Service Editor after the new method has been added. The 'radar' service interface now has a row in the 'Methods' table with the name 'Method\_1'. The 'Methods' table columns include 'Event', 'Data Type', 'Method', 'Fire & Forget', and 'Possible'. The 'Method' column for 'Method\_1' contains a green checkmark icon. A red box highlights this row.



# RadarService: Services 설정

## ▪ Service Interface 설정 (7)

- ✓ 추가한 Method의 이름을 'Adjust'로 수정함

Service Editor

The screenshot shows the Service Editor interface with the following details:

- DESIGN View:** Shows a tree view with nodes: Service Interface, radar, and several application error and port connection items.
- Table View:** Displays the configuration for the "radar" service interface.
  - Service Interface:** radar
  - Namespace:** ara::com::sample
  - Events:**
    - brakeEvent (Data Type: RadarObjects)
    - parkingBrakeEvent (Data Type: RadarObjects)
  - Methods:** A single method named "Adjust" is listed, highlighted with a red box.

# RadarService: Services 설정

## ■ Service Interface 설정 (8)

- ✓ 'Adjust'에서 우클릭하여 매개변수를 추가함

Service Editor

The screenshot shows the Service Editor interface with the 'DESIGN' tab selected. In the 'Events' table, there are two rows: 'brakeEvent' and 'parkingBrakeEvent', both with 'RadarObjects' as their data type. The 'Methods' column contains a single row for the 'Adjust' method. A context menu is open over the 'Adjust' row, with the 'Add New Argument' option highlighted. Other options in the menu include 'Create Service Interface', 'Delete Service Interface', 'Add New Event', 'Delete Event', 'Add New Method', 'Delete Method', 'Add New Field', 'Delete Field', and 'Create Service Interface with mText'. A large downward arrow points from the menu area to the second screenshot.

Service Editor

The screenshot shows the Service Editor interface with the 'DESIGN' tab selected. In the 'Events' table, there are two rows: 'brakeEvent' and 'parkingBrakeEvent', both with 'RadarObjects' as their data type. The 'Methods' column contains a single row for the 'Adjust' method. A new argument named 'Argument\_1' with type 'IN' has been added to the 'Adjust' method row. The argument is highlighted with a red box.



# RadarService: Services 설정

## ■ Service Interface 설정 (9)

- ✓ 매개변수의 이름, INPUT/OUTPUT, Data Type 설정을 수정함
- ✓ 다음 그림과 같이 2개의 매개변수를 더 추가하고 수정함

Service Editor

The screenshot shows the Service Editor interface with the following details:

- Left Sidebar (DESIGN):**
  - Service Interface
  - ApApplication Error
  - Port Connection
- Events Table:**

Event	Data Type	Method	Fire & Forget	Possible Error	Possible Error
brakeEvent	RadarObjects	Adjust			
parkingBrakeEvent	RadarObjects				
- Methods Table:**

Method	Fire & Forget	Possible Error	Possible Error	Argument	Dir	Data Type
Adjust				target_position	IN	Position
				success	OUT	bool
				effective_position	OUT	Position
- Methods: Args Table:**

Argument	Dir	Data Type
target_position	IN	Position
success	OUT	bool
effective_position	OUT	Position

# RadarService: Services 설정

## ■ Service Interface 설정 (10)

- ✓ 동일한 방법으로 4개의 Method를 추가하고 다음 그림과 같이 수정함
  - ✓ 'Echo' Method는 Fire & Forget 방식을 사용하도록 해당 설정을 체크함

Service Editor

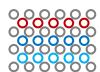
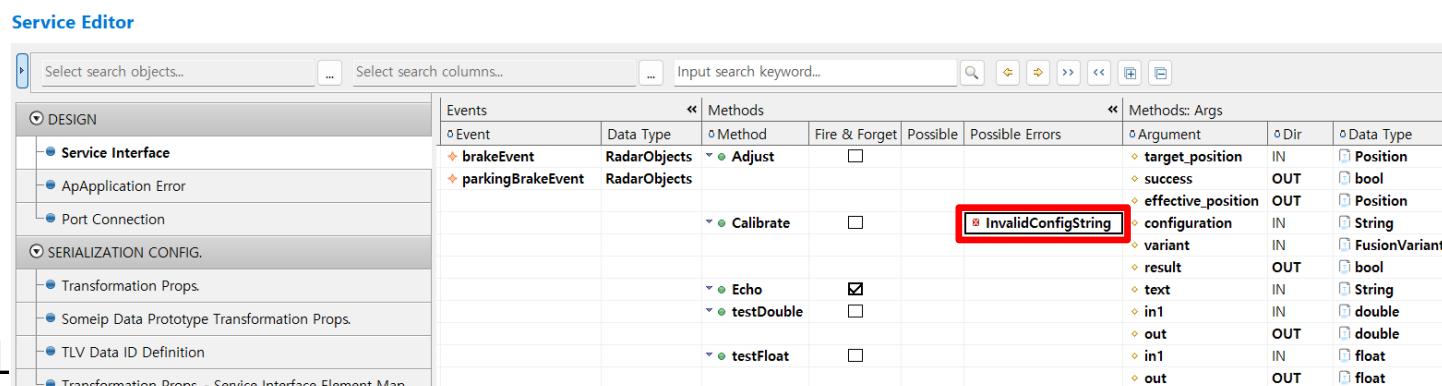
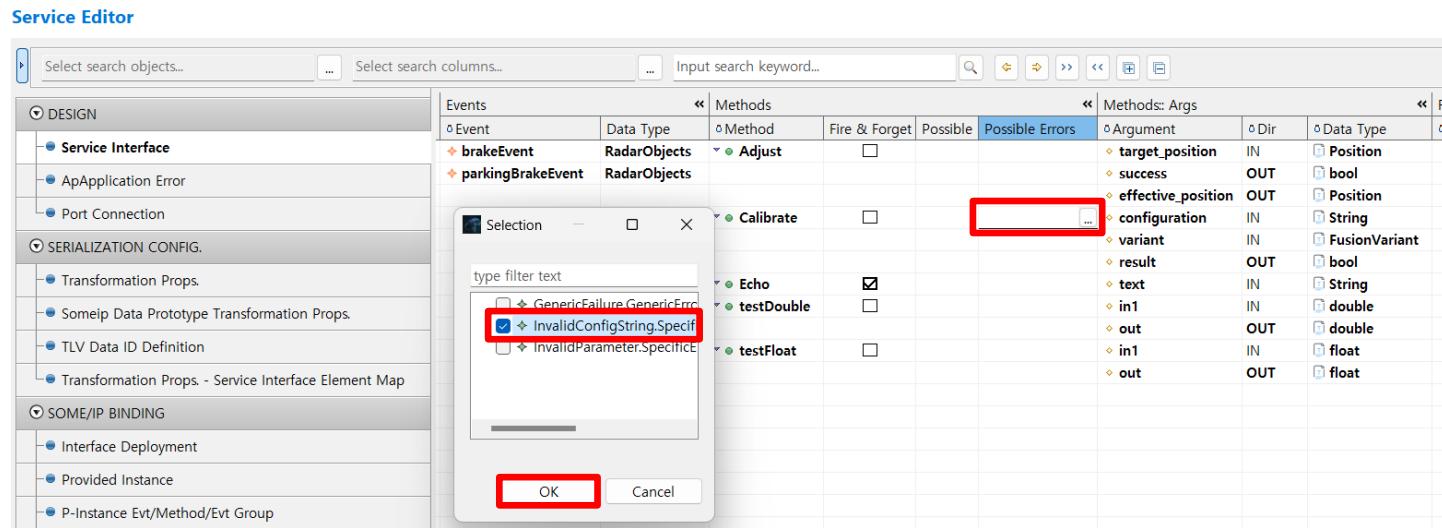
Events		Methods			Methods: Args		
Event	Data Type	Method	Fire & Forget	Possible Errors	Argument	Dir	Data Type
brakeEvent	RadarObjects	Adjust	<input type="checkbox"/>		target_position	IN	Position
parkingBrakeEvent	RadarObjects	Calibrate	<input type="checkbox"/>		success	OUT	bool
		Echo	<input checked="" type="checkbox"/>		effective_position	OUT	Position
		testDouble	<input type="checkbox"/>		configuration	IN	String
		testFloat	<input type="checkbox"/>		variant	IN	FusionVariant
					result	OUT	bool
					text	IN	String
					in1	IN	double
					out	OUT	double
					in1	IN	float
					out	OUT	float



# RadarService: Services 설정

## ■ Service Interface 설정 (11)

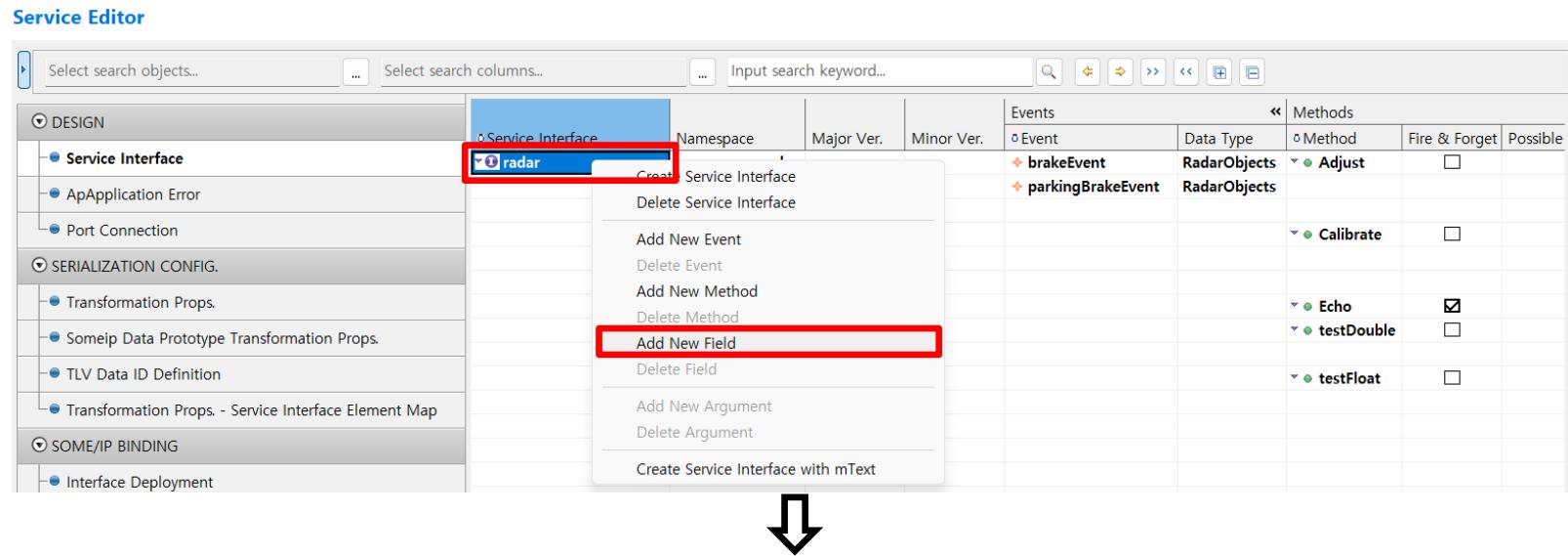
- ✓ 'Calibrate' Method는 정의한 에러값을 사용하도록 Possible Errors를 'InvalidConfigString'으로 수정함



# RadarService: Services 설정

## ■ Service Interface 설정 (12)

- ✓ 'radar'에서 우클릭하여 Field를 추가함



Service Editor

Methods				Methods: Args			Fields		
Method	Fire & Forget	Possible	Possible Errors	Argument	Dir	Data Type	Field	Data Type	Getter
Adjust	<input type="checkbox"/>			target_position	IN	Position	Field_1		<input type="checkbox"/>
Calibrate	<input type="checkbox"/>		InvalidConfigString	success	OUT	bool			
Echo	<input checked="" type="checkbox"/>			effective_position	OUT	Position			
testDouble	<input type="checkbox"/>			configuration	IN	String			
testFloat	<input type="checkbox"/>			variant	IN	FusionVariant			
				result	OUT	bool			
				text	IN	String			
				in1	IN	double			
				out	OUT	double			
				in1	IN	float			
				out	OUT	float			



# RadarService: Services 설정

## ▪ Service Interface 설정 (13)

- ✓ 추가한 Field의 이름과 Data Type, 그리고 Getter / Setter / Notifier 사용 여부를 수정함

Service Editor

Field	Data Type	Getter	Setter	Notifier
FrontObjectDistance	uint16_t	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>



# RadarService: Services 설정

## ■ Service Interface 설정 (14)

- ✓ 동일한 방법으로 3개의 Field를 더 추가하고, 다음 그림과 같이 수정함

Service Editor

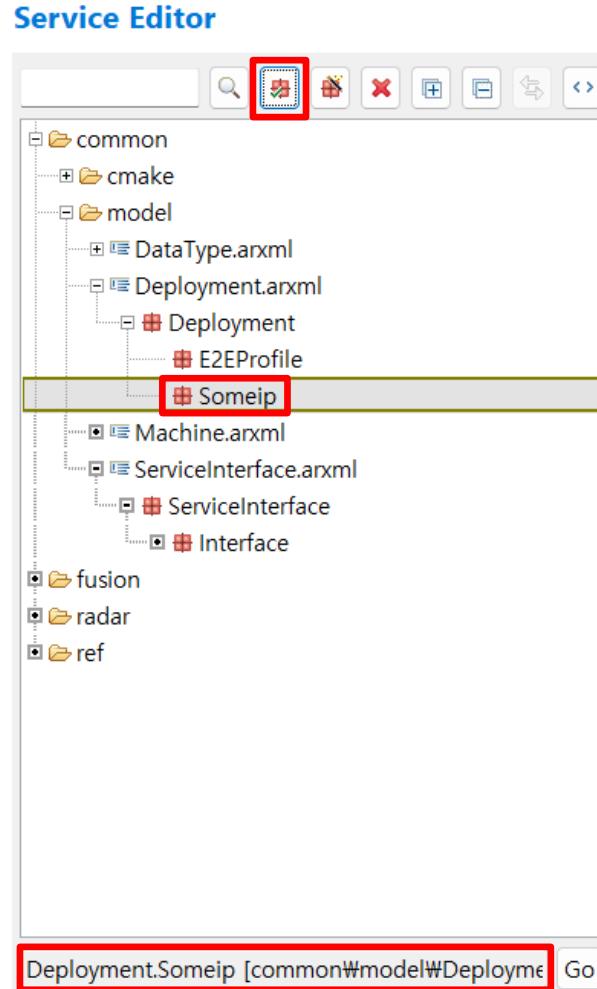
Role Errors	Argument	Dir	Data Type	Field	Data Type	Getter	Setter	Notifier
alidConfigString	◊ target_position	IN	Position	◊ FrontObjectDistance	uint16_t	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	◊ success	OUT	bool	◊ ObjectDetectionLimit	uint16_t	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	◊ effective_position	OUT	Position	◊ RearObjectDistance	uint16_t	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	◊ configuration	IN	String	◊ UpdateRate	uint32_t	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	◊ variant	IN	FusionVariant					
	◊ result	OUT	bool					
	◊ text	IN	String					
	◊ in1	IN	double					
	◊ out	OUT	double					
	◊ in1	IN	float					
	◊ out	OUT	float					



# RadarService: Services 설정

- **Default Package** 변경

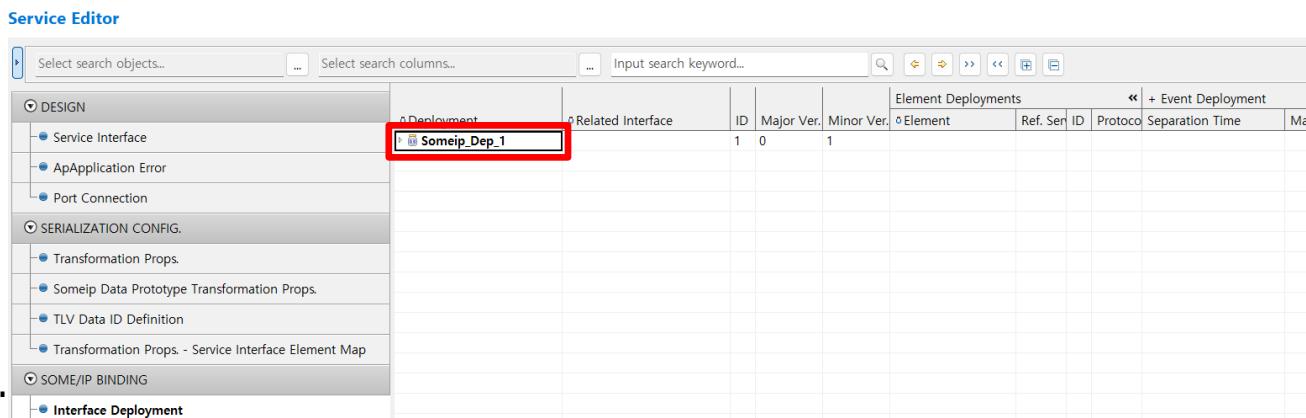
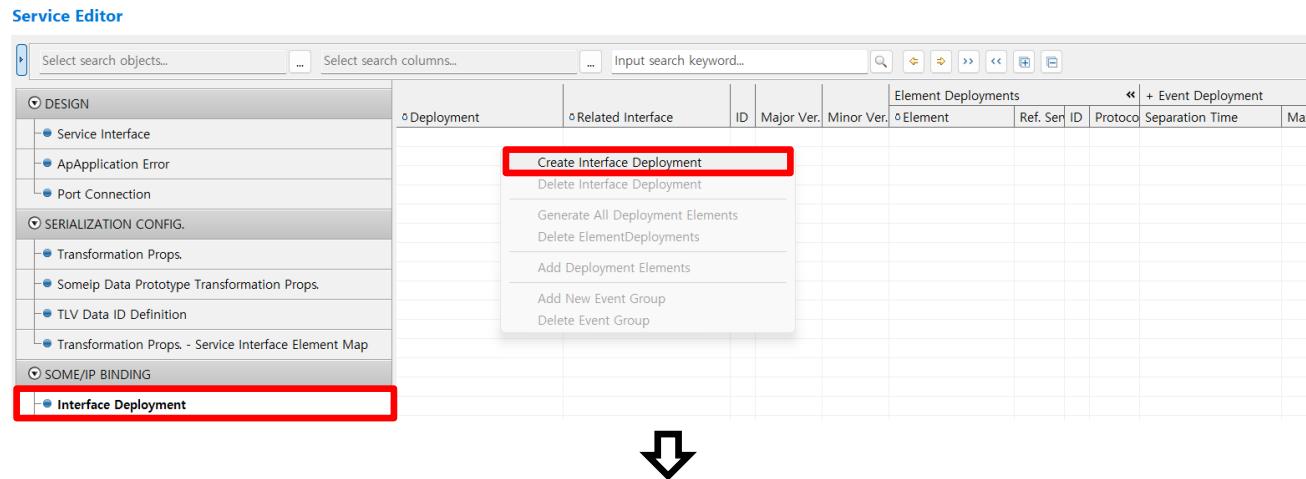
- ✓ 'Someip' Package를 Default Package로 설정하고 아래 창을 통해 확인함



# RadarService: Services 설정

## ▪ Interface Deployment 설정 (1)

- ✓ 좌측의 'Interface Deployment' 탭으로 이동함
- ✓ 빈 곳에서 우클릭하여 Interface Deployment를 생성함



# RadarService: Services 설정

## ▪ Interface Deployment 설정 (2)

- ✓ 생성한 Interface Deployment의 이름, 인터페이스, ID, Version을 다음 그림과 같이 수정함

Service Editor

The screenshot shows the Service Editor interface with the 'DESIGN' tab selected. On the left, there is a tree view under 'DESIGN' containing 'Service Interface', 'ApApplication Error', and 'Port Connection'. Under 'SOME/IP BINDING', 'Interface Deployment' is expanded, showing 'Provided Instance', 'P-Instance Evt/Method/Evt Group', and 'Required Instance'. The main area displays a table titled 'Element Deployments' with the following columns: Deployment, Related Interface, ID, Major Ver., Minor Ver., Element, Ref. Ser, ID, Protocol, Separation Time, and Major. A row for 'Someip\_radarDep\_1' is selected and highlighted with a red box. The 'Related Interface' column shows 'radar', 'ID' is '99', 'Major Ver.' is '1', and 'Minor Ver.' is '0'. The 'Element' column contains a small icon.

Deployment	Related Interface	ID	Major Ver.	Minor Ver.	Element	Ref. Ser	ID	Protocol	Separation Time	Major
Someip_radarDep_1	radar	99	1	0	(Icon)					

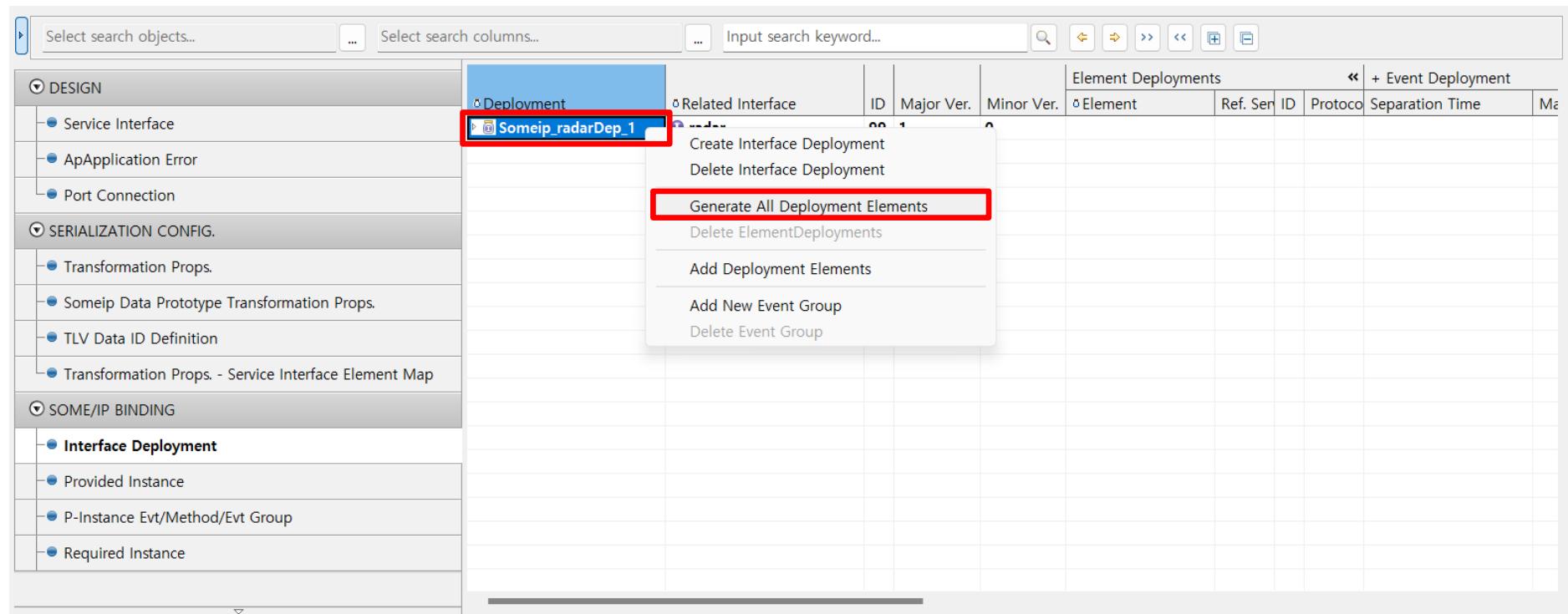


# RadarService: Services 설정

## ▪ Interface Deployment 설정 (3)

- ✓ Service Interface 정보로부터 Interface Deployment 설정을 자동 생성하기 위해 'Someip\_radarDep\_1'에서 우클릭하여 'Generate All Deployment Elements'를 클릭함

Service Editor

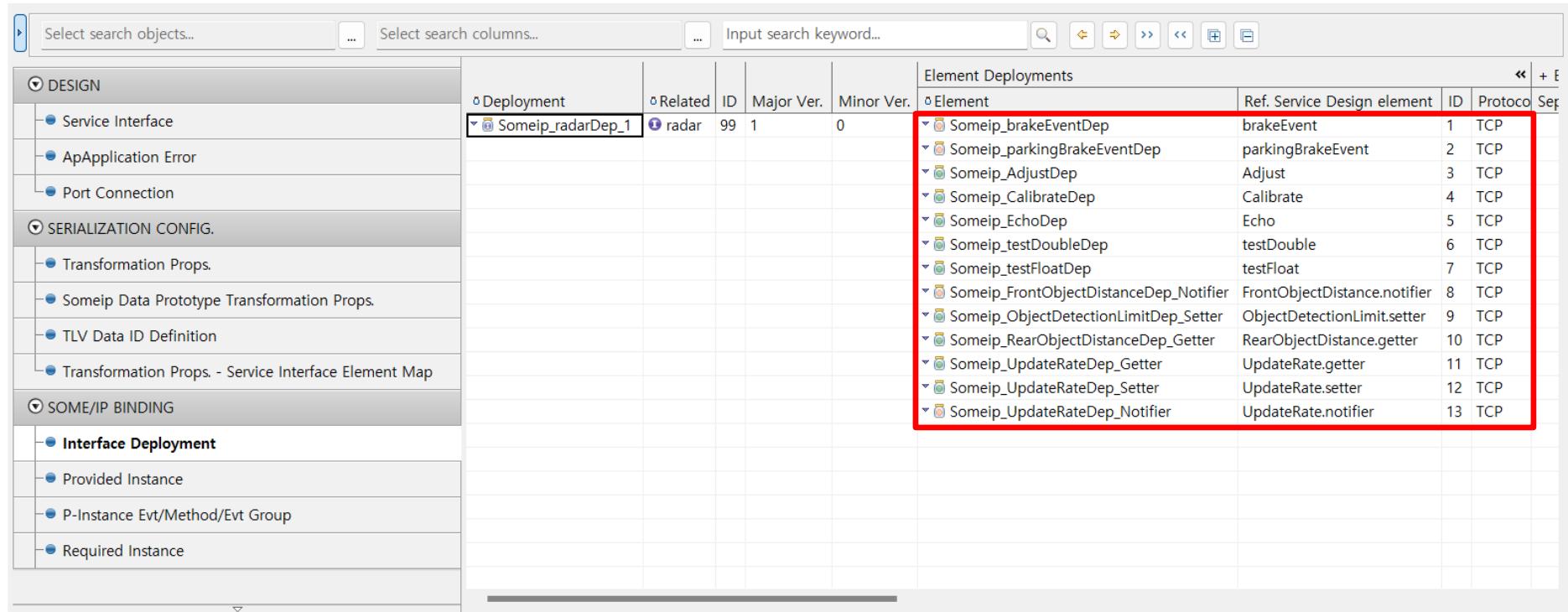


# RadarService: Services 설정

## ▪ Interface Deployment 설정 (4)

- ✓ Service Interface 정보를 바탕으로 자동으로 Deployment가 수행된 것을 확인함

Service Editor



Element	Ref. Service Design element	ID	Protocol	Seq.
Someip_brakeEventDep	brakeEvent	1	TCP	
Someip_parkingBrakeEventDep	parkingBrakeEvent	2	TCP	
Someip_AdjustDep	Adjust	3	TCP	
Someip_CalibrateDep	Calibrate	4	TCP	
Someip_EchoDep	Echo	5	TCP	
Someip_testDoubleDep	testDouble	6	TCP	
Someip_testFloatDep	testFloat	7	TCP	
Someip_FrontObjectDistanceDep_Notifier	FrontObjectDistance.notifier	8	TCP	
Someip_ObjectDetectionLimitDep_Setter	ObjectDetectionLimit.setter	9	TCP	
Someip_RearObjectDistanceDep_Getter	RearObjectDistance.getter	10	TCP	
Someip_UpdateRateDep_Getter	UpdateRate.getter	11	TCP	
Someip_UpdateRateDep_Setter	UpdateRate.setter	12	TCP	
Someip_UpdateRateDep_Notifier	UpdateRate.notifier	13	TCP	



# RadarService: Services 설정

## ▪ Interface Deployment 설정 (5)

- ✓ 자동 생성된 Deployment 설정에서 수정이 필요한 부분을 다음 그림과 같이 수정함

Service Editor

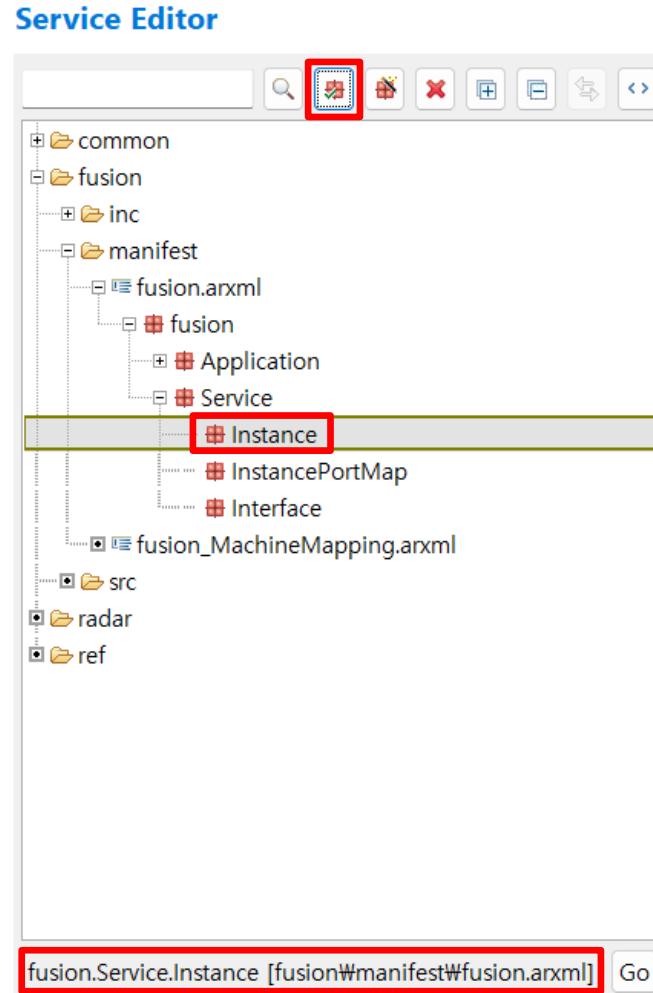
Element Deployments						
Element	Ref. Service Design element	ID	Protocol	Separation Time	Max. Segment Length	Serializer
Someip_brakeEventDep	brakeEvent	1	UDP	0.0	0	SOMEIP
Someip_parkingBrakeEventDep	parkingBrakeEvent	2	UDP	0.0	0	SOMEIP
Someip_AdjustDep	Adjust	32000	UDP			
Someip_CalibrateDep	Calibrate	32010	UDP			
Someip_EchoDep	Echo	32020	UDP			
Someip_testDoubleDep	testDouble	32022	UDP			
Someip_testFloatDep	testFloat	32021	UDP			
Someip_FrontObjectDistanceDep_Notifier	FrontObjectDistance.notifier	3004	UDP	0.0	0	SOMEIP
Someip_ObjectDetectionLimitDep_Setter	ObjectDetectionLimit.setter	3011	TCP			
Someip_RearObjectDistanceDep_Getter	RearObjectDistance.getter	3006	TCP			
Someip_UpdateRateDep_Getter	UpdateRate.getter	3001	TCP			
Someip_UpdateRateDep_Setter	UpdateRate.setter	3002	TCP			
Someip_UpdateRateDep_Notifier	UpdateRate.notifier	3000	UDP	0.0	0	SOMEIP



# Fusion: Services 설정

- **Default Package** 변경

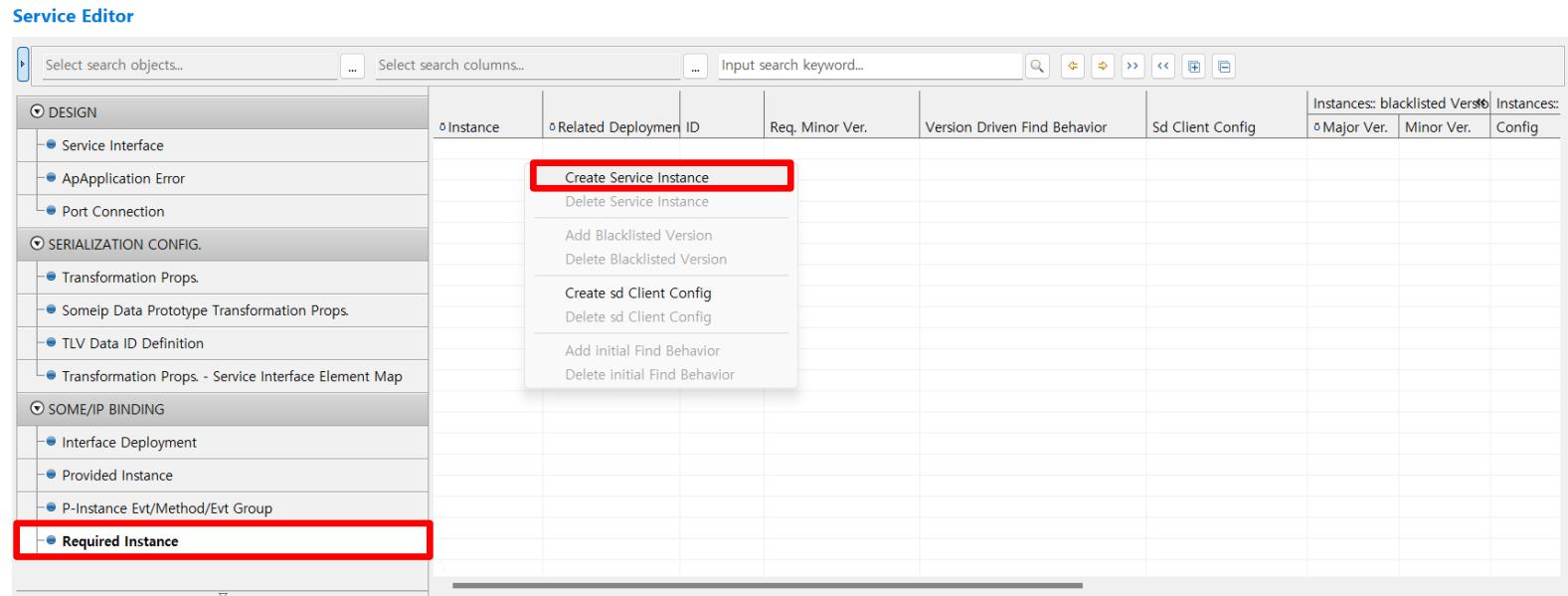
- ✓ 'Instance' Package를 Default Package로 설정하고 아래 창을 통해 확인함



# Fusion: Services 설정

## ▪ Required Instance 설정 (1)

- ✓ 좌측의 'Required Instance' 탭으로 이동함
- ✓ 빈 곳에서 우클릭하여 Service Instance를 생성함



The screenshot shows the Service Editor after creating a new instance. The 'Required Instance' row in the table is highlighted with a red box. The 'Instance' column contains 'Someip\_R\_Inst\_1', 'Related Deployment ID' is 'ANY', 'Req. Minor Ver.' is '0', and 'Version Driven Find Behavior' is 'MINIMUM-MINOR-VERSION'. The rest of the table structure is identical to the previous screenshot.

Instance	Related Deployment ID	Req. Minor Ver.	Version Driven Find Behavior	Sd Client Config	Instances: blacklisted Ver	Instances: Config	Major Ver.	Minor Ver.	Config
Someip_R_Inst_1	ANY	0	MINIMUM-MINOR-VERSION						
Service Interface									
ApApplication Error									
Port Connection									

# Fusion: Services 설정

## ▪ Required Instance 설정 (2)

- ✓ 생성된 Service Instance를 다음과 같이 수정함
  - ✓ Instance : 'radar\_RequiredSomeipServiceInstance'
  - ✓ Related Deployment : 'Someip\_radarDep\_1'

Service Editor

The screenshot shows the Service Editor interface with the 'Required Instances' table highlighted by a red box. The table has columns for Instance, Related Deployment, ID, Req. Minor Ver., Version Driven Find Behavior, Sd Client Config, and Instance Majc. The selected row contains the values: Instance: 'radar\_RequiredSomeipServiceInstance', Related Deployment: 'Someip\_radarDep\_1', ID: ANY, Req. Minor Ver: 0, Version Driven Find Behavior: MINIMUM-MINOR-VERSION, Sd Client Config: (empty), and Instance Majc: (empty). The left sidebar shows categories like DESIGN, SERIALIZATION CONFIG., and SOME/IP BINDING, with 'Required Instance' currently selected.

Select search objects...	Select search columns...	Input search keyword...	Search icons				
DESIGN	Instance	Related Deployment	ID	Req. Minor Ver.	Version Driven Find Behavior	Sd Client Config	Instance Majc
<ul style="list-style-type: none"><li>Service Interface</li><li>ApApplication Error</li><li>Port Connection</li></ul>	radar_RequiredSomeipServiceInstance	Someip_radarDep_1	ANY	0	MINIMUM-MINOR-VERSION		

# Fusion: Services 설정

## ▪ Required Instance 설정 (3)

- ✓ 빈 곳에서 우클릭하여 SD Client Config를 생성함

Service Editor

The screenshot shows the Service Editor interface. On the left, there's a tree view under 'DESIGN' with nodes like 'Service Interface', 'ApApplication Error', 'Port Connection', 'SERIALIZATION CONFIG.', 'Transformation Props.', 'Someip Data Prototype Transformation Props.', 'TLV Data ID Definition', and 'Transformation Props. - Service Interface Element Map'. In the main table, there's one row for 'radar\_RequiredSomeipServiceInstance' with columns for 'Instance' (containing 'radar\_RequiredSomeipServiceInstance'), 'Related Deployment' (containing 'Someip\_radarDep\_1'), 'ID' (containing 'ANY'), 'Req. Minor Ver.' (containing '0'), 'Version Driven Find Behavior' (containing 'MINIMUM-MINOR-VERSION'), and 'Sd Client Config' (containing 'None'). A context menu is open over the 'radar\_RequiredSomeipServiceInstance' row, with the 'Create sd Client Config' option highlighted by a red box.



Service Editor

This screenshot shows the same Service Editor interface after the 'Create sd Client Config' action was performed. The 'Sd Client Config' column now contains 'ClientSdConfig\_1', which is also highlighted by a red box. The rest of the table and interface elements remain the same as in the previous screenshot.



# Fusion: Services 설정

## ■ Required Instance 설정 (4)

- ✓ 생성한 SD Client Config를 다음과 같이 수정함
  - ✓ Config : 'radar\_SomeipClientServiceInstanceConfig'
  - ✓ Service Find TTL : '10'

Service Editor

The screenshot shows the Service Editor interface with the 'Required Instance' tab selected. On the left, there is a tree view with categories like DESIGN, SERIALIZATION CONFIG., and SOME/IP BINDING. Under DESIGN, 'Service Interface' is expanded. In the main area, a table titled 'Required Instances: sd Client Config' is displayed. The table has columns: 'Inor. Config', 'Service Find TTL', 'Delay Max.', 'Delay Min.', 'Repetition Max.', and 'Repetitions Base Delay'. A single row is present, showing 'radar\_SomeipClientServiceInstanceConfig' in the 'Inor. Config' column and '10' in the 'Service Find TTL' column. The 'Service Find TTL' cell is highlighted with a red box.

Inor. Config	Service Find TTL	Delay Max.	Delay Min.	Repetition Max.	Repetitions Base Delay
radar_SomeipClientServiceInstanceConfig	10				

# Fusion: Services 설정

## ▪ Required Instance 설정 (5)

- ✓ 'radar\_SomeipClientServiceInstanceConfig'에서 우클릭하여 Initial Find Behavior을 추가함

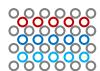
Service Editor

The screenshot shows the Service Editor interface. On the left, there's a tree view under 'DESIGN' with nodes like 'Service Interface', 'ApApplication Error', 'Port Connection', and 'SERIALIZATION CONFIG.' with sub-nodes such as 'Transformation Props.', 'Someip Data Prototype Transformation Props.', 'TLV Data ID Definition', and 'Transformation Props. - Service Interface Element Map'. In the main area, a table lists service instances. One row for 'radar\_SomeipClientServiceInstanceConfig' is selected and highlighted with a red box. A context menu is open over this row, also outlined in red. The menu items are: 'Create Service Instance', 'Delete Service Instance', 'Add Blacklisted Version', 'Delete Blacklisted Version', 'Create sd Client Config', 'Delete sd Client Config', 'Add initial Find Behavior' (which is highlighted with a red box), and 'Delete initial Find Behavior'. At the bottom right of the menu, there's a small note: 'Delay Max. Delay Min. Repetition Max. Repetitions Base Delay'.



Service Editor

This screenshot shows the same Service Editor interface after the configuration has been updated. The 'radar\_SomeipClientServiceInstanceConfig' row in the table now has an additional column labeled '+initial Find Behavior'. This column contains a single row with the value '10'. Below this row, there are four input fields: 'Delay Max.' (set to 10.0), 'Delay Min.' (set to 10.0), 'Repetition Max.' (set to 10), and 'Repetitions Base Delay' (set to 10.0). All these fields are also outlined in red.



# Fusion: Services 설정

## ▪ Required Instance 설정 (6)

- ✓ 추가된 Initial Find Behavior을 다음 그림과 같이 수정함

Service Editor

The screenshot shows the Service Editor interface. On the left, there's a tree view with categories like DESIGN, SERIALIZATION CONFIG., and SOME/IP BINDING. Under DESIGN, 'Required Instance' is selected. In the main pane, there's a table titled 'Initial Find Behavior' with the following data:

Minor Config	Service Find TTL	Delay Max.	Delay Min.	Repetition Max.	Repetitions Base Delay
radar_SomeipClientServiceInstanceConfig	10	0.1	0.01	3	0.2

# Fusion: Services 설정

## ▪ Required Instance 설정 (7)

- ✓ 앞서 생성한 SD Client Config를 'radar\_RequiredSomeipServiceInstance'에 연결함

Service Editor

The screenshot shows the Service Editor interface with the 'DESIGN' tab selected. On the left, there is a sidebar with categories: DESIGN, SERIALIZATION CONFIG., and SOME/IP BINDING. Under DESIGN, 'Required Instance' is selected. The main area is a table titled 'Required Instance' with the following columns: Instance, Related Deployment, ID, Req. N, Version Driven Find Behavior, and Sd Client Config. A single row is present in the table:

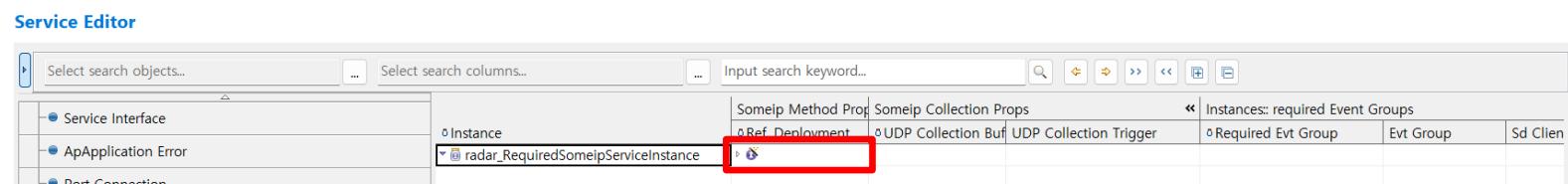
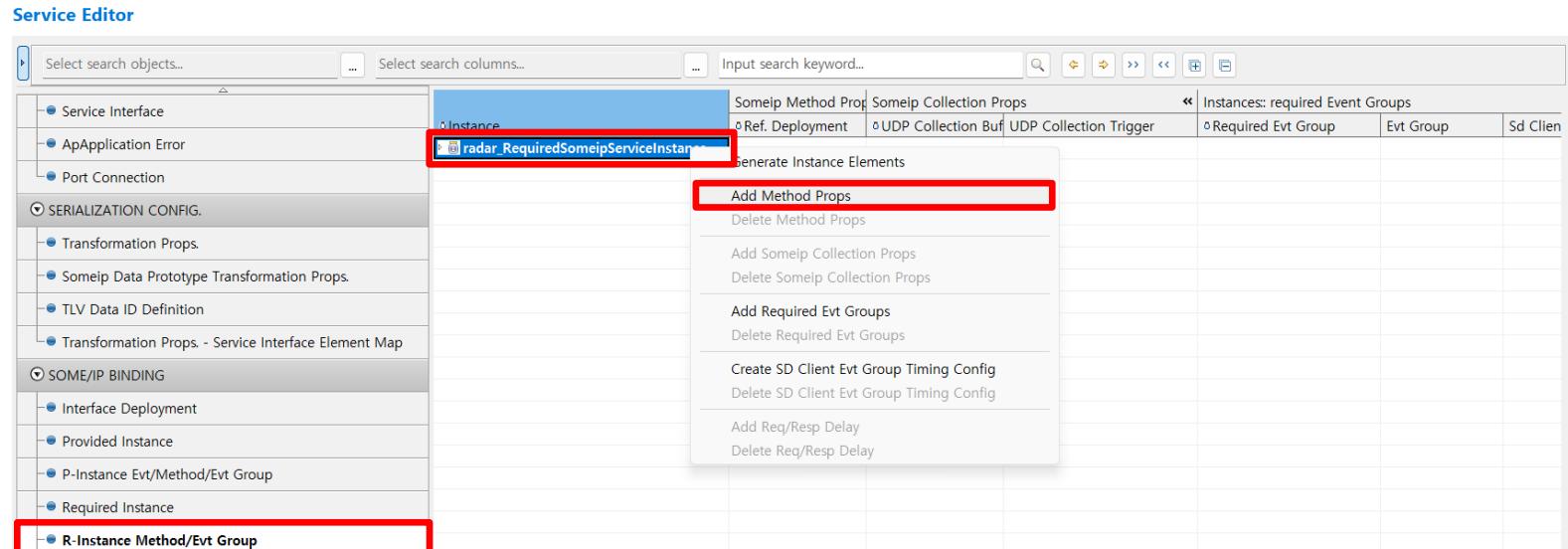
Instance	Related Deployment	ID	Req. N	Version Driven Find Behavior	Sd Client Config
radar_RequiredSomeipServiceInstance	Someip_radarDep_1	ANY	0	MINIMUM-MINOR-VERSION	radar_SomeipClientServiceInstanceConfig



# Fusion: Services 설정

## ■ Required Instance 설정 (8)

- ✓ 좌측의 'R-Instance Method/Evt Group' 탭으로 이동함
- ✓ 'radar\_RequiredSomeipServiceInstance'에서 우클릭하여 Method Props를 추가함



# Fusion: Services 설정

## ▪ Required Instance 설정 (9)

- ✓ 추가된 Method Props를 'Someip\_ObjectDetectionLimitDep\_Setter'로 수정함
- ✓ 동일한 방법으로 Method Props를 3개 더 추가하고, 다음 그림과 같이 수정함

Service Editor

Select search objects...	Select search columns...	Input search keyword...	Ref. Deployment	Someip Collection Props	Instance
Interface Deployment					
Provided Instance					

Instance

Someip Method Props	Someip Collection Props	Instance
Someip_ObjectDetectionLimitDep_Setter		

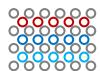


Service Editor

Select search objects...	Select search columns...	Input search keyword...	Ref. Deployment	Someip Collection Props	Instance
Interface Deployment					
Provided Instance					
P-Instance Evt/Method/Evt Group					
Required Instance					
R-Instance Method/Evt Group					
Instance-Machine Map					
Instance-Port Map					

Instance

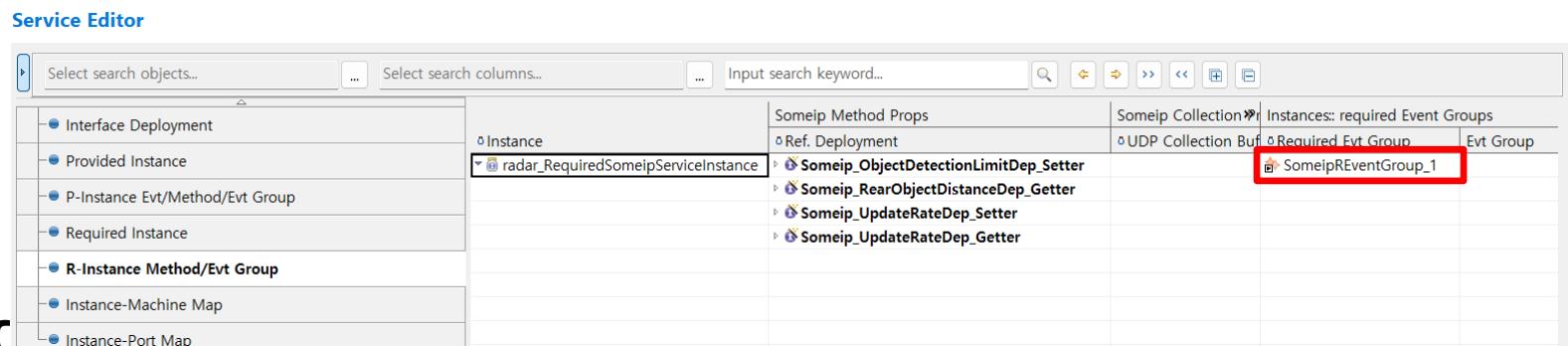
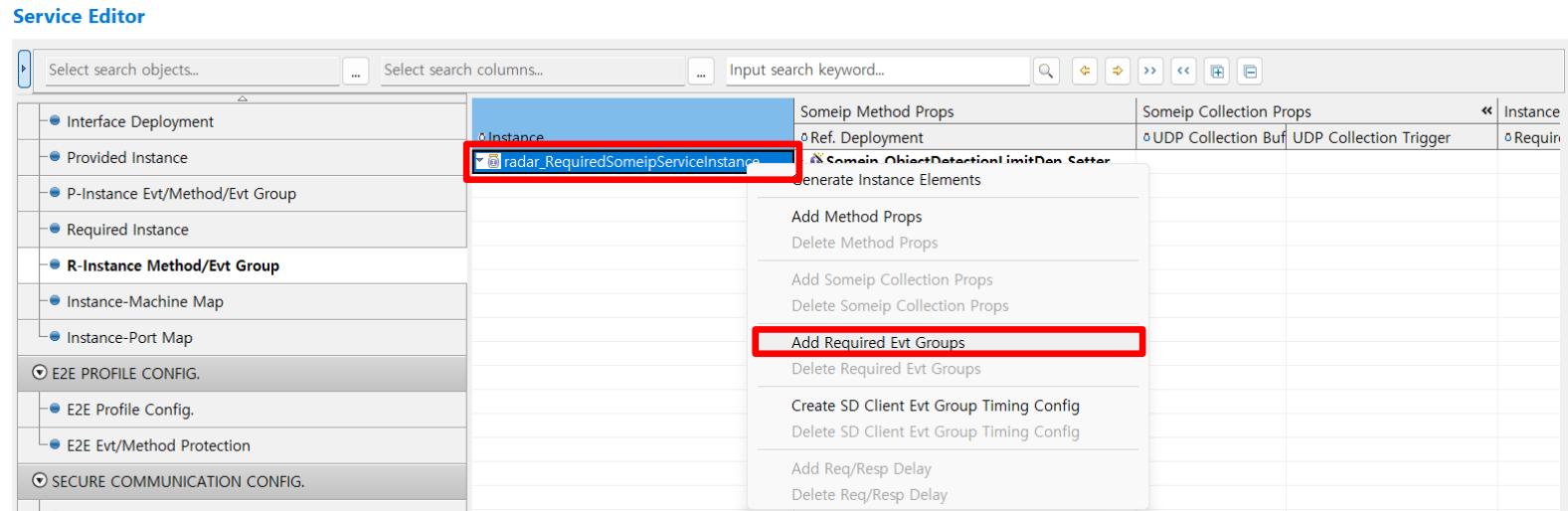
Someip Method Props	Someip Collection Props	Instance
Someip_ObjectDetectionLimitDep_Setter		
Someip_RearObjectDistanceDep_Getter		
Someip_UpdateRateDep_Setter		
Someip_UpdateRateDep_Getter		



# Fusion: Services 설정

## ▪ Required Instance 설정 (10)

- ✓ 'radar\_RequiredSomeipServiceInstance'에서 우클릭하여 Required Event Groups를 추가함



# Fusion: Services 설정

## ▪ Required Instance 설정 (11)

- ✓ 생성된 Required Event Group의 Event Group을 'EventGroup\_1'으로 수정함

Service Editor

The screenshot shows the Service Editor interface with the following details:

- Left Sidebar:** A tree view of configuration categories:
  - Interface Deployment
  - Provided Instance
  - P-Instance Evt/Method/Evt Group
  - Required Instance
  - R-Instance Method/Evt Group
  - Instance-Machine Map
  - Instance-Port Map
  - E2E PROFILE CONFIG.** (selected)
    - E2E Profile Config.
    - E2E Evt/Method Protection
- SECURE COMMUNICATION CONFIG.** (selected)

  - TLS Secure Com. Props.
  - Crypto Cipher Suite.PSK Identity
  - Instance-Machine Map for Secure Com.

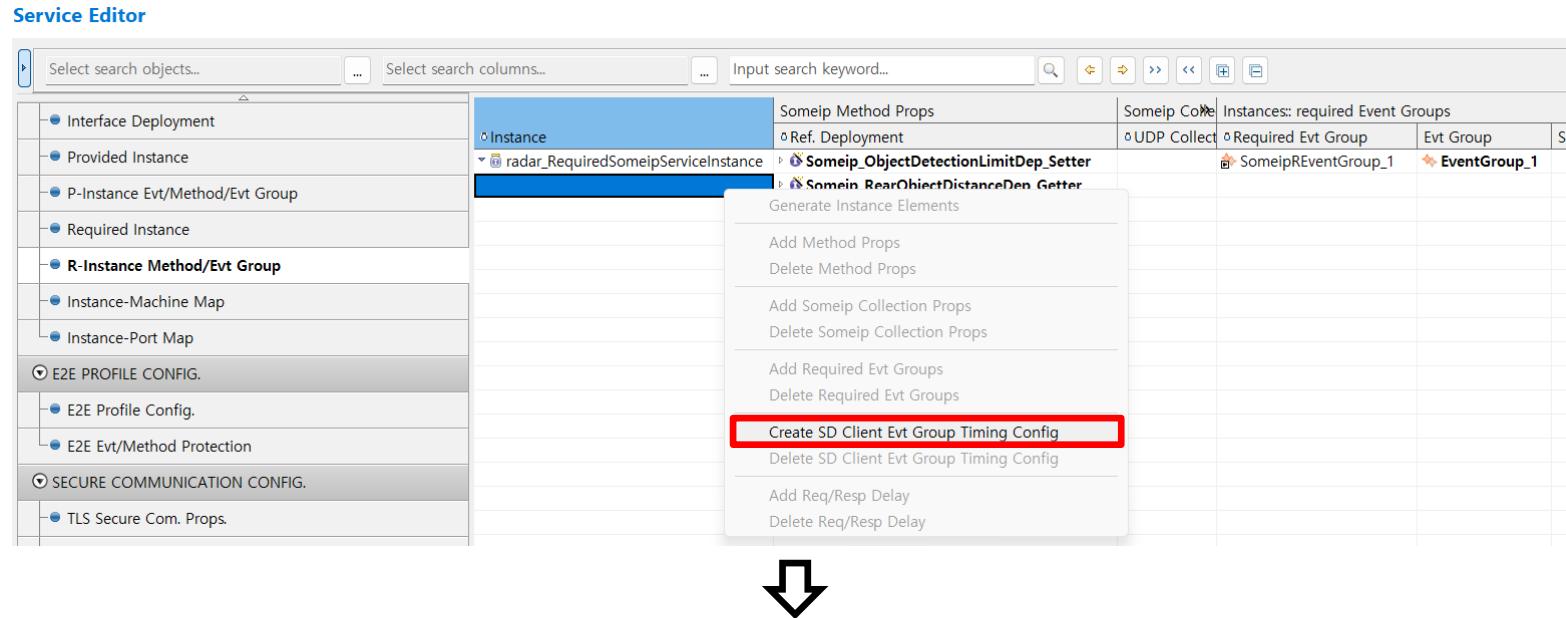
- Main View:** A table with columns for search, instance type, properties, deployment, and event groups.

Select search objects...	Select search columns...	Input search keyword...	Someip Method Props	Someip Core	Instances: required Event Groups		
...	...	...	Ref. Deployment	UDP Collect	Required Evt Group	Evt Group	...
					SomeipREventGroup_1	EventGroup_1	
	Instance	radar_RequiredSomeipServiceInstance			Someip_ObjectDetectionLimitDep_Setter		
					Someip_RearObjectDistanceDep_Getter		
					Someip_UpdateRateDep_Setter		
					Someip_UpdateRateDep_Getter		

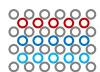

# Fusion: Services 설정

## ▪ Required Instance 설정 (12)

- ✓ 빈 곳에서 우클릭하여 SD Client Event Group Timing Config를 생성함



Sd Client Event Group Timing Config				
Sd Client Evt Group Timing Conf.	TTL	Subscribe Evt Group Retry Delay	Subscribe Evt Group Retry Max.	+ Req/Resp Delay
client_timingConf_1	0	10	10	



# Fusion: Services 설정

## ▪ Required Instance 설정 (13)

- ✓ 생성된 SD Client Event Group Timing Config에서 우클릭하여 Req/Resp Delay를 추가함

Service Editor

The screenshot shows the Service Editor interface. On the left is a tree view with categories like Interface Deployment, Provided Instance, P-Instance Evt/Method/Evt Group, Required Instance, R-Instance Method/Evt Group, Instance-Machine Map, Instance-Port Map, E2E PROFILE CONFIG., E2E Profile Config., E2E Evt/Method Protection, SECURE COMMUNICATION CONFIG., and TIS Secure Com. Prop. A specific entry, 'client\_timingConf\_1', is selected and highlighted with a red box. A context menu is open over this entry, also highlighted with a red box. The menu items are: Add Method Props, Delete Method Props, Add Someip Collection Props, Delete Someip Collection Props, Add Required Evt Groups, Delete Required Evt Groups, Create SD Client Evt Group Timing Config, Delete SD Client Evt Group Timing Config, Add Req/Resp Delay (which is also highlighted with a red box), and Delete Req/Resp Delay.



Service Editor

The screenshot shows the Service Editor interface again. The tree view on the left is identical to the previous one. The main area now displays a table for 'Sd Client Event Group Timing Config'. It has columns: Sd Client Evt Group Timing Conf., TTL, Subscribe Evt Group Retry Delay, and + Req/Resp Delay (Max. and Min.). The row for 'client\_timingConf\_1' is selected and highlighted with a red box. The 'Req/Resp Delay' column shows values 10.0 for Max. and 10.0 for Min., which are also highlighted with a red box.



# Fusion: Services 설정

## ▪ Required Instance 설정 (14)

- ✓ 앞서 생성한 SD Client Event Group Timing Config을 Required Event Group에 연결함

Service Editor

The screenshot shows the Service Editor interface with the following details:

- Left Sidebar:** A tree view of service configurations, including Interface Deployment, Provided Instance, P-Instance Evt/Method/Evt Group, Required Instance, R-Instance Method/Evt Group, Instance-Machine Map, Instance-Port Map, E2E PROFILE CONFIG. (selected), E2E Profile Config., E2E Evt/Method Protection, and SECURE COMMUNICATION CONFIG. (selected), which includes TLS Secure Com. Props., Crypto Cipher Suite.PSK Identity, and Instance-Machine Map for Secure Com.
- Top Bar:** Search fields for "Select search objects...", "Select search columns...", and "Input search keyword..." along with navigation icons.
- Central View:** A table titled "Instances:: required Event Groups". It lists two entries:
  - Required Evt Group: SomeipREventGroup\_1, Evt Group: EventGroup\_1, Sd Client Evt Group Timing Conf.: client\_timingConf\_1
  - Required Evt Group: EventGroup\_1, Evt Group: EventGroup\_1, Sd Client Evt Group Timing Conf.: client\_timingConf\_1
- Right Panel:** A detailed view of the "Sd Client Event Group Timing Config" for the selected entry. The table has columns: Sd Client Evt Group Timing Conf., TTL, and Subscribe Evt Group Retry. The data row is: client\_timingConf\_1, 0, 10.



# Fusion: Services 설정

## ▪ Required Instance 설정 (15)

- ✓ 동일한 방법으로 Required Event Group을 1개 더 생성하고, 다음 그림과 같이 수정함

Service Editor

The screenshot shows the Service Editor interface with the following details:

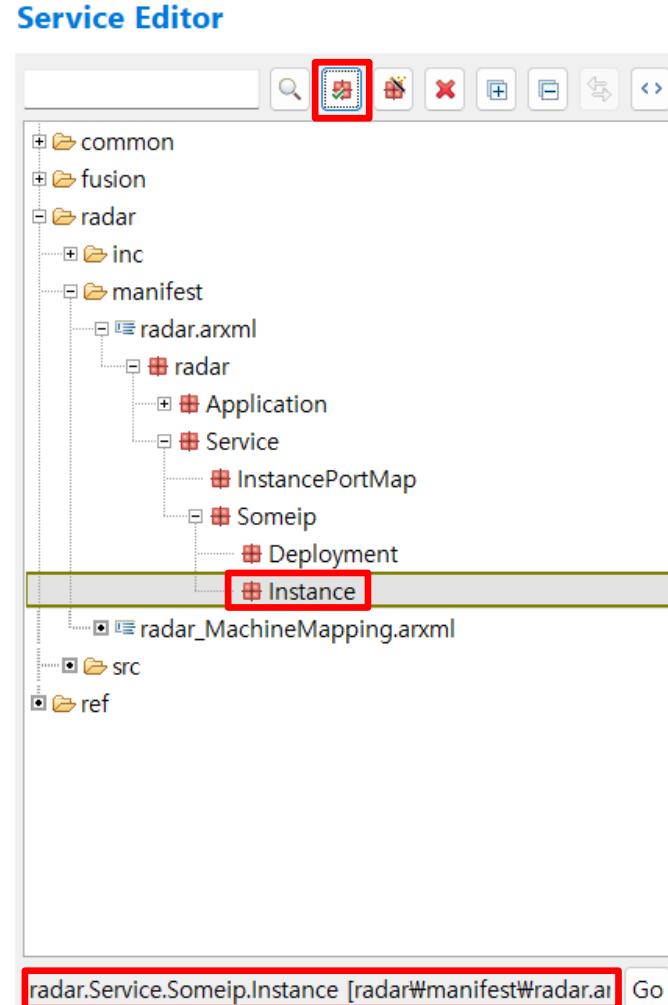
- Left Sidebar:** A tree view of service configurations. The "Required Instance" node is expanded, showing "Required Evt Group", "Evt Group", and "Sd Client Evt Group Timing Conf." under "Instances: required Event Groups".
- Main View:** A table titled "Instances: required Event Groups". It lists two entries:
  - SomeipREventGroup\_1: Contains "EventGroup\_1" and "client\_timingConf\_1".
  - SomeipREventGroup\_2: Contains "EventGroup\_2" and "client\_timingConf\_2".
- Table Headers:** The table has columns for "Required Evt Group", "Evt Group", and "Sd Client Evt Group Timing Conf.". To the right of the table, there are two more tables:
  - "Sd Client Event Group Timing Config" (with columns TTL, Subscr, Subscript Max., and Min.)
  - "+ Req/Resp Delay" (with columns TTL, Subscr, Subscript Max., and Min.)
- Row Selection:** The rows for "EventGroup\_1" and "EventGroup\_2" are highlighted with a red border.



# Radar: Services 설정

## ▪ Default Package 설정

- ✓ 'Instance' Package를 Default Package로 설정하고 아래 창을 통해 확인함



# Radar: Services 설정

## ■ Provided Instance 설정 (1)

- ✓ 좌측의 'Provided Instance' 탭으로 이동함
- ✓ 빈 곳에서 우클릭하여 Service Instance를 생성함

Service Editor

Instance	Related Deployer ID	LB Priority	LB Weight	Sd Server Config	Instances: sd Server Config	Someip Sd Server Service Instance Conf.	Offer Cyclic D

Create Service Instance

- Delete Service Instance
- Create sd Server Config
- Delete sd Server Config
- Add Initial Offer Behavior
- Delete Initial Offer Behavior
- Add Request Response Delay
- Delete Request Response Delay



Service Editor

Instance	Related Deployer ID	LB Priority	LB Weight	Sd Server Config	Instances: sd Server Config	Someip Sd Server Service Instance Conf.	Offer Cyclic D
Someip_P_Inst_1	1						

# Radar: Services 설정

## ▪ Provided Instance 설정 (2)

- ✓ 생성된 Service Instance을 다음 그림과 같이 수정함

Service Editor

The screenshot shows the Service Editor interface with the following details:

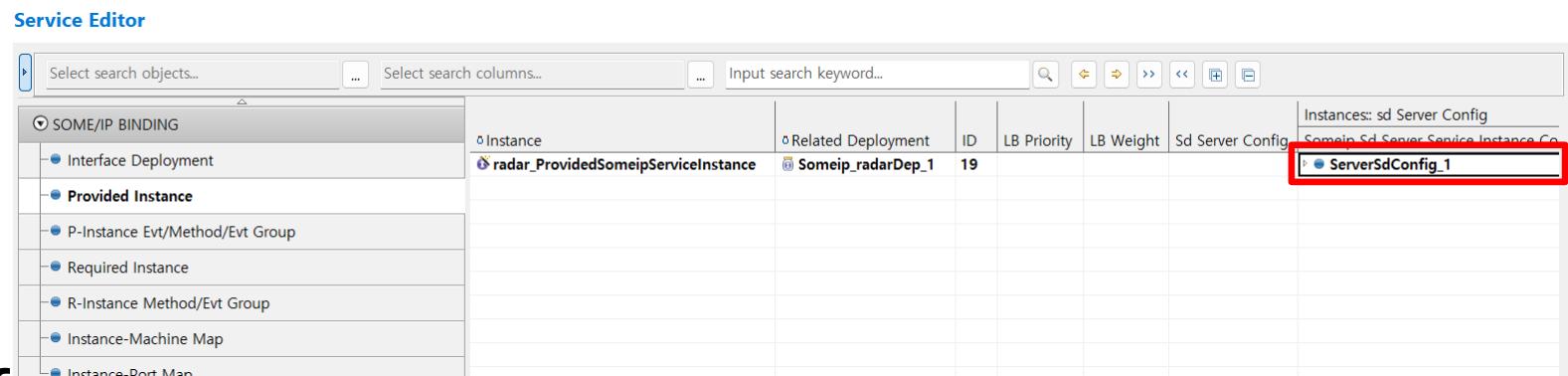
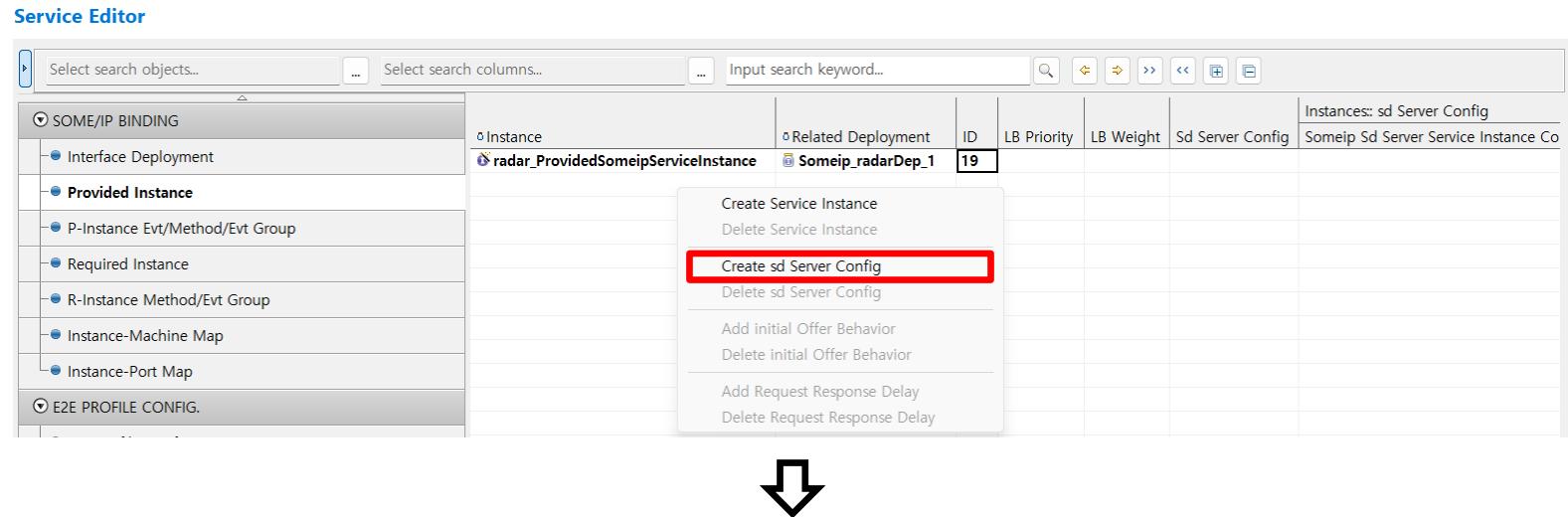
- SOME/IP BINDING** section expanded.
- Provided Instance** selected under **SOME/IP BINDING**.
- Instance** column: **radar\_ProvidedSomeipServiceInstance** (highlighted with a red box).
- Related Deployment** column: **Someip\_radarDep\_1**.
- ID** column: **19**.
- LB Priority**, **LB Weight**, and **Sd Server Config** columns are empty.
- Instance** column header: **Someip**.



# Radar: Services 설정

## ■ Provided Instance 설정 (3)

- ✓ 빈 곳에서 우클릭하여 SD Server Config를 생성함



# Radar: Services 설정

## ■ Provided Instance 설정 (4)

- ✓ Fusion 관련 설정에서의 방법과 동일하게 SD Server Config를 다음 그림과 같이 수정함
  - ✓ Initial Offer Behavior과 Request Response Delay를 추가해서 수정함

Service Editor

The screenshot shows the Service Editor interface with the following details:

- Left Sidebar:** A tree view of service configurations under "SOME/IP BINDING". The "Provided Instance" node is selected.
- Top Bar:** Includes search fields for "Select search objects...", "Select search columns...", and "Input search keyword...".
- Table View:** A table titled "Instances: sd Server Config" showing a single row for "Someip Sd Server Service Instance Conf".
- Row Data:** The row contains the following values:

	+initial Offer Behavior		+request Response Delay	
	Offer	Server Offer TTI	Delay Max	Delay Min
Someip Sd Server Service Instance Conf	0.0	10	0.1	0.01
radar_SomeipServerServiceInstanceConfig	3	0.2	1.5	1.5

# Radar: Services 설정

## ▪ Provided Instance 설정 (5)

- ✓ Service Instance와 SD Server Config를 연결함

Service Editor

The screenshot shows the Service Editor interface with a tree view on the left and a table view on the right.

**Tree View (Left):**

- SOME/IP BINDING
  - Interface Deployment
  - Provided Instance** (selected)
  - P-Instance Evt/Method/Evt Group
  - Required Instance
  - R-Instance Method/Evt Group
  - Instance-Machine Map
  - Instance-Port Map
- E2E PROFILE CONFIG.
- SECURE COMMUNICATION CONFIG.
  - TLS Secure Com. Props.
  - Crypto Cipher Suite.PSK Identity

**Table View (Right):**

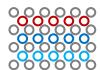
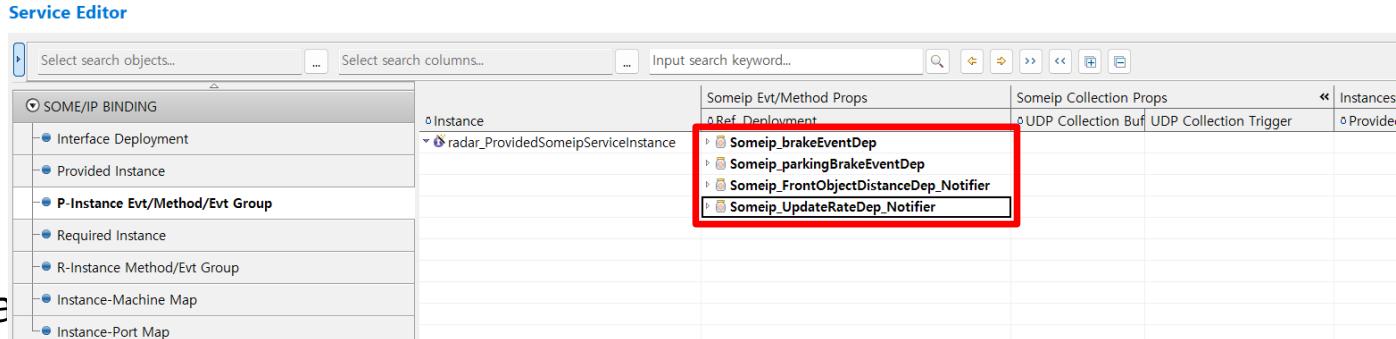
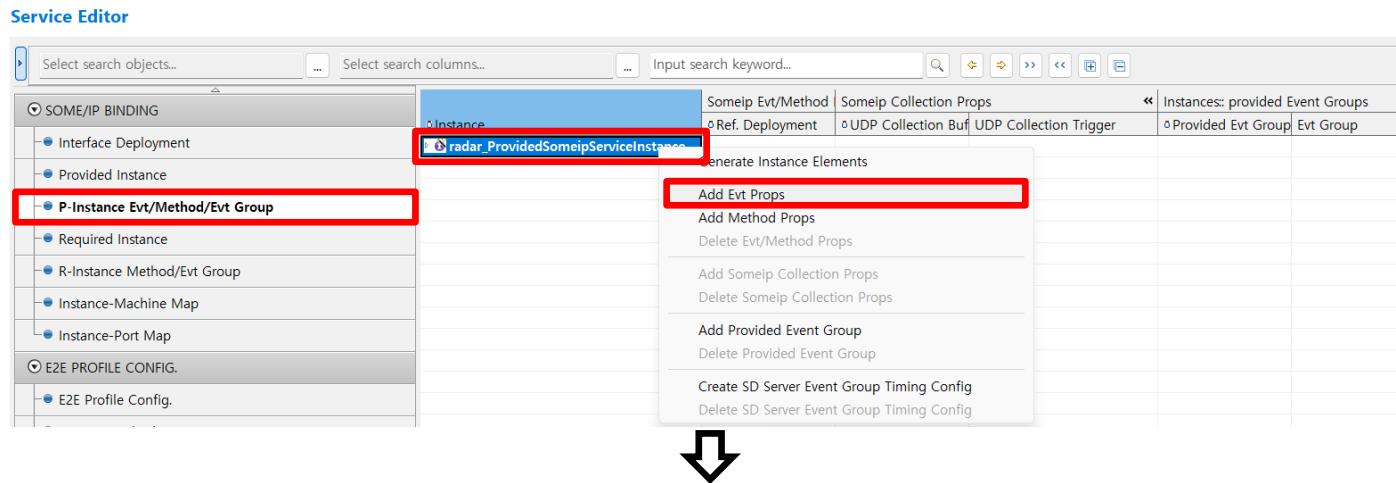
Instance	Related Deployment	ID	LB Priority	LB Weight	Sd Server Config
radar_ProvidedSomeipServiceInstance	Someip_radarDep_1	19			<b>radar_SomeipServerServiceInstanceConfig</b>



# Radar: Services 설정

## ■ Provided Instance 설정 (6)

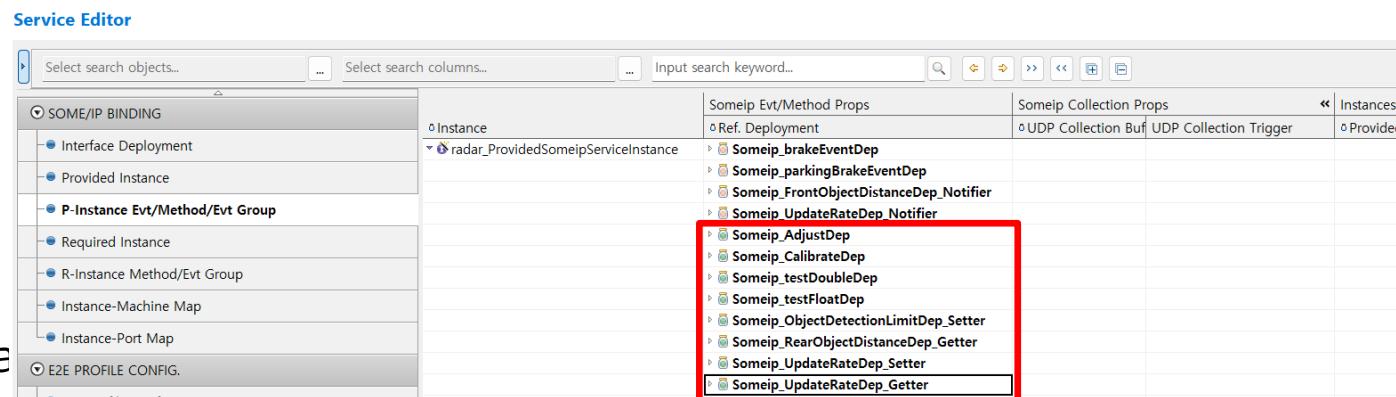
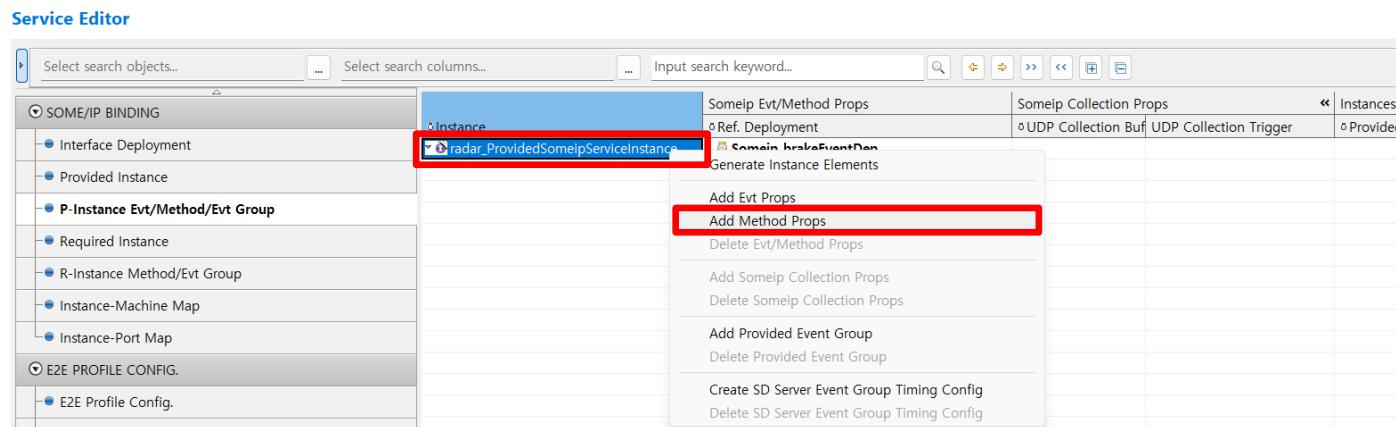
- ✓ 좌측의 'P-Instance Evt/Method/Evt Group' 탭으로 이동함
- ✓ 'radar\_ProvidedSomeipServiceInstance'에서 우클릭하여 Event Props를 생성함
- ✓ 생성된 Event Props는 다음 그림과 같이 수정함



# Radar: Services 설정

## ■ Provided Instance 설정 (7)

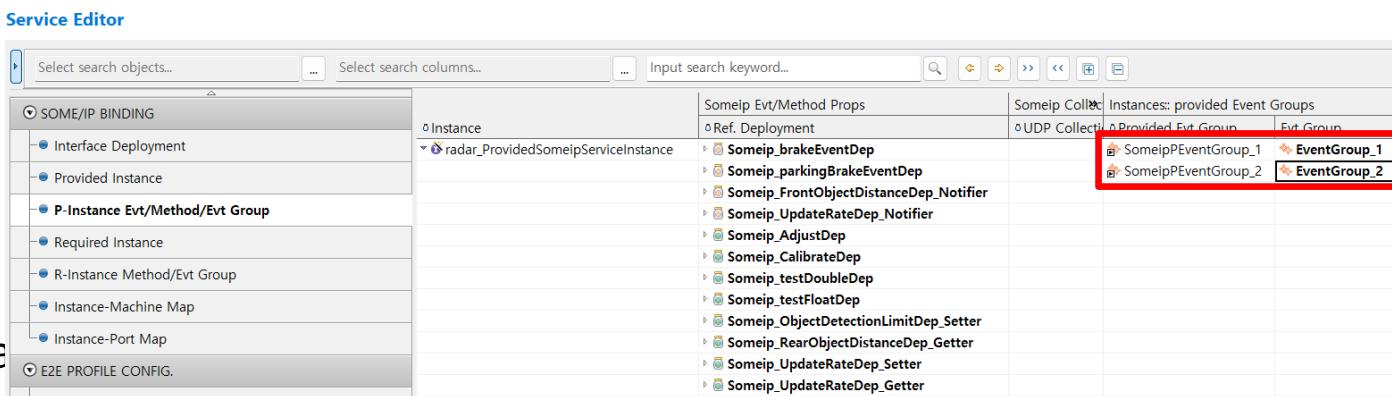
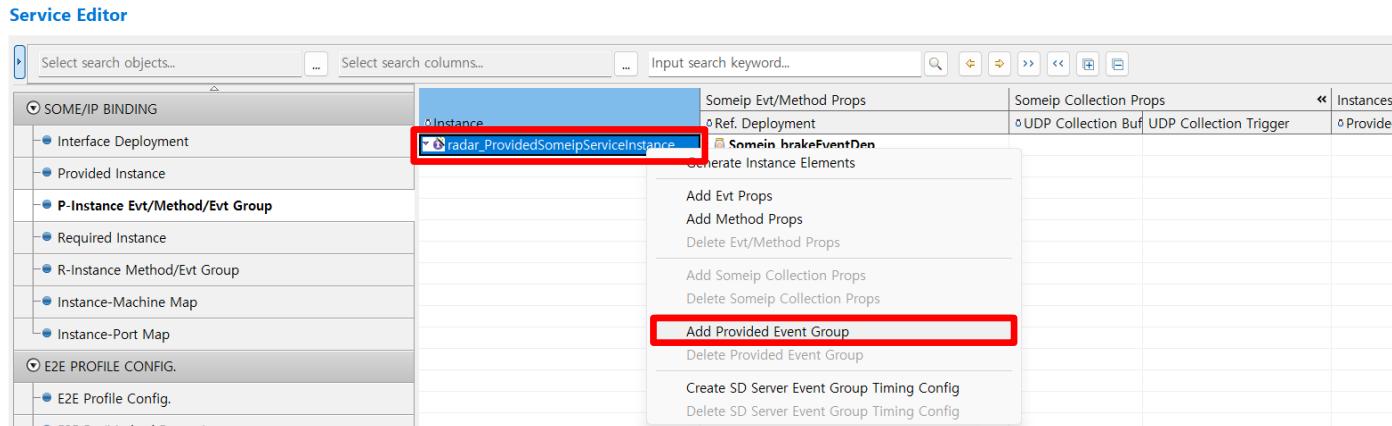
- ✓ 'radar\_ProvidedSomeipServiceInstance'에서 우클릭하여 Method Props를 생성함
- ✓ 생성된 Method Props는 다음 그림과 같이 수정함



# Radar: Services 설정

## ■ Provided Instance 설정 (8)

- ✓ 'radar\_ProvidedSomeipServiceInstance'에서 우클릭하여 Provided Event Group을 생성함
- ✓ 생성된 Provided Event Group은 다음 그림과 같이 수정함



# Radar: Services 설정

## ■ Provided Instance 설정 (9)

- ✓ 빈 곳에서 우클릭하여 SD Server Event Group Timing Config를 생성함
- ✓ 생성된 SD Server Event Group Timing Config를 다음 그림과 같이 Provided Event Group과 연결함

Service Editor

The screenshot shows the Service Editor interface. On the left, there is a tree view under the 'SOME/IP BINDING' section with nodes like 'Interface Deployment', 'Provided Instance', 'P-Instance Evt/Method/Evt Group', 'Required Instance', etc. In the center, a table titled 'Instances: provided Event Groups' lists two entries: 'SomeipPEventGroup\_1' and 'SomeipPEventGroup\_2'. Each entry has an 'Evt Group' column containing 'EventGroup\_1' and 'EventGroup\_2' respectively. A context menu is open over the second row, with the 'Create SD Server Event Group Timing Config' option highlighted with a red box. A large black arrow points downwards from this menu to the second screenshot.

Provided Evt Group	Evt Group	Multicast Threshold	Evt. Multicast UDP Port	IPv4 Multicast Addr.	IPv6 Multicast Addr.
SomeipPEventGroup_1	EventGroup_1	0	0	0.0.0	0:0:0:0:0:0
SomeipPEventGroup_2	EventGroup_2	0	0	0.0.0	0:0:0:0:0:0

Generate Instance Elements

- Add Evt Props
- Add Method Props
- Delete Evt/Method Props
- Add Someip Collection Props
- Delete Someip Collection Props
- Add Provided Event Group
- Delete Provided Event Group

Create SD Server Event Group Timing Config

Delete SD Server Event Group Timing Config

↓

Service Editor

This screenshot shows the same Service Editor interface after the configuration has been applied. The table now includes a new column 'Sd Server Evt Group Timing Config' next to the existing columns. The last two rows ('SomeipPEventGroup\_1' and 'SomeipPEventGroup\_2') now have additional columns for 'ReqResDelay\_1' and 'ReqResDelay\_2', which are highlighted with a red box. The 'ReqResDelay\_1' and 'ReqResDelay\_2' values are both set to '10.0'.

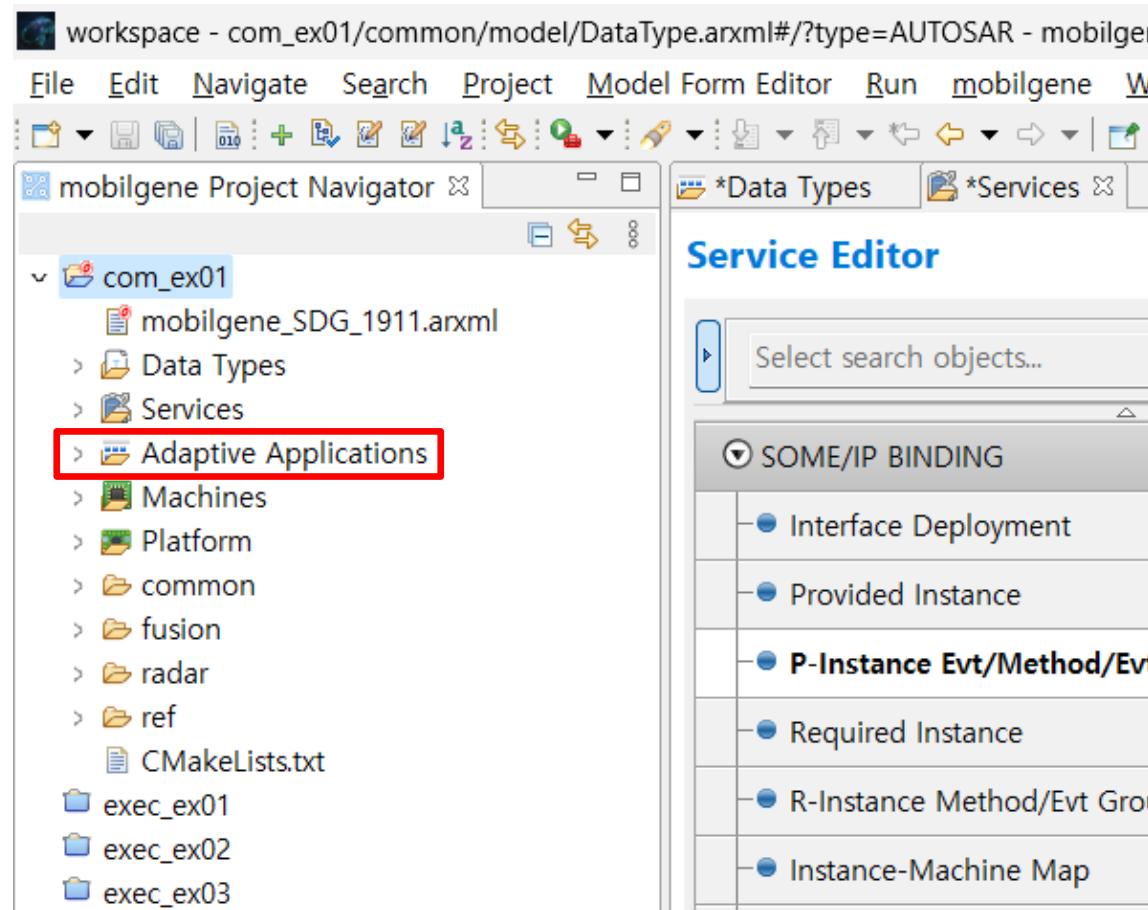
Provided Evt Group	Evt Group	Multicast Threshold	Evt. Multicast UDP Port	IPv4 Multicast Addr.	IPv6 Multicast Addr.	Sd Server Evt Group Timing Config
SomeipPEventGroup_1	EventGroup_1	0	0	0.0.0	0:0:0:0:0:0	ReqResDelay_1
SomeipPEventGroup_2	EventGroup_2	0	0	0.0.0	0:0:0:0:0:0	ReqResDelay_2



# Radar/Fusion: Port 설정

## ▪ Adaptive Applications Editor 활성화

- ✓ 왼쪽의 'mobilgene Project Navigator' 창에서 해당 Project의 'Adaptive Applications'를 더블 클릭함



# Radar/Fusion: Port 설정

- **Adaptive Applications Editor 활성화 확인**

- ✓ 활성화 된 Adaptive Applications Editor를 확인함

The screenshot shows the Application Editor interface with the following details:

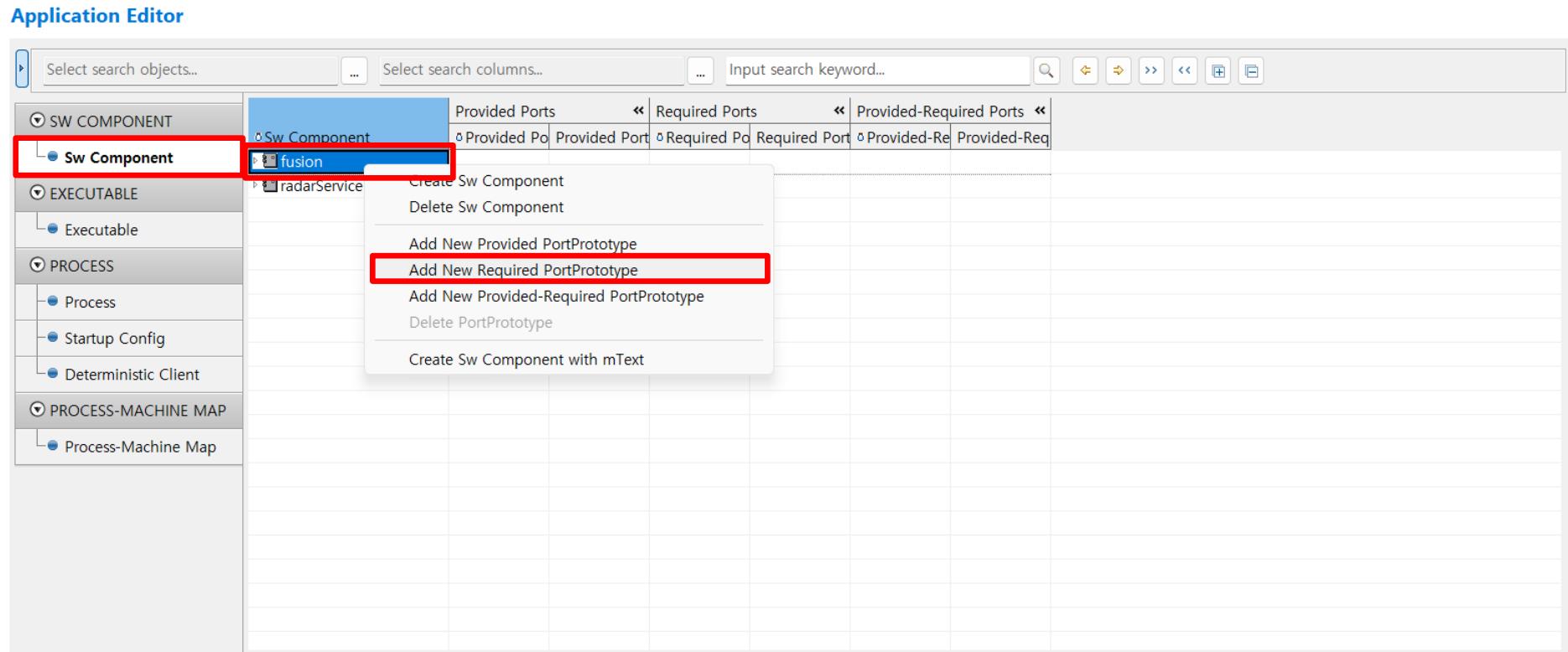
- Top Bar:** \*Data Types, \*Services, \*Adaptive Applications (selected).
- Toolbar:** Includes icons for search, filter, refresh, and various application operations.
- Search Bar:** Contains fields for "Select search objects...", "Select search columns...", and "Input search keyword...".
- Left Sidebar:** A tree view of project components:
  - common
  - fusion
  - radar
  - ref
- Central View:** A table for managing ports. The columns are:
  - Provided Ports
  - Required Ports
  - Provided-Required Ports
- Data in Table:**

Sw Component	Provided Po	Required Po	Required Port	Provided-Re	Provided-Req
fusion					
radarService					
- Bottom Status Bar:** radar.Service.Someip.Instance [radar#manifest#ra] Go

# Radar/Fusion: Port 설정

## ▪ Required Port 추가

- ✓ Required Port 추가를 위해 좌측의 'Sw Component' 탭으로 이동함
- ✓ 'fusion'에서 우클릭하여 'Add New Required PortPrototype'을 클릭함



# Radar/Fusion: Port 설정

## ▪ Required Port 추가 확인 및 설정

- ✓ 생성된 Required Port를 확인하고 다음과 같이 수정함
  - ✓ Required Port : 'radar\_RPort'
  - ✓ Required Port Interface : 'radar'

Application Editor

The screenshot shows the Application Editor interface with a search bar at the top and a tree view on the left. The main area displays a table with columns for Provided Ports, Required Ports, and Provided-Required Ports. A row for the 'fusion' component is selected, highlighting the 'Required Port' and 'Required Port Interface' fields, which are both set to 'radar\_RPort' and 'radar' respectively. The table has a red border around the highlighted row.

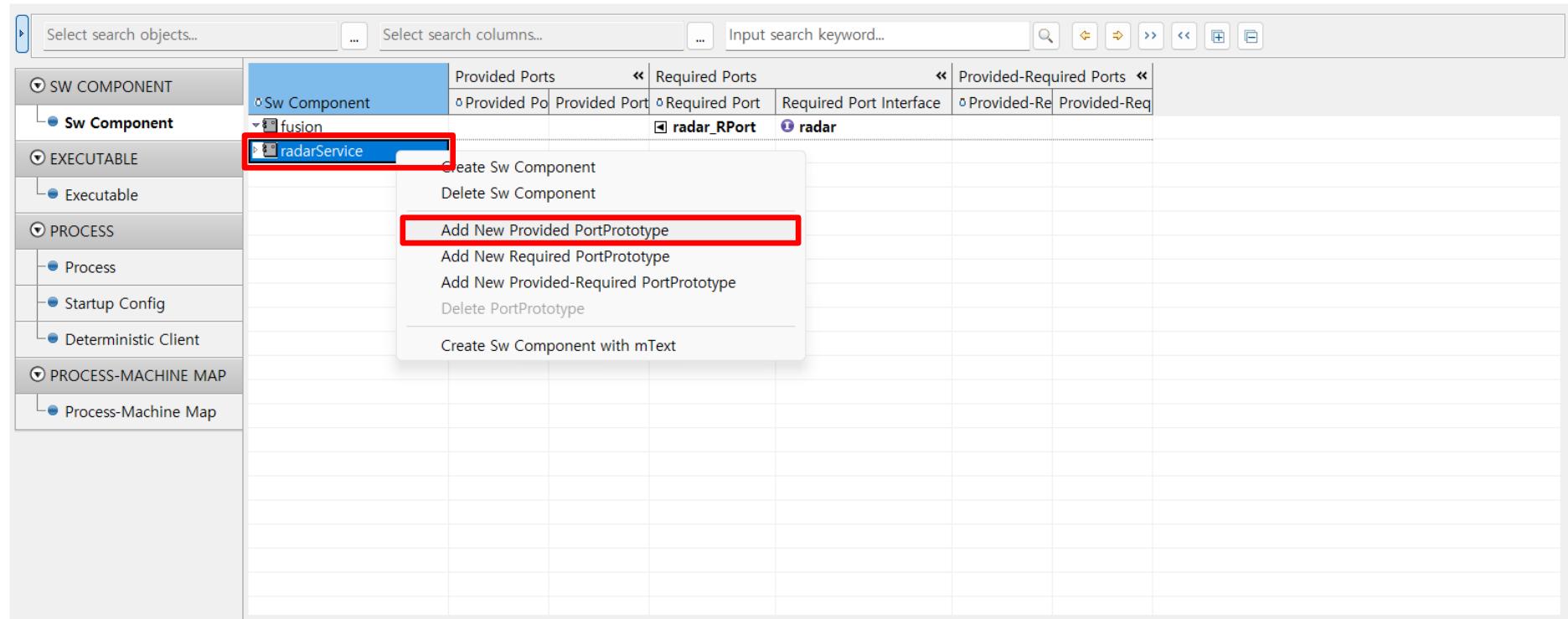
SW COMPONENT	Sw Component	Provided Ports		Required Ports		Provided-Required Ports	
		Provided Po	Provided Port	Required Port	Required Port Interface	Provided-Re	Provided-Req
EXECUTABLE	fusion			radar_RPort	radar		
PROCESS	radarService						

# Radar/Fusion: Port 설정

## ■ Provided Port 추가

- ✓ 'radarService'에서 우클릭하여 'Add New Provided PortPrototype'을 클릭함

Application Editor



# Radar/Fusion: Port 설정

## ■ Provided Port 추가 확인 및 설정

- ✓ 생성된 Provided Port를 확인하고 다음과 같이 수정함
  - ✓ Provided Port : 'radar\_PPort'
  - ✓ Provided Port Interface : 'radar'

Application Editor

The screenshot shows the Application Editor interface with the following details:

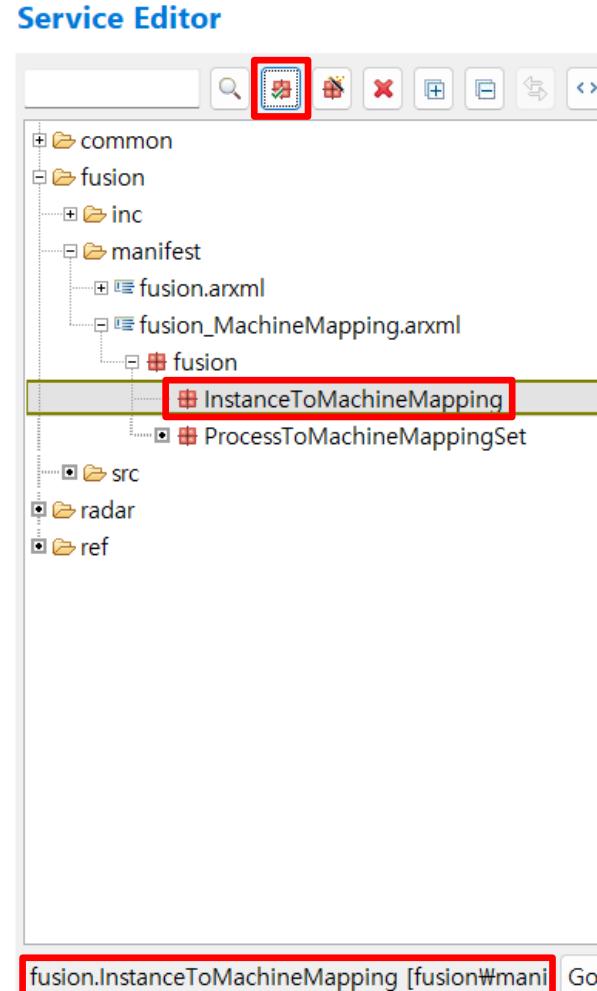
- Search Bar:** Select search objects..., Select search columns..., Input search keyword... with search icons.
- Object Tree:** SW COMPONENT (Sw Component), EXECUTABLE (Executable), PROCESS (Process, Startup Config, Deterministic Client), and PROCESS-MACHINE MAP (Process-Machine Map).
- Table View:** A grid view showing components and their ports. The columns are: Sw Component, Provided Ports, Required Ports, and Provided-Required Ports.
- Data in Table:**

Sw Component	Provided Ports	Required Ports	Provided-Required Ports
fusion		radar_RPort	radar
radarService	radar_PPort	radar	
- Red Box:** A red box highlights the row for the 'radarService' component, specifically the 'radar\_PPort' and 'radar' cells in the 'Provided Ports' and 'Required Ports' columns respectively.

# Fusion: Service Instance Mapping 설정

## ▪ Default Package 설정

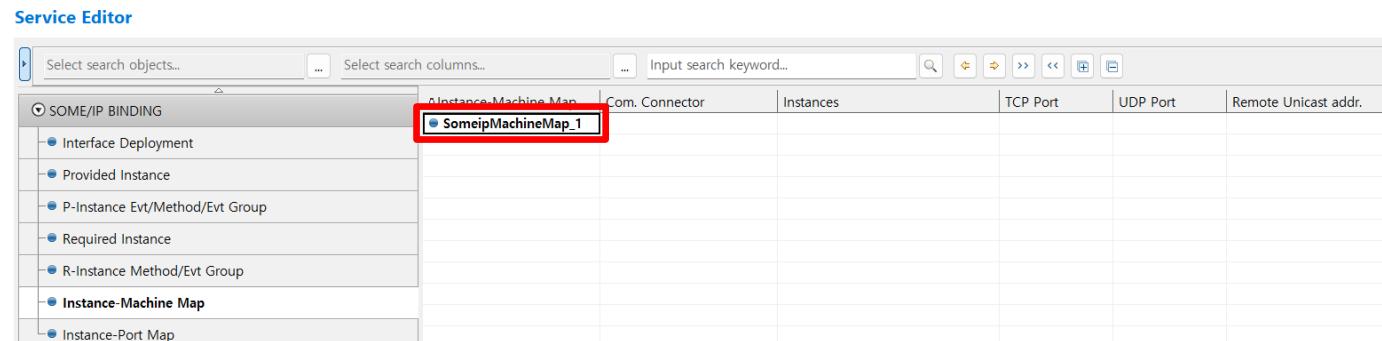
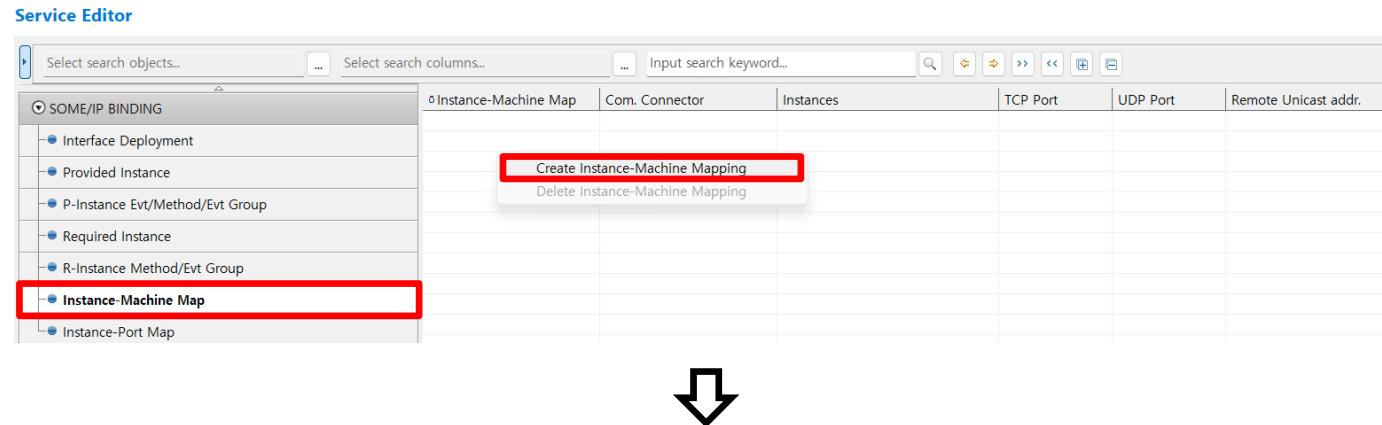
- ✓ 'InstanceToMachineMapping' Package를 Default Package로 설정하고 아래 창을 통해 확인함



# Fusion: Service Instance Mapping 설정

## ▪ Instance-Machine Map 설정 (1)

- ✓ 좌측의 'Instance-Machine Map' 탭으로 이동함
- ✓ 빈 곳에서 우클릭하여 Instance-Machine Mapping을 생성함



# Fusion: Service Instance Mapping 설정

## ▪ Instance-Machine Map 설정 (2)

- ✓ 생성한 Instance-Machine Mapping을 다음 그림과 같이 수정함
  - ✓ Required Instance와 Machine의 Connector를 연결함

Service Editor

Select search objects...	Select search columns...	Input search keyword...	Search	First	Previous	Next	Last	Reset	Print	Copy	Close
SOME/IP BINDING		Instance-Machine Map	Com. Connector	Instances	TCP Port	UDP Port	Ren.				
Interface Deployment		radar_RequiredServiceInstance_toMachine	Connector_ECU	radar_RequiredSomeipServiceInstance		5001					
Provided Instance											
P-Instance Evt/Method/Evt Group											
Required Instance											
R-Instance Method/Evt Group											
Instance-Machine Map											
Instance-Port Map											
E2E PROFILE CONFIG.											
E2E Profile Config.											
E2E Evt/Method Protection											
SECURE COMMUNICATION CONFIG.											
TLS Secure Com. Props.											
Crypto Cipher Suite.PSK Identity											

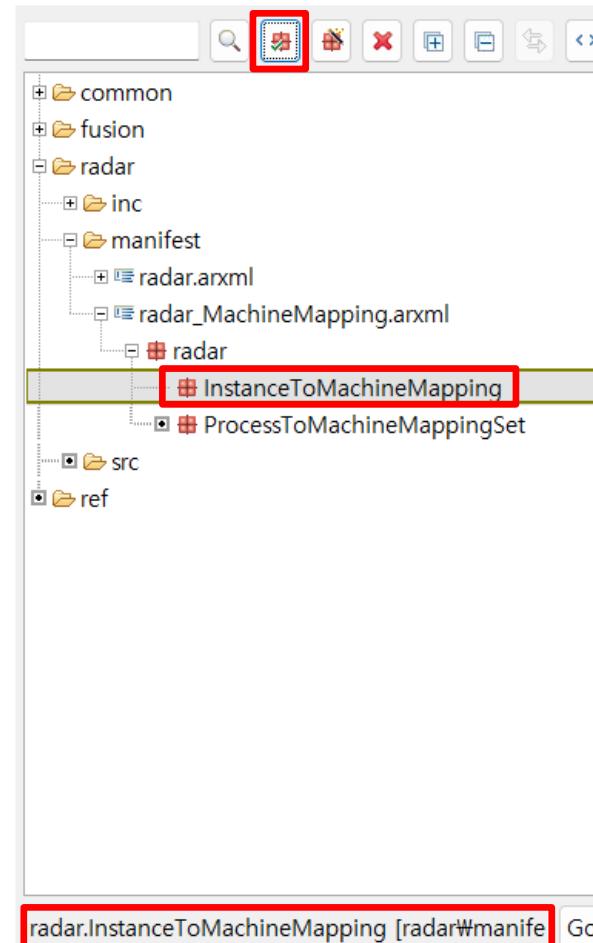


# Radar: Service Instance Mapping 설정

## ▪ Default Package 설정

- ✓ 'InstanceToMachineMapping' Package를 Default Package로 설정하고 아래 창을 통해 확인함

Service Editor



# Radar: Service Instance Mapping 설정

## ▪ Instance-Machine Map 설정

- ✓ Fusion 관련 설정에서의 방법과 동일하게 Instance-Machine Map 설정을 다음 그림과 같이 수정함

Service Editor

The screenshot shows the Service Editor interface with the 'Instance-Machine Map' section selected in the left sidebar. The main area displays a table of service instances mapped to machines. Two rows are highlighted with a red border:

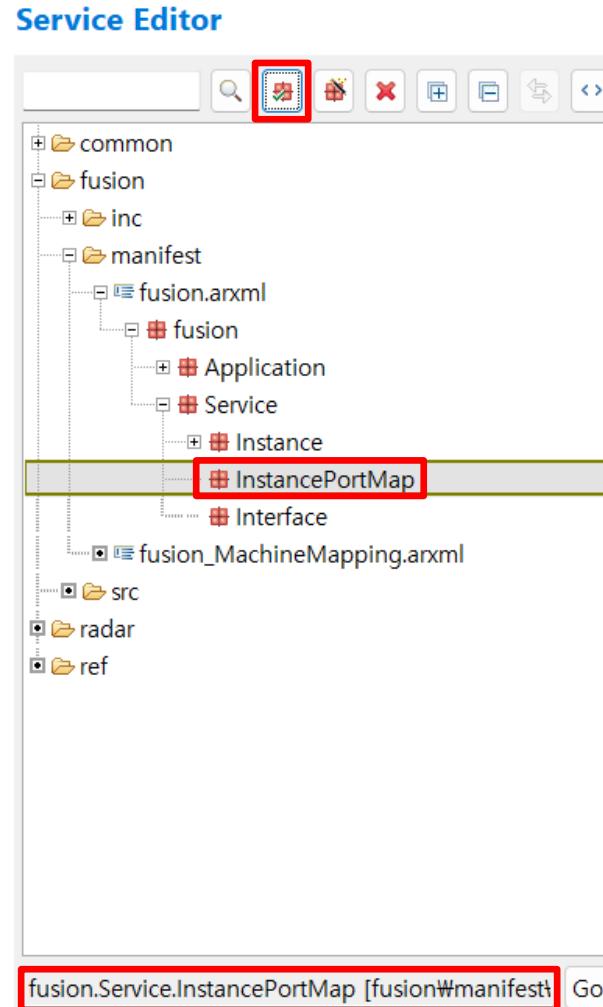
Instance-Machine Map	Com. Connector	Instances	TCP Port	UDP Port	Ren.
radar_RequiredServiceInstance_toMachine	Connector ECU	radar_RequiredSomeipServiceInstance	5001		
radar_ProvidedServiceInstance_toMachine	Connector_ECU	radar_ProvidedSomeipServiceInstance		5000	



# Fusion: Service Instance Mapping 설정

- **Default Package 변경**

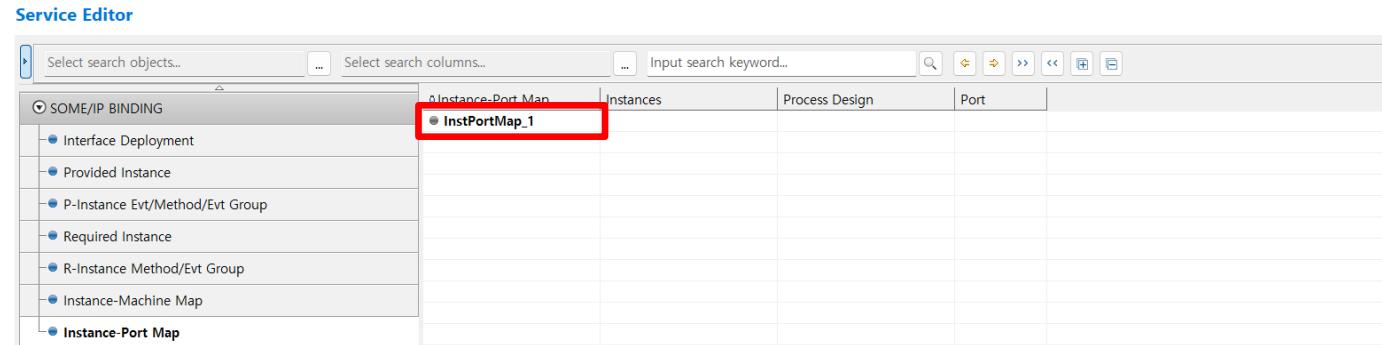
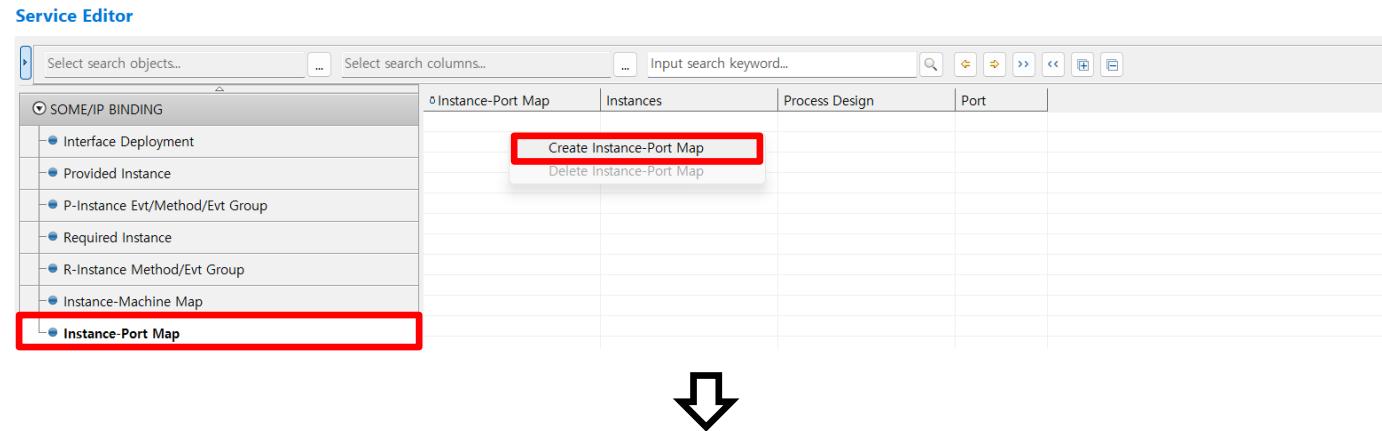
- ✓ 'InstancePortMap' Package를 Default Package로 설정하고 아래 창을 통해 확인함



# Fusion: Service Instance Mapping 설정

## ■ Instance-Port Map 설정 (1)

- ✓ 좌측의 'Instance-Port Map' 탭으로 이동함
- ✓ 빈 곳에서 우클릭하여 Instance-Port Map을 생성함



# Fusion: Service Instance Mapping 설정

## ▪ Instance-Port Map 설정 (2)

- ✓ 생성한 Instance-Port Map을 다음 그림과 같이 수정함
  - ✓ Required Instance와 Process의 Port를 연결함

Service Editor

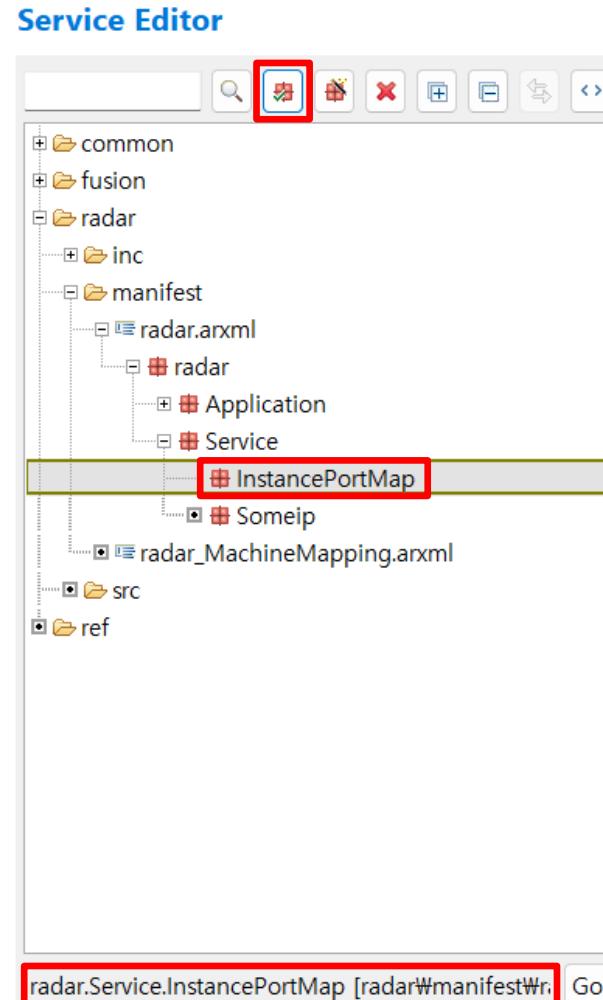
Select search objects...	Select search columns...	Input search keyword...	Instances	Process Design	Port
<ul style="list-style-type: none"><li>▼ SOME/IP BINDING<ul style="list-style-type: none"><li>● Interface Deployment</li><li>● Provided Instance</li><li>● P-Instance Evt/Method/Evt Group</li><li>● Required Instance</li><li>● R-Instance Method/Evt Group</li><li>● Instance-Machine Map</li><li>● Instance-Port Map</li></ul></li><li>▼ E2E PROFILE CONFIG.<ul style="list-style-type: none"><li>● E2E Profile Config.</li><li>● E2E Evt/Method Protection</li></ul></li><li>▼ SECURE COMMUNICATION CONFIG.<ul style="list-style-type: none"><li>● TLS Secure Com. Props.</li><li>● Crypto Cipher Suite.PSK Identity</li></ul></li></ul>	Instance-Port Map	<input checked="" type="radio"/> radar_RequiredServiceInstance_toPort	<input checked="" type="radio"/> radar_RequiredSomeipServiceInstance	<input checked="" type="radio"/> fusion_ProcessDesign	<input checked="" type="checkbox"/> radar_RPort [fusion]



# Radar: Service Instance Mapping 설정

- **Default Package 변경**

- ✓ 'InstancePortMap' Package를 Default Package로 설정하고 아래 창을 통해 확인함



# Radar: Service Instance Mapping 설정

## ▪ Instance-Port Map 설정

- ✓ Fusion 관련 설정에서의 방법과 동일하게 Instance-Port Map 설정을 다음 그림과 같이 수정함

Service Editor

The screenshot shows the Service Editor interface with the 'Instance-Port Map' section selected in the left sidebar. The main area displays a table with the following data:

Instances	Process Design	Port
radar_RequiredServiceInstance_toPort	fusion_ProcessDesign	radar_RPort [fusion]
radar_ProvidedServiceInstance_toPort	radarService_ProcessD...	radar_PPort [radarSe...]



# Fusion: CPP 코드 분석

## ▪ Fusion

- ✓ 'fusion\_activity.h'
  - ✓ 생성된 Proxy 헤더 파일 include
  - ✓ Fusion 클래스 정의
- ✓ 'fusion\_activity.c'
  - ✓ Fusion 클래스 내 메소드 구현 (Proxy 메소드 사용)
  - ✓ 데이터 수신 (Event, Field)에 대한 콜백 함수 구현
- ✓ 'main\_fusion.c'
  - ✓ 메인 함수 구현
  - ✓ EM 상태 보고 구현
  - ✓ 시그널 처리 구현

# Radar: CPP 코드 분석

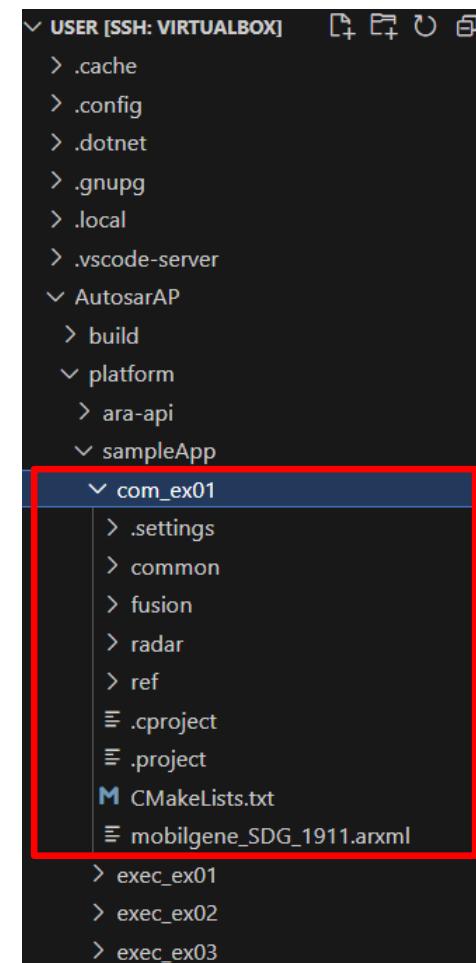
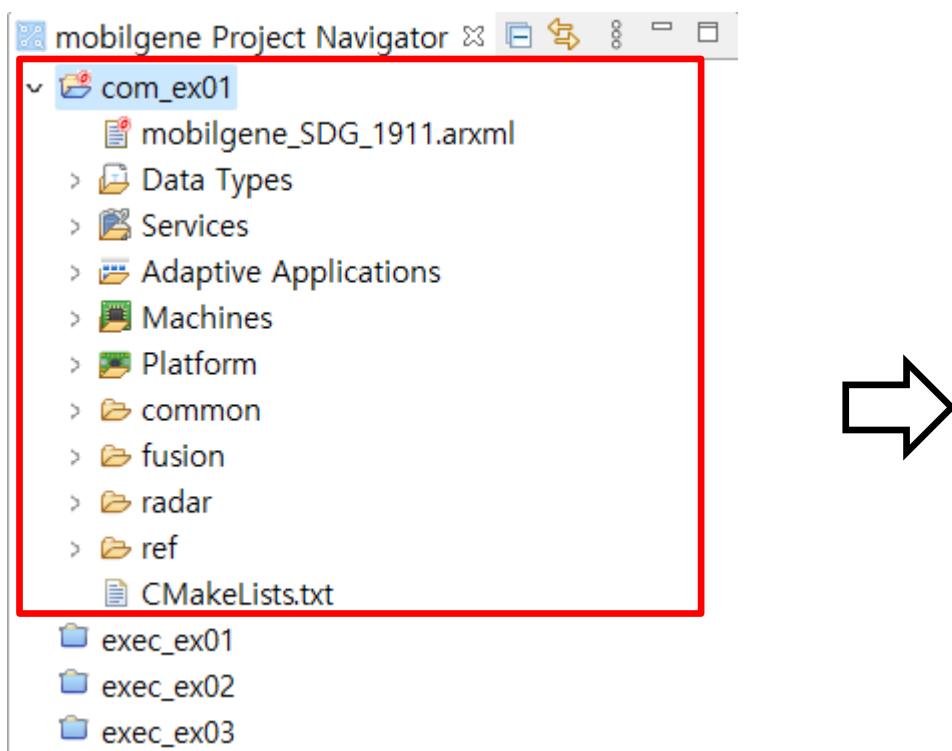
## ▪ Radar

- ✓ 'radar\_activity.h'
  - ✓ 생성된 Skeleton 헤더 파일 include
  - ✓ Skeleton을 상속받아 동작 구현 클래스 정의
  - ✓ Radar 클래스 정의
- ✓ 'radar\_activity.c'
  - ✓ Skeleton을 상속받은 동작 구현 클래스 내 메소드 구현
  - ✓ Radar 클래스 내 메소드 구현
- ✓ 'main\_radar.c'
  - ✓ 메인 함수 구현
  - ✓ EM 상태 보고 구현
  - ✓ 시그널 처리 구현

# COM: Build

- 개발 내용 빌드 환경으로 복사

- ✓ mobilgene A Studio에서 개발한 Adaptive Application을 빌드 환경으로 복사



# COM: Build

## ▪ Adaptive Application 관련 매크로 설정 추가

- ✓ 'build' – 'config' - 'Config.cmake'에 추가하고자 하는 Adaptive Application 관련 매크로 설정 추가

The screenshot shows a code editor with the following file structure:

- EXPLORER: Shows a project tree with 'USER [SSH: VIRTUALBOX]' expanded, containing '.cache', '.config', '.dotnet', '.gnupg', '.local', '.vscode-server', 'AutosarAP' (expanded), 'build' (expanded), 'arxmls', 'bin', 'config' (expanded), 'Config.cmake' (selected), 'Custom.cmake', 'Dependency.cmake', 'Environment.cmake', 'Install.cmake', 'docker', 'build.sh', 'platform' (expanded), 'ara-api'.
- ... (dropdown menu)
- Config.cmake (selected tab):

```
AutosarAP > build > config > Config.cmake
317 if ((NOT MGA_CONFIG_BUILD_SAMPLES) AND (${MGA_CONFIG_SOMEIP} MATCHES "vsomeip"))
321 endif()
322
323 if (MGA_CONFIG_BUILD_SAMPLES)
324     set(MGA_CONFIG_EXEC_EX01 ${BUILD_ENABLE})
325     set(MGA_CONFIG_EXEC_EX01_PRJ_NAME "EXEC_EX01")
326     set(MGA_CONFIG_EXEC_EX01_PRJ_PATH "sampleApp/exec_ex01")
327
328     set(MGA_CONFIG_EXEC_EX02 ${BUILD_ENABLE})
329     set(MGA_CONFIG_EXEC_EX02_PRJ_NAME "EXEC_EX02")
330     set(MGA_CONFIG_EXEC_EX02_PRJ_PATH "sampleApp/exec_ex02")
331
332     set(MGA_CONFIG_EXEC_EX03 ${BUILD_ENABLE})
333     set(MGA_CONFIG_EXEC_EX03_PRJ_NAME "EXEC_EX03")
334     set(MGA_CONFIG_EXEC_EX03_PRJ_PATH "sampleApp/exec_ex03")
335
336     set(MGA_CONFIG_COM_EX01 ${BUILD_ENABLE})
337     set(MGA_CONFIG_COM_EX01_PRJ_NAME "COM_EX01")
338     set(MGA_CONFIG_COM_EX01_PRJ_PATH "sampleApp/com_ex01")
339
340     set(MGA_CONFIG_EXEC_DEMO ${BUILD_ENABLE})
341     set(MGA_CONFIG_EXEC_DEMO_PRJ_NAME "EXEC_DEMO")
342     set(MGA_CONFIG_EXEC_DEMO_PRJ_PATH "ara-api/exec/samples/exec_demo")
```
- Dependency.cmake
- CMakeLists.txt

The code in the Config.cmake file defines several variables for different executables and a component. A specific section for 'COM\_EX01' is highlighted with a red box:

```
set(MGA_CONFIG_COM_EX01 ${BUILD_ENABLE})
set(MGA_CONFIG_COM_EX01_PRJ_NAME "COM_EX01")
set(MGA_CONFIG_COM_EX01_PRJ_PATH "sampleApp/com_ex01")
```

# COM: Build

## ▪ Adaptive Application 관련 의존성 설정 추가

- ✓ 'build' – 'config' - 'Dependency.cmake'에 추가하고자 하는 Adaptive Application 관련 의존성 설정 추가

The screenshot shows a terminal window with the following structure:

- EXPLORER pane on the left lists files and folders under 'USER [SSH: VIRTUALBOX]'. The 'Dependency.cmake' file is highlighted.
- Top tabs: Config.cmake, Dependency.cmake (highlighted), CMakeLists.txt.
- Content area shows the 'Dependency.cmake' file content:

```
AutosarAP > build > config > Dependency.cmake
45 set(ARA_CRYPTO_DAEMON_DEPENDS
46 set(ARA_CRYPTO_LIBRARY_DEPENDS
47 set(ARA_CRYPTO_GTEST_DEPENDS
48 set(ARA_CRYPTO_DAEMON_FOR_PACKAGER_DEPENDS
49 set(EXEC_DEMO_DEPENDS
50 set(EXEC_EX01_DEPENDS
51 set(EXEC_EX02_DEPENDS
52 set(EXEC_EX03_DEPENDS
53 set(COM_EX01_DEPENDS
54 set(NM_CONSUMER_EXAMPLE_DEPENDS
55 set(RADAR_FUSION_DEPENDS
56 set(RADAR_FUSION_VSOMEIP_DEPENDS
57 set(RADAR_FUSION_IPC_DEPENDS
58 set(IAM_GRANT_RADAR_FUSION_DEPENDS
59 set(CM_MULTI_PROTOCOL_DEPENDS
60 set(CM_MULTI_INSTANCE_SOMEIP_DEPENDS
61 set(CM_MULTI_INSTANCE_IPC_DEPENDS
62 set(SOMEIP_MULTI_CHANNEL_TEST_DEPENDS
63 set(LAUNCHER_DEPENDS

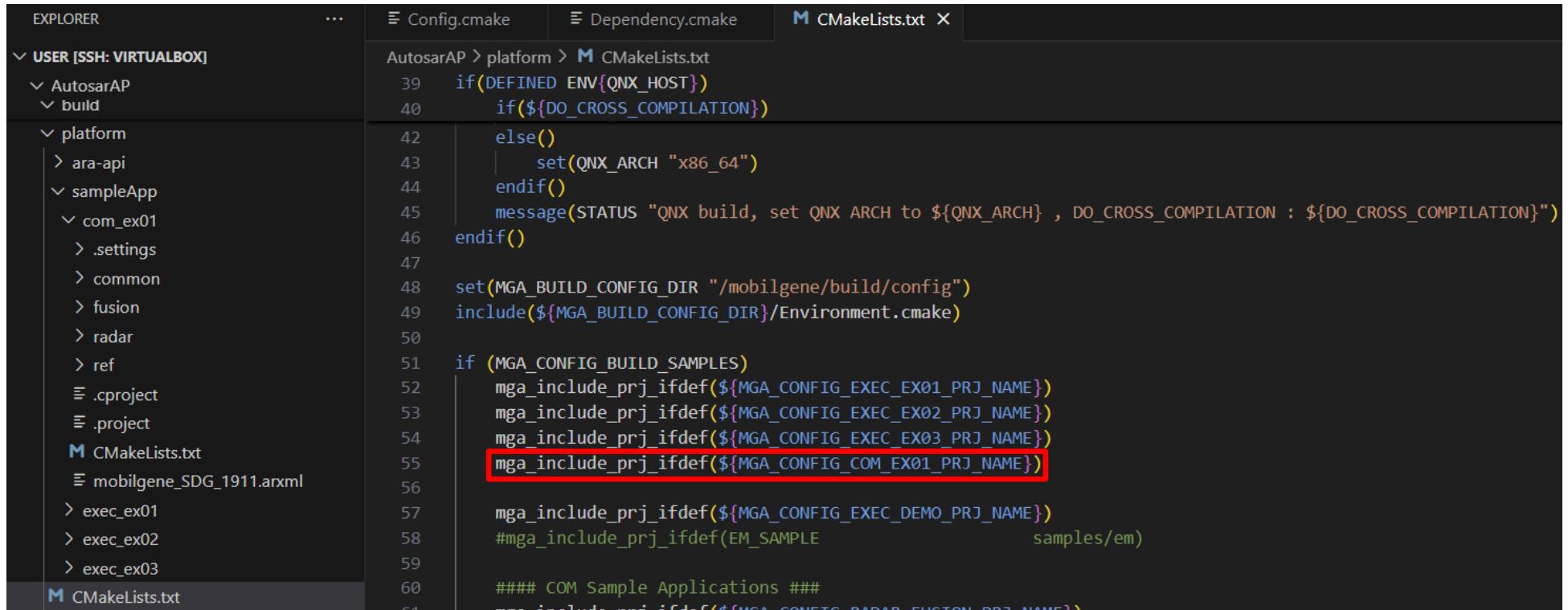
SOFTHSM ARA_LOG ARA_EXEC_API )
SOFTHSM ARA_TOOLS_GENERATOR ARA_LOG)
ARA_CRYPTO_LIBRARY ARA_CRYPTO_DAEMON)
SOFTHSM E2E_LIBCRC ARA_PER )
ARA_EXEC_API ARA_COM)
ARA_EXEC_API)
ARA_EXEC_API)
ARA_EXEC_API)
ARA_EXEC_API)
ARA_EXEC_API ARA_COM)
ARA_PER ARA_COM)
ARA_COM )
```

The line 'set(COM\_EX01\_DEPENDS' is highlighted with a red box.

# COM: Build

## ▪ Adaptive Application을 Build 목록에 추가

- ✓ Adaptive Application을 'platform' - 'CMakeLists.txt'에 추가하여 Build 목록에 추가



EXPLORER ... Config.cmake Dependency.cmake CMakeLists.txt X

USER [SSH: VIRTUALBOX]

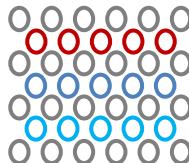
AutosarAP > platform > CMakeLists.txt

```
39 if(DEFINED ENV{QNX_HOST})
40   if(${DO_CROSS_COMPILATION})
41     else()
42       set(QNX_ARCH "x86_64")
43     endif()
44     message STATUS "QNX build, set QNX ARCH to ${QNX_ARCH} , DO_CROSS_COMPILATION : ${DO_CROSS_COMPILATION}"
45   endif()
46
47
48 set(MGA_BUILD_CONFIG_DIR "/mobilgene/build/config")
49 include(${MGA_BUILD_CONFIG_DIR}/Environment.cmake)
50
51 if (MGA_CONFIG_BUILD_SAMPLES)
52   mga_include_prj_ifdef(${MGA_CONFIG_EXEC_EX01_PRJ_NAME})
53   mga_include_prj_ifdef(${MGA_CONFIG_EXEC_EX02_PRJ_NAME})
54   mga_include_prj_ifdef(${MGA_CONFIG_EXEC_EX03_PRJ_NAME})
55   mga_include_prj_ifdef(${MGA_CONFIG_COM_EX01_PRJ_NAME})
56
57   mga_include_prj_ifdef(${MGA_CONFIG_EXEC_DEMO_PRJ_NAME})
58   #mga_include_prj_ifdef(EM_SAMPLE samples/em)
59
60 ##### COM Sample Applications #####
61   mga_include_prj_ifdef(${MGA_CONFIG_COMETE_PRJ_NAME})\n
```



# Q & A

**Thank you for your attention**



Architecture and  
Compiler  
for Embedded Systems Lab.

**School of Electronics Engineering, KNU**  
ACE Lab ([junho7513@knu.ac.kr](mailto:junho7513@knu.ac.kr))