HYUNDAI AUTOEVER

CDD Router User Manual

DOC. NO

SCOPE OF APPLICATION All Project/Engineering
Responsibility : Classic AUTOSAR Team

File Name CDD_Router_UM-EN.pdf
Creation MJ Kim 2023/09/12
Check HM Kim 2023/09/12
Approval JS Jang 2023/09/12
Edition Date: 2023/09/12
Document Management System

HYUNDAI AutoEver

Document Change Histroy				
Date (YYYY-MM-DD)	Ver.	Editor	Chap	Description (before -> after revision)
2016-08-24	1.5.0	Chan Kim	All	• Initial Creation
2017-03-20	1.5.1	Chan Kim	4.2 4.3	• Scope of the release changed Change Logs updated
2017-07-18	1.5.2	Chan Kim	4.2 4.3 8.3	• Scope of the release changed • Change Logs updated • Tool user manual added (Link)
2018-08-08	2.0.0	Chan Kim	3.1 4.2 4.3 4.4.1 8.1 8.2	• Module description added • Scope of the release changed • Change Logs updated • Limitation information added • Generation data related content added • SoAd connection relationship content added
2019-09-17	2.0.1	Jongsun Lim	4.2 4.3 8.1.7	• Scope of the release changed • Change Logs updated 411 sample code content added
2020-06-30	2.0.2.0	Hyungmin Shin	2 4.2 4.3	• SoAd added to reference list • Scope of the release changed • Change Logs updated
2021-03-02	2.1.0.0	Hyungmin Shin	4.2 4.3 5.6 8.1.5.1	• Scope of the release changed • Change Logs updated • CddPduRLowerLayerTxPdu configuration items added • Additional implementation method for CddPduRLowerLayerTxPdu/CddIfUpperLayerPduRef configuration
2021-03-25	2.2.0.0	Hyungmin Shin	4.2 4.3 5.8~11 8.1.8~11	• Scope of the release changed • Change Logs updated • CDD IPC related configuration item added • Additional implementation method for CDD IPC configuration
2021-09-27	2.2.0.1	Hyungmin Shin	4.2 4.3 8.4	• Scope of the release modified • Change Logs updated • Guide for CDD_Router module integration added
2021-11-09	2.2.0.2	Hyungmin Shin	4.2 4.3 8	• Scope of the release modified • Change Logs updated • Tool names removed
2021-11-30	2.2.1.0	Saemi Kwon	4.2 4.3	• Scope of the release modified • Change Logs updated
2022-01-25	2.3.0.0	Saemi Kwon	4.2 4.3 5.12~13 6.3.1 8.1.5 8.1.12~13 8.2	• Scope of the release modified • Change Logs updated • CanTp related configuration item added • CDD_RouterIF_CanTpTransmit function added • Description added • Info and examples on generator creation data for CanTp configuration added • CanTp connection relationship content added

			8.2.5	• Examples on CanTp module's PDU connection configuration added
2022-02-18	2.3.1.0	Hyungmin Shin	4.2 4.3	• Scope of the Release modified • Change Logs updated
2022-05-20	2.3.2.0	HyoungTae Kim	4.2 4.3	• Scope of the Release modified • Change Log modified
2023-03-31	2.3.2.1	Minji Kim	4.2 4.3	• Scope of the Release modified • Change Log modified
2023-06-09	2.3.3.0	Minji Kim	4.2 4.3 8.1.8~11 8.2.6 8.3	• Scope of the Release modified • Change Log modified • Change ddTargetAddressId to uint32 • CDD IPC connection relationship content added • Add IPC Import content
2023-08-25	2.3.4.0	Minji Kim	4.2 4.3	• Scope of the Release modified • Change Log modified
2023-09-12	2.3.5.0	Minji Kim	4.2 4.3	• Scope of the Release modified • Change Log modified

Table of Contents

1 Overview

This document provides references and guidance for users on parameter configuration and system design in the Hyundai AutoEver CDD Router module. CDD Router module was created based on AUTOSAR CDD concept and is not a standardized module. It is a sample type module which assists gateway processing, not supported by AUTOSAR standard, and provides ways to interface with AUTOSAR communication modules to application developers in the form of sample codes. Application developers can build new modules that meet requirements based on this module, or modify or apply the CDD Router module to use. If an application developer decides to modify or apply the CDD Router module to use, the developer shall be responsible for validating the operation of the module.

The following terms on configuration category mean:

- Changeable (C) : Items that can be configured by user
- Fixed (F) : Items that cannot be changed by user
- NotSupported (N) : Unavailable items

2 Reference

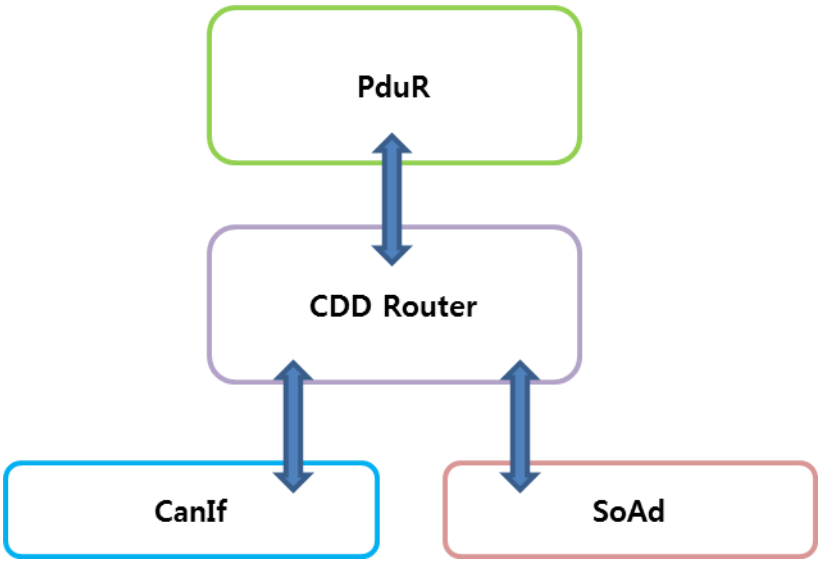
Sl. No.	Title	Version
1	AUTOSAR_SWS_PDURouter.pdf	3.2.0
2	AUTOSAR_SWS_CanInterface.pdf	5.0.0
3	AUTOSAR_SWS_SocketAdaptor.pdf	1.2.0

3 AUTOSAR System (Non-standard)

3.1 CDD Router Module

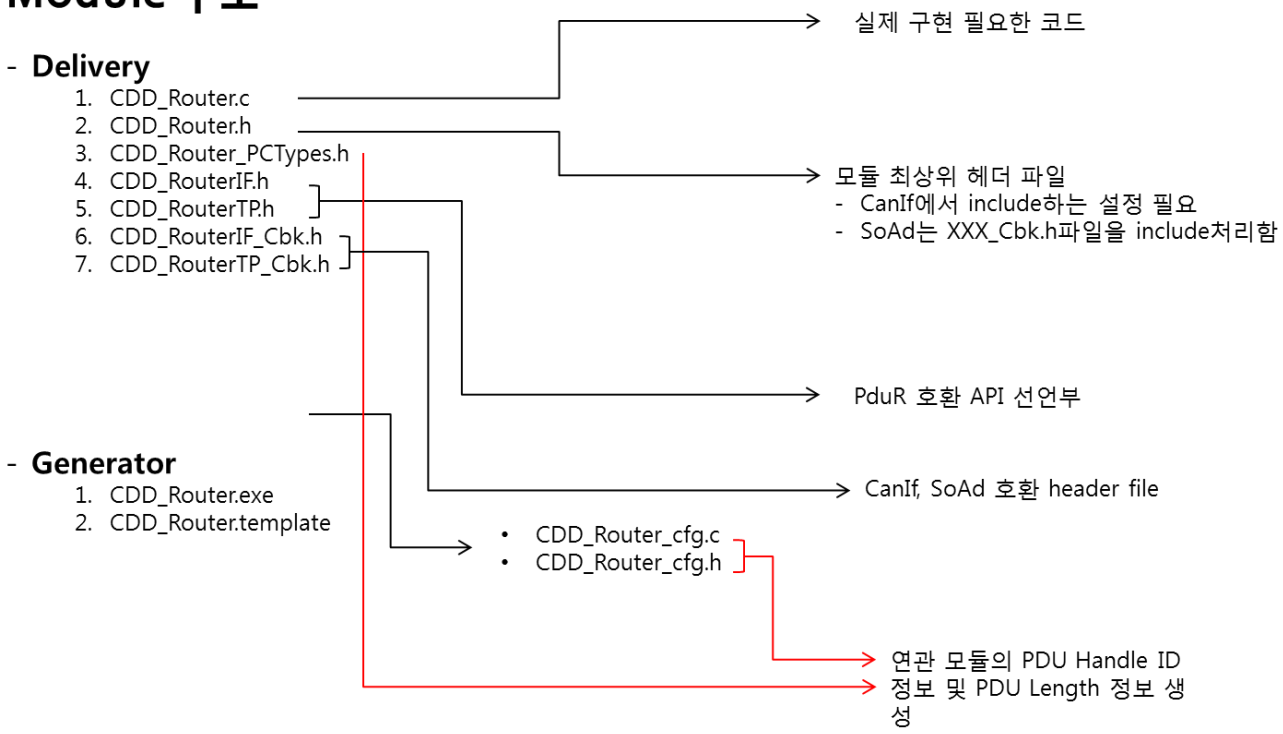
The CDD Router module deals with the transmission and reception of messages within ECU.

- It exists as a parent module to CanIf module, and can receive and/or send messages from/to CanIf module.
- It exists as a parent module to SoAd module, and can receive and/or send messages from/to SoAd module.
- It exists as a child module to PduR module, and can report reception of message to PduR and/or receive transmission request from PduR module.



The structure of CDD Router module is as follows.

Module 구조



4 Product Release Notes

4.1 Overview

This chapter provides the release information of Hyundai AutoEver CDD Router module, describing the features and restrictions of different versions of the CDD Router software product.

4.2 Scope of the Release

All content in this document applies only to the following Hyundai AutoEver CDD Router module.

Module name	AUTOSAR version	SWS version	Module version
CDD Router	N/A	N/A	2.3.5

※ Module version refers to the SW version defined in the BswModule Description file (Bswmd) of each module.

4.3 Change Log

4.3.1 Version 2.3.5.0

➢ Bug

- Duplication error for InterfaceLayer Target PDU generated in CanTpIfTxPduInfo fixed

Cause	Error in regular expression used in PduReference lookup. PDUs must be considered the same only when the ShortName matches exactly, but malfunctioned to generate for PDUs in the same has-a relationship
Operation effect	None
Setting effect	None
ASW Action	None

4.3.2 Version 2.3.4.0

➢ Improvements

- Improved CDD_Router_GaaUpperCanTpIfTxPduInfo to align with CanTpLowerLayerHandleId

Cause	CDD_Router_GaaUpperCanTpIfTxPduInfo is not aligned with CanTpLowerLayerHandleId, so the index order changes every generation
Operation effect	None
Setting effect	None
ASW Action	None

➢ Improvements

- Complement annotations for sample code in CDD_Router.c file

Cause	Detailed description of sample code
Operation effect	None
Setting effect	None
ASW Action	None

➢ Improvements

- Pdf change according to CDD_IPC Target Address Id 29bit support

Cause	Change from CDD_IPC Target Address Id 11bit Support to 29bit Support
Operation effect	None
Setting effect	None
ASW Action	None

4.3.3 Version 2.3.3.0

➢ Improvements

- Support CDD_IPC Target Address Id 29bit

Cause	Support CDD_IPC Target Address Id 29bit
Operation effect	None
Setting effect	None
ASW Action	None

➢ **Improvements**

- Improved alignment of input arxml annotations in generation files

Cause	Cause Modifying generator to output by sorting Input File list in Generated File
Operation effect	None
Setting effect	None
ASW Action	None

➢ **Improvements**

- Adding a description of CDD IPC to UM

Cause	Add IPC related figures and content
Operation effect	None
Setting effect	None
ASW Action	None

4.3.4 Version 2.3.2.1

➢ **Improvements**

- An English UM document added

Cause	Request for English UM document
Operation effect	None
Setting effect	None
ASW Action	None

4.3.5 Version 2.3.2.0

➢ **Improvements**

- Sample code and sample configuration revised for the public use of CDD_ROUTER

Cause	Include CDD_ROUTER as a default module in SWPs to be released
Operation effect	None
Setting effect	None
ASW Action	None

4.3.6 Version 2.3.1.0

➢ **Improvements**

- Duplication error for InterfaceLayer Target PDU generated in CDD_Router_GaaUpperIfTxPduInfo fixed

Cause	Error in regular expression used in PduReference lookup. PDUs must be considered the same only when the ShortName matches exactly, but malfunctioned to generate for PDUs in the same has-a relationship
Operation effect	None
Setting effect	None
ASW Action	None

4.3.7 Version 2.3.0.0

➢ **Features**

- CanTP module developed to be supported as upper layer

Cause	CanTp should be designated as upper layer to CDD Router to support RXSWIN feature at the platform level
Operation effect	None
Setting effect	Should configure CddCanTpLowerLayerRxPdu, CddCanTpLowerLayerTxPdu (Refer to User Manual 5.12~13) Should add Ecud_CanTp as input for CDD Router generation in SCons.arxml (SCons/RTSW/Generation/Module/CDD_Router/InputFileList) CanTp 1.12.0.0 or higher, and CanIf 3.0.2.0 or higher versions required
ASW Action	If RXSWIN feature is required, should change associated configuration and add Cdd_Router.c logic

4.3.8 Version 2.2.1.0

➢ **Improvements**

- Condition added to include sample code header file

Cause	Preprocessor condition added to include CanIf.h and SoAd.h when needed
Operation effect	None
Setting effect	None
ASW Action	None

4.3.9 Version 2.2.0.2

➢ **Improvements**

- Tool names in user manual removed

Cause	Tool names removed
Operation effect	None
Setting effect	None
ASW Action	None

4.3.10 Version 2.2.0.1

➢ **Improvements**

- Guide on module integration added to user manual

Cause	Users should be able to refer to the manual to configure and integrate CDD Router, which is to be included in platform deployment as default
Operation effect	None
Setting effect	None
ASW Action	None

4.3.11 Version 2.2.0.0

➢ **Features**

- API added for IPC support of CDD Router

Cause	API is required for IPC support
Operation effect	None
Setting effect	CDD/CDD_IPC_IF/CddComStackContribution/ CddPduRLowerLayerContribution CDD/CDD_IPC_TP/CddComStackContribution/ CddPduRLowerLayerContribution
ASW Action	None

4.3.12 Version 2.1.0.0

➢ **Features**

- Connection relationship added for PduRLowerLayer and IfUpperLayer of TxPdu

Cause	Connection relationship required for PduRLowerLayerPdu and IfUpperLayerPdu
Operation effect	None
Setting effect	CddComStackContribution/CddPduRLowerLayerContribution/ CddPduRLowerLayerTxPdu/CddIfUpperLayerPduRef
ASW Action	None

4.3.13 Version 2.0.2.0

➢ **Improvements**

- Variable names for I-PDU Handle ID, created via generation, changed

Cause	I-PDU Handles variable names had over 32 repeating characters which is a violation of MISRA
Operation effect	None
Setting effect	None
ASW Action	None

4.3.14 Version 2.0.1

➢ **Improvements**

- Sample code updated for AUTOSAR 4.1.1 StartOfReception support

Cause	Sample code updated for AUTOSAR 4.1.1 StartOfReception support
Operation effect	None
Setting effect	None
ASW Action	None

4.3.15 Version 2.0.0

➢ **Features**

- SoAd module support

Cause	SoAd module interface support added to enable special handling of Ethernet messages aside from AUTOSAR spec
Operation effect	None
Setting effect	None
ASW Action	None

➢ **Improvements**

- Sample code updated

Cause	Sample codes for Tx and SoAd related handling updated
Operation effect	None
Setting effect	None
ASW Action	None

- CDD Router dedicated PDF separated

Cause	Need to transition to module-dedicated PDF for module management limitations and maintenance, to allow usage of definitions from shared CDD configuration
Operation effect	None
Setting effect	None
ASW Action	None

4.3.16 Version 1.5.2

➢ **Improvements**
- Corrected validation check routine logic

Cause	If a PDU is named “XXXX” and another PDU exists with the name of “XXXX”+@, incorrect validation error is output
Operation effect	None
Setting effect	None
ASW Action	None

4.3.17 Version 1.5.1

➢ **Improvements**
- Specified configuration related Generator error messages

Cause	Generation error messages are unclear
Operation effect	None
Setting effect	None
ASW Action	None

4.3.18 Version 1.5.0

➢ **Improvements**
- Sample code updated

Cause	Provide sample codes for the usage of CDD Router
Operation effect	None
Setting effect	None
ASW Action	None

4.4 Module Release Notes

4.4.1 Limitations

➢ Application Code Implementation
Application developers should implement required actions directly to the CDD_Router.c file. Only function definitions that enable interfacing with PduR module, CanIf module, and SoAd module shall be provided.

➢ CDD_Router.c Sample Code
The CDD_Router.c file included in deployment consists of sample codes. The responsibility of implementation and validation as per actual requirements falls on individual application developers.

➢ Below interface functions are defined in sample codes but features are not supported. Consult with your deployment manager if necessary.
- CDD_RouterIF_CancelTransmit

- CDD_RouterTP_Transmit
- CDD_RouterTP_CancelTransmit
- CDD_RouterTP_CancelReceive

4.4.2 Deviations

None

5 Configuration Guide

CDD Router configuration as deployed by Hyundai AutoEver offers automated configuration for some features for DB file import when set as CDD_Router Type.

[CddPduRApiType configuration is currently unavailable.]
[CddTargetAddressId configuration is available only in IPC related configuration.]

5.1 CDD_RouterIF-CddComStackContribution-CddComIfUpperLayerContribution-CddComIfUpperLayerRxPdu Configuration

The following is a configuration for Rx PDUs received from CanIf module.

Parameter Name	Value	Category
Short Name	User Defined	C
CddComIfHandleId	User Defined	C
CddComIfPduRef	User Defined	C

1) Short Name
   - Component name, must be a unique value.
2) CddComIfHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddComIfPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.

5.2 CDD_RouterIF-CddComStackContribution-CddComIfUpperLayerContribution-CddComIfUpperLayerTxPdu Configuration

The following is a configuration for Tx PDUs transmitted to CanIf module.

Parameter Name	Value	Category
Short Name	User Defined	C
CddComIfHandleId	User Defined	C
CddComIfPduRef	User Defined	C

1) Short Name
   - Component name, must be a unique value.
2) CddComIfHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddComIfPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.

5.3 CDD_RouterIF-CddComStackContribution-CddSoAdUpperLayerContribution-CddSoAdUpperLayerRxPdu Configuration

The following is a configuration for Rx PDUs received from SoAd module.

Parameter Name	Value	Category
Short Name	User Defined	C
CddSoAdUpperLayerHandleId	User Defined	C
CddSoAdUpperLayerPduRef	User Defined	C

1) Short Name
   - Component name, must be a unique value.
2) CddSoAdUpperLayerHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddComIfPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.

5.4 CDD_RouterIF-CddComStackContribution-CddSoAdUpperLayerContribution-CddSoAdUpperLayerTxPdu Configuration

The following is a configuration for Tx PDUs transmitted to SoAd.

Parameter Name	Value	Category
Short Name	User Defined	C
CddSoAdUpperLayerHandleId	User Defined	C
CddSoAdUpperLayerPduRef	User Defined	C

1) Short Name
   - Component name, must be a unique value.
2) CddSoAdUpperLayerHandleId

- Must be a unique ID value of PDU.
- Should increase in consecutive way starting from 0.

3) CddComIfPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.

5.5 CDD_RouterIF-CddComStackContribution-CddPduRLowerLayerContribution-CddPduRLowerLayerRxPdu Configuration

The following is a configuration for Rx PDUs transported to PduR module.

Parameter Name	Value	Category
Short Name	User Defined	C
CddPduRLowerLayerHandleId	User Defined	C
CddPduRLowerLayerPduRef	User Defined	C

1) Short Name
   - Component name, must be a unique value.
2) CddPduRLowerLayerHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddPduRLowerLayerPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.

5.6 CDD_RouterIF-CddComStackContribution-CddPduRLowerLayerContribution-CddPduRLowerLayerTxPdu Configuration

The following is a configuration for Tx PDUs received from PduR.

Parameter Name	Value	Category
Short Name	User Defined	C
CddPduRLowerLayerHandleId	User Defined	C
CddPduRLowerLayerPduRef	User Defined	C
CddIfUpperLayerPduRef	User Defined	C

1) Short Name
   - Component name, must be a unique value.
2) CddPduRLowerLayerHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddPduRLowerLayerPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.
4) CddIfUpperLayerPduRef
   - PDU Reference value

- Reference the PDU defined in EcuC.
- Input CddComIfUpperLayerPduRef (or CddSoAdUpperLayerPduRef), which should connect with CddPduRLowerLayerPduRef via Cdd_Router.
- When configuring the reference, provide additional implementation method of CDD_RouterIF_Transmit(). (See Appendix 8.1.5.1)
- The configuration is required for using the PduRTxBuffer of Direct Gateway, and to pass Tx PDUs to
  PduR → Cdd_Router → CanIf for Can Bus-Off Handling .

5.7 CDD_RouterTP-CddComStackContribution-CddPduRLowerLayerContribution-CddPduRLowerLayerRxPdu Configuration

The following is a configuration for TP type Rx PDUs transported to PduR.

Parameter Name	Value	Category
Short Name	User Defined	C
CddPduRLowerLayerHandleId	User Defined	C
CddPduRLowerLayerPduRef	User Defined	C

1) Short Name
   - Component name, must be a unique value.
2) CddPduRLowerLayerHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddPduRLowerLayerPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.

5.8 CDD_IPC_IF-CddComStackContribution-CddPduRLowerLayerContribution-CddPduRLowerLayerTxPdu Configuration

The following is a configuration for Tx PDUs received from PduR.

Parameter Name	Value	Category
Short Name	User Defined	C
CddPduRLowerLayerHandleId	User Defined	C
CddPduRLowerLayerPduRef	User Defined	C
CddTargetAddressId	User Defined	C
CddIfUpperLayerPduRef	-	N

1) Short Name
   - Component name, must be a unique value.
2) CddPduRLowerLayerHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddPduRLowerLayerPduRef
   - PDU Reference value

- Reference the PDU defined in EcuC.

4) CddTargetAddressId
   - Logical Address ID of the Target, received via IPC

5.9 CDD_IPC_IF-CddComStackContribution-CddPduRLowerLayerContribution-CddPduRLowerLayerRxPdu Configuration

The following is a configuration for Rx PDUs received from PduR module.

Parameter Name	Value	Category
Short Name	User Defined	C
CddPduRLowerLayerHandleId	User Defined	C
CddPduRLowerLayerPduRef	User Defined	C
CddTargetAddressId	User Defined	C

1) Short Name
   - Component name, must be a unique value.
2) CddPduRLowerLayerHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddPduRLowerLayerPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.
4) CddTargetAddressId
   - Logical Address ID of the Target, received via IPC

5.10 CDD_IPC_TP-CddComStackContribution-CddPduRLowerLayerContribution-CddPduRLowerLayerTxPdu Configuration

The following is a configuration for Tx PDUs received from PduR.

Parameter Name	Value	Category
Short Name	User Defined	C
CddPduRLowerLayerHandleId	User Defined	C
CddPduRLowerLayerPduRef	User Defined	C
CddTargetAddressId	User Defined	C
CddIfUpperLayerPduRef	-	N

1) Short Name
   - Component name, must be a unique value.
2) CddPduRLowerLayerHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddPduRLowerLayerPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.
4) CddTargetAddressId

- Logical Address ID of the Target, received via IPC

5.11 CDD_IPC_TP-CddComStackContribution-CddPduRLowerLayerContribution-CddPduRLowerLayerRxPdu Configuration

The following is a configuration for Rx PDUs received from PduR module.

Parameter Name	Value	Category
Short Name	User Defined	C
CddPduRLowerLayerHandleId	User Defined	C
CddPduRLowerLayerPduRef	User Defined	C
CddTargetAddressId	User Defined	C

1) Short Name
   - Component name, must be a unique value.
2) CddPduRLowerLayerHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddPduRLowerLayerPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.
4) CddTargetAddressId
   - Logical Address ID of the Target, received via IPC

5.12 CDD_RouterIF-CddComStackContribution-CddCanTpLowerLayerContribution-CddCanTpLowerLayerRxPdu Configuration

The following is a configuration for Rx PDUs transported to CanTp module.

Parameter Name	Value	Category
Short Name	User Defined	C
CddCanTpLowerLayerHandleId	User Defined	C
CddCanTpLowerLayerPduRef	User Defined	C

1) Short Name
   - Component name, must be a unique value.
2) CddCanTpLowerLayerHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddCanTpLowerLayerPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.

5.13 CDD_RouterIF-CddComStackContribution-CddCanTpLowerLayerContribution-CddCanTpLowerLayerTxPdu Configuration

The following is a configuration for Tx PDUs received from CanTp module.

Parameter Name	Value	Category
Short Name	User Defined	C
CddCanTpLowerLayerHandleId	User Defined	C
CddCanTpLowerLayerPduRef	User Defined	C
CddIfUpperLayerPduRef	User Defined	C

1) Short Name
   - Component name, must be a unique value.
2) CddCanTpLowerLayerHandleId
   - Must be a unique ID value of PDU.
   - Should increase in consecutive way starting from 0.
3) CddCanTpLowerLayerPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.
4) CddIfUpperLayerPduRef
   - PDU Reference value
   - Reference the PDU defined in EcuC.
   - Input CddComIfUpperLayerPduRef,
     which should connect with CddCanTpLowerLayerPduRef via Cdd_Router.
   - When configuring the reference, provide additional implementation method of CDD_RouterIF_CanTpTransmit(). (See Appendix 8.1.12.1)

6 Application Programming Interface (API)

6.1 Type Definitions

None

6.2 Macro Constants

None

6.3 Functions

6.3.1 COMMUNICATION SERVICE

Function Name	CDD_RouterIF_TxConfirmation
Syntax	FUNC(void, CDD_ROUTER_CODE) CDD_RouterIF_TxConfirmation(PduIdType CddTxPduId)
Service ID	0x03
Sync/Async	synchronous

Reentrancy	No Reentrancy
Parameters (In)	CddTxPduId
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	The API called when Tx transmission to CanIf module is completed and confirmed
Preconditions	None
Configuration Dependency	None

Function Name	CDD_RouterIF_Transmit
Syntax	FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_RouterIF_Transmit(PduIdType CddTxPduId, P2CONST(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr)
Service ID	0x02
Sync/Async	synchronous
Reentrancy	No Reentrancy
Parameters (In)	CddTxPduId, PduInfoPtr
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	The API called when transmission request is made from PduR module to CDD Router module
Preconditions	None
Configuration Dependency	None

Function Name	CDD_RouterIF_RxIndication
Syntax	FUNC(void, CDD_ROUTER_CODE) CDD_RouterIF_RxIndication(PduIdType CddRxPduId, P2VAR(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr)
Service ID	0x00
Sync/Async	synchronous
Reentrancy	No Reentrancy
Parameters (In)	CddRxPduId, PduInfoPtr
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	The API called when CanIf module receives an Rx message
Preconditions	None
Configuration Dependency	None

Function Name	CDD_RouterIF_CancelTransmit(Not Supported)
Syntax	FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_RouterIF_CancelTransmit (PduIdType CddTxPduId)
Service ID	0x01
Sync/Async	synchronous
Reentrancy	No Reentrancy

Parameters (In)	CddTxPduId
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	The API called when a request is made to cancel transmission from PduR
Preconditions	None
Configuration Dependency	None

Function Name	CDD_RouterTP_Transmit(Not Supported)
Syntax	FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_RouterTP_Transmit (PduIdType CddTxPduId, P2CONST(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr)
Service ID	0x00
Sync/Async	synchronous
Reentrancy	No Reentrancy
Parameters (In)	CddTxPduId, PduInfoPtr
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	The API called when a request is made to transmit TP messages from PduR module
Preconditions	None
Configuration Dependency	None

Function Name	CDD_RouterTP_CancelTransmit(Not Supported)
Syntax	FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_RouterTP_CancelTransmit (PduIdType CddTxPduId)
Service ID	0x02
Sync/Async	synchronous
Reentrancy	No Reentrancy
Parameters (In)	CddTxPduId
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	The API called when transmission of TP messages is canceled from PduR module
Preconditions	None
Configuration Dependency	None

Function Name	CDD_RouterTP_CancelReceive(Not Supported)
Syntax	FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_RouterTP_CancelReceive (CONST(PduIdType, CDD_ROUTER_CODE) CddRxPduId)
Service ID	0x01
Sync/Async	synchronous
Reentrancy	No Reentrancy
Parameters (In)	CddRxPduId

Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	The API called when reception of TP messages is canceled from PduR module
Preconditions	None
Configuration Dependency	None

Function Name	CDD_IPC_IF_Transmit
Syntax	FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_IPC_IF_Transmit(PduIdType CddTxPduId, P2CONST(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr)
Service ID	0x02
Sync/Async	synchronous
Reentrancy	No Reentrancy
Parameters (In)	CddTxPduId, PduInfoPtr
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	The API called when transmission request is made from PduR module to CDD Router module
Preconditions	None
Configuration Dependency	None

Function Name	CDD_IPC_TP_Transmit
Syntax	FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_IPC_TP_Transmit (PduIdType CddTxPduId, P2CONST(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr)
Service ID	0x00
Sync/Async	synchronous
Reentrancy	No Reentrancy
Parameters (In)	CddTxPduId, PduInfoPtr
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	The API called when a request is made to transmit TP messages from PduR module
Preconditions	None
Configuration Dependency	None

Function Name	CDD_IPC_RxIndication
Syntax	FUNC(void, CDD_ROUTER_CODE) CDD_IPC_RxIndication (P2CONST(uint8, AUTOMATIC, CDD_ROUTER_APPL_DATA)DataPtr)
Service ID	0x00
Sync/Async	synchronous
Reentrancy	No Reentrancy
Parameters (In)	DataPtr
Parameters (Inout)	None

Parameters (Out)	None
Return Value	void
Description	Call made to transport data from IPC Driver to Cdd Router
Preconditions	None
Configuration Dependency	None

Function Name	CDD_RouterIF_CanTpTransmit
Syntax	FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_RouterIF_CanTpTransmit (PduIdType CddTxPduId, P2CONST(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr)
Service ID	0x06
Sync/Async	synchronous
Reentrancy	No Reentrancy
Parameters (In)	CddTxPduId, PduInfoPtr
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Std_ReturnType
Description	The API called when a request to transmit Tx Pdu is made from CanTP module to CDD Router
Preconditions	None
Configuration Dependency	None

7 Generator

7.1 Generator Option

Options	Description
-H/-Help	To display help regarding usage of the tool.
-O/-Output	To generate the output files in the specified directory location.
-V/-Version	To display the copyright information and the tool version.
-L/-Log	To generate “$BswConfig::Lis_File_Name” file.
-D/-DryRun	To execute in validation mode.

Options	Description
-I/-Info	To disable an Information Message(s).
-W/-Warn	To disable Warning Message(s).
-DDT	Not to generate the time stamp in the generated files.

7.2 Generator Message

This section helps to analyze the errors or warnings displayed during the execution of the tool. It ensures conformance of input file(s) with syntax and semantics.

The Generation Tool displays errors or warnings or information when the user has configured incorrect inputs. The format of Error/Warning/Information message is as shown below:

- ERR/WRN/INF<mid><xxx>: < Error/Warning/Information Message>
- Where,
- <mid>: 050 –Module Id (099) for user configuration checks.
- 000 – for command line checks.
- <xxx>: 001 – 999 – Message ID.
- File Name : Name of the file in which the error has occurred
- Path : Absolute path of the container in which the parameter is present

'File Name' and 'Path' are optional.

Below section provides the list of module specific error, warning and information messages.

7.2.1 Error Message

The following section gives the list of error messages displayed by the Generation Tool.

ERR050001: The input arxmls are not validated against the schema. Please correct the arxml as per schema or contact HYUNDAI AUTOEVER Co.,Ltd. for any support.

This is an Unexpected Error. On the occurrence of this error contact Hyundai-AutoEver Basic Platform Develoment Team.

ERR050002: Unexpected Error Found. This error may be due to the incorrect configuration of the element(s) 'Element Name'. Please correct the arxml as per schema or contact HYUNDAI AUTOEVER Co.,Ltd. for any support.

This error occurs, if the structure fields that are to be generated in the C Source file are empty. Contact Hyundai-AutoEver Basic Platform Develoment Team.

ERR050003: 'Component Name' Component is not present in the input file(s).

This error occurs, if any one of CanIf or EcuC or PduR component is not present in any of the input ECU Configuration Description File(s).

8 Appendix

8.1 Information on data generated by Generator and examples

8.1.1 Information on Tx PDU of CanIf module connected with CDD Router

```
/* Structure for Lower Layer IF Tx PDU */
typedef struct STagCDD_Router_LowerIfTxPduInfo
{
  PduIdType ddTxPduHandleId;
  PduLengthType ddPduLength;
}CDD_Router_LowerIfTxPduInfo;

CONST(CDD_Router_LowerIfTxPduInfo, CDD_ROUTER_CONST) CDD_Router_GaaLowerIfTxPduInfo[]
```

The value of ddTxPduHandleId defined in this data refers to the unique Handle ID for the PDU as configured in CanIf.

Example: When requesting a message transmission to CanIf module
CanIf_Transmit((PduIdType)CDD_Router_GaaLowerIfTxPduInfo[HandleID].ddTxPduHandleId, PduInfoPtr);

The HandleID above refers to the HandleID of PDU configured in CDD_ROUTER connected with Tx PDU of CanIf module.
Utilize the MACRO generated under “IF Tx Lower I-PDU Handles” in the CDD_Router_Cfg.h file.

8.1.2 Information on Rx PDU of CanIf module connected with CDD Router

```
/* Structure for Lower Layer IF Rx PDU */
typedef struct STagCDD_Router_LowerIfRxPduInfo
{
  PduIdType ddRxPduHandleId;
  PduLengthType ddPduLength;
}CDD_Router_LowerIfRxPduInfo;

CONST(CDD_Router_LowerIfRxPduInfo, CDD_ROUTER_CONST) CDD_Router_GaaLowerIfRxPduInfo[]
```

The value of ddRxPduHandleId defined in this data refers to the unique HandleID of the PDU as configured in CanIf module.
While the Rx PDU data is not used for the implementation of basic transmission code, it is created for any future application designs and possible requirement for scalability.

8.1.3 Information on Tx PDU of SoAd module connected with CDD Router

```
/* Structure for Lower Layer IF Tx PDU */
typedef struct STagCDD_Router_LowerIfTxPduInfo
{
  PduIdType ddTxPduHandleId;
  PduLengthType ddPduLength;
}CDD_Router_LowerIfTxPduInfo;

CONST(CDD_Router_LowerIfTxPduInfo, CDD_ROUTER_CONST) CDD_Router_GaaLowerSoAdIfTxPduInfo[]
```

The value of ddTxPduHandleId in the data refers to the unique HandleID of the PDU configured in SoAd.

Example: When requesting message transfer to SoAd module
SoAdIf_Transmit((PduIdType)CDD_Router_GaaLowerSoAdIfTxPduInfo[HandleID].ddTxPduHandleId, PduInfoPtr);

The HandleID above refers to the HandleID of PDU configured in CDD_ROUTER connected with Tx PDU of CanIf module.
Utilize the MACRO generated under “IF Tx Lower SoAd I-PDU Handles” in the CDD_ROUTER_Cfg.h file.

8.1.4 Information on Rx PDU of SoAd module connected with CDD Router

```
/* Structure for Lower Layer IF Rx PDU */
typedef struct STagCDD_Router_LowerIfRxPduInfo
{
  PduIdType ddRxPduHandleId;
  PduLengthType ddPduLength;
}CDD_Router_LowerIfRxPduInfo;

CONST(CDD_Router_LowerIfRxPduInfo, CDD_ROUTER_CONST)
CDD_Router_GaaLowerSoAdIfRxPduInfo[]
```

The value of ddRxPduHandleId defined in this data refers to the unique HandleID for the PDU as configured in SoAd module.
While the Rx PDU data is not used for the implementation of basic transmission code, it is created for any future application designs and possible requirement for scalability.

8.1.5 Information on Tx PDU of PduR module connected with CDD Router

```
/* Structure for Upper Layer IF Tx PDU */
typedef struct STagCDD_Router_UpperIfTxPduInfo
{
  PduIdType ddTxPduHandleId;
PduLengthType ddPduLength;

PduIdType ddLoTargetPduHandleId;
Cdd_Router_Layer ddLayerType;
}CDD_Router_UpperIfTxPduInfo;

CONST(CDD_Router_UpperIfTxPduInfo, CDD_ROUTER_CONST) CDD_Router_GaaUpperIfTxPduInfo[]
```

The value of ddTxPduHandleId defined in this data refers to the unique HandleID of the PDU as configured in PduR.
The data is necessary for transporting the confirmation of delivery for specific PDUs to PduR module.
Example: When transporting TxConfirmation data, received from CanIf module, to PduR module
PduR_CDD_RouterIFTxConfirmation((PduIdType)CDD_Router_GaaUpperIfTxPduInfo[HandleID].ddTxPduHandleId);

The aboveHandleID refers to the HandleID for Cdd Router Tx PDU of PduR, and is the same as the CddTxPduId called via FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_RouterIF_Transmit(PduIdType CddTxPduId, P2CONST(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr).
Utilize the MACRO generated under “IF Tx Upper I-PDU Handles” in the CDD_Router_Cfg.h file.

8.1.5.1 Additional implementation method for CddPduRLowerLayerTxPdu/CddIfUpperLayerPduRef configuration

The value for ddLoTargetPduHandleId refers to the unique HandleID configured for the PDU in 'CddSoAdUpperLayerTxPdu' or 'CddComIfUpperLayerTxPdu.'
If the ddLayerType is CddComIfUpperLayerTxPdu, it means 'CddComIfUpperLayerTxPdu' (Cdd Router Tx Pdu of CanIf) is connected to the CddIfUpperLayerPduRef of Cdd Router Tx Pdu in PduR. If CddSoAdUpperLayerTxPdu, 'CddSoAdUpperLayerTxPdu' (Cdd Router Tx Pdu of SoAd) is connected.
When using the above configuration, it is possible to implement a code that calls Transmit function in CanIf or SoAd Module via Cdd Router Tx Pdu Id of PduR from CDD_RouterIF_Transmit function, like below.

```
FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_RouterIF_Transmit(PduIdType CddTxPduId,
          P2CONST(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr)
{
   Std_ReturnType Ldd_ReturnValue = E_NOT_OK;

   if(CddTxPduId < CDD_ROUTER_MAX_UPPER_IF_TX_PDU_COUNT)
     {
        if (CDD_Router_GaaUpperIfTxPduInfo[CddTxPduId].ddLayerType == CddComIfUpperLayer)
        {
           #if (CDD_ROUTER_MAX_LOWER_IF_TX_PDU_COUNT > 0)
           Ldd_ReturnValue = CanIf_Transmit((PduIdType)(CDD_Router_GaaLowerIfTxPduInfo[
           CDD_Router_GaaUpperIfTxPduInfo[CddTxPduId].ddLoTargetPduHandleId].ddTxPduHandleId), PduInfoPtr);
           #endif
        }
        else
        {
           #if (CDD_ROUTER_MAX_LOWER_SOAD_IF_TX_PDU_COUNT > 0)
           Ldd_ReturnValue = SoAd_IfTransmit((PduIdType)(CDD_Router_GaaLowerIfTxPduInfo[
           CDD_Router_GaaUpperIfTxPduInfo[CddTxPduId].ddLoTargetPduHandleId].ddTxPduHandleId), PduInfoPtr);
           #endif
        }
     }

   return Ldd_ReturnValue;
}
```

8.1.6 Information on Rx PDU of PduR module connected with CDD Router

```
/* Structure for Upper Layer IF Rx PDU */
typedef struct STagCDD_Router_UpperIfRxPduInfo
{
  PduIdType ddRxPduHandleId;
  PduLengthType ddPduLength;
}CDD_Router_UpperIfRxPduInfo;

CONST(CDD_Router_UpperIfRxPduInfo, CDD_ROUTER_CONST) CDD_Router_GaaUpperIfRxPduInfo[]
```

The value of ddRxPduHandleId defined in this data refers to the unique HandleID for the PDU as configured in PduR.
The data is required for transporting the reception data for specific PDUs to PduR module.

Example: When transporting RxIndication data, received from CanIf module, to PduR module

PduR_CDD_RouterIFRxIndication((PduIdType)CDD_Router_GaaUpperIfRxPduInfo[HandleID].ddRxPduHandleId, PduInfoPtr);

The above HandleID refers to the Handle ID of PDU, as configured in CDD_ROUTER module, connected with Rx PDU of PduR module.
Utilize the MACRO generated under “IF Rx Upper I-PDU Handles” in the CDD_ROUTER_Cfg.h file.

8.1.7 Information on TP Rx PDU of PduR connected with CDD Router

```
/* Structure for Upper Layer TP Rx PDU */
typedef struct STagCDD_Router_UpperTpRxPduInfo
{
  PduIdType ddRxPduHandleId;
  PduLengthType ddPduLength;
}CDD_Router_UpperTpRxPduInfo;

CONST(CDD_Router_UpperTpRxPduInfo, CDD_ROUTER_CONST) CDD_Router_GaaUpperTpRxPduInfo[]
```

The value of ddRxPduHandleId defined in this data refers to the unique HandleID for the PDU as configured in PduR.
The data is required for transporting the reception data of specific PDUs to PduR module.

Example: When transporting the reception of TP message to PduR module
Below is an example that can substitute the CanTP actions with CDD_Router module.

```
PduLengthType TpLength = 0;
NotifResultType NotiResult = NTFRSLT_E_NOT_OK;
PduInfoType LddPduInfo;

/* Allocate values to length and data */
LddPduInfo.SduLength = PduInfoPtr->SduDataPtr[0];
LddPduInfo.SduDataPtr = (PduInfoPtr->SduDataPtr + 1);

/* Defined for the support of AUTOSAR 4.1.1 API. To use 4.1.1 API, should define as 1. */
/* Pre-compile option for 4.1.1 support */
#define CDD_Router_FOUR_ONE_ONE                    0

#if (CDD_Router_FOUR_ONE_ONE == 0)
/* Handle the reception of TP Rx Pdu data */
if(BUFREQ_OK == PduR_CDD_RouterTPStartOfReception
((PduIdType)CDD_Router_GaaUpperTpRxPduInfo[HandleID].ddRxPduHandleId,          LddPduInfo.SduLength,
(PduLengthType*)&TpLength))
{
if(BUFREQ_OK == PduR_CDD_RouterTPCopyRxData
((PduIdType)CDD_Router_GaaUpperTpRxPduInfo[HandleID].ddRxPduHandleId,                &LddPduInfo,
(PduLengthType*)&TpLength))
    {
        NotiResult = NTFRSLT_OK;
    }
}
#else
if(BUFREQ_OK == PduR_CDD_RouterTPStartOfReception
```

```
((PduIdType)CDD_Router_GaaUpperTpRxPduInfo[HandleID].ddRxPduHandleId,      (PduInfoType*)&LddPduInfo,
LddPduInfo.SduLength, (PduLengthType*)&TpLength))
{
if(BUFREQ_OK == PduR_CDD_RouterTPCopyRxData
((PduIdType)CDD_Router_GaaUpperTpRxPduInfo[HandleID].ddRxPduHandleId,                &LddPduInfo,
(PduLengthType*)&TpLength))
    {
        NotiResult = NTFRSLT_OK;
    }
}
#endif
PduR_CDD_RouterTPRxIndication((PduIdType)CDD_Router_GaaUpperTpRxPduInfo[HandleID].ddRxPduHandleId,
NotiResult);
```

The above HandleID refers to the HandleID of PDU as configured in CDD_Router module, connected with TP Rx PDU of PduR module.
Utilize the MACRO under “TP Rx Upper I-PDU Handles” in the CDD_Router_Cfg.h file.

8.1.8 Information on Rx PDU of PduR module connected with CDD IPC IF

```
/* Structure for IPC-If Rx PDU */
typedef struct
{
  PduIdType ddRxPduHandleId;
  PduLengthType ddPduLength;
  CddMsgIdType ddTargetAddressId;
}CDD_IPC_UpperIfRxPduInfo;

CONST(CDD_IPC_UpperIfRxPduInfo, CDD_ROUTER_CONST) CDD_IPC_GaaUpperIfRxPduInfo[]
```

The value of ddRxPduHandleId defined in this data refers to the unique HandleID for the PDU as configured in PduR.
The data is required for transporting the reception data for specific PDUs to PduR module.

Example: When transporting RxIndication data, received from IPC Driver, to PduR module
PduR_CDD_RouterIFRxIndication((PduIdType)CDD_IPC_GaaUpperIfRxPduInfo[HandleID].ddRxPduHandleId,
PduInfoPtr);

The above HandleID refers to the index for
CDD_IPC_GaaUpperIfRxPduInfo with the matching TargetAddressId for the data received from IPC.

8.1.9 Information on Tx PDU of PduR module, connected with CDD IPC IF

```
/* Structure for IPC-If Tx PDU */
typedef struct
{
  PduIdType ddTxPduHandleId;
  PduLengthType ddPduLength;
  CddMsgIdType ddTargetAddressId;
}CDD_IPC_UpperIfTxPduInfo;
```

CONST(CDD_IPC_UpperIfTxPduInfo, CDD_ROUTER_CONST) CDD_IPC_GaaUpperIfTxPduInfo[]
The value of ddTxPduHandleId defined in this data refers to the unique HandleID of the PDU as configured in PduR.
The data is required for transporting TxConfirmation data for specific PDUs to PduR module.

Example: When transporting the TxConfirmation data, received from IPC Driver (or calling self-generated TxConfirmation from CDD_Router) to PduR
PduR_CDD_IPC_IFTxConfirmation((PduIdType)CDD_IPC_GaaUpperIfTxPduInfo[HandleID].ddTxPduHandleId);

The above HandleID refers to the CddTxPduId called via FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_IPC_IF_Transmit(PduIdType CddTxPduId, P2CONST(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr) and the index for CDD_IPC_GaaUpperIfRxPduInfo with the matching ddTxPduHandleId.

8.1.10 Information on Rx PDU of PduR module connected with CDD IPC TP

```
/* Structure for IPC-Tp Rx PDU */
typedef struct
{
  PduIdType ddRxPduHandleId;
  PduLengthType ddPduLength;
  CddMsgIdType ddTargetAddressId;
}CDD_IPC_UpperTpRxPduInfo;
```

CONST(CDD_IPC_UpperTpRxPduInfo, CDD_ROUTER_CONST) CDD_IPC_GaaUpperTpRxPduInfo[]

The value of ddRxPduHandleId defined in this data refers to the unique HandleID for the PDU as configured in PduR.
The data is required for transporting the reception data for specific PDUs to PduR module.

Example: When transporting data received from IPC Driver to PduR and Dcm, the following three APIs must be called according to Transport Data Process.

- FUNC(BufReq_ReturnType, PDUR_CODE) PduR_CDD_IPC_TPStartOfReception (PduIdType TpRxPduId, PduLengthType TpSduLength,P2VAR(PduLengthType, AUTOMATIC, PDUR_DATA) bufferSizePtr)
- FUNC(BufReq_ReturnType, PDUR_CODE) PduR_CDD_IPC_TPCopyRxData(PduIdType TpRxPduId, P2CONST(PduInfoType, AUTOMATIC, PDUR_DATA) info, P2VAR(PduLengthType, AUTOMATIC, PDUR_DATA) bufferSizePtr)
- FUNC(void, PDUR_CODE) PduR_CDD_IPC_TPRxIndication(PduIdType TpRxPduId, NotifResultType Result)

TpRxPduId refers to CDD_IPC_GaaUpperTpRxPduInfo[HandleID].ddRxPduHandleId, and HandleID refers to the index of CDD_IPC_GaaUpperTpRxPduInfo with the matching TargetAddressID for the data received from IPC.

8.1.11 Information on Tx PDU of PduR module connected with CDD IPC TP

```
/* Structure for IPC-Tp Tx PDU */
typedef struct
{
```

```
  PduIdType ddTxPduHandleId;
  PduLengthType ddPduLength;
  CddMsgIdType ddTargetAddressId;
}CDD_IPC_UpperTpTxPduInfo;
```

CONST(CDD_IPC_UpperTpTxPduInfo, CDD_ROUTER_CONST) CDD_IPC_GaaUpperTpTxPduInfo[]
The value of ddTxPduHandleId defined in this data refers to the unique HandleID of the PDU as configured in PduR.
The data is required for transporting TxConfirmation data for specific PDUs to PduR module.

Example: When transporting the TxConfirmation data, received from IPC Driver (or calling self-generated TxConfirmation from CDD_Router) to PduR
PduR_CDD_IPC_TpTxConfirmation((PduIdType)CDD_IPC_GaaUpperTpTxPduInfo[HandleID].ddTxPduHandleId);

The above HandleID refers to CddTxPduId called via FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_IPC_TP_Transmit(PduIdType CddTxPduId, P2CONST(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr) and the index of CDD_IPC_GaaUpperTpRxPduInfo with matching ddTxPduHandleId.

8.1.12 Information on Tx PDU of CanTp module connected with CDD Router

```
/* Structure for Upper Layer IF CanTp Tx PDU */
typedef struct
{
  PduIdType ddTxPduHandleId;
  PduLengthType ddPduLength;

  /* PduId used for lower layer data transfer communication */
  PduIdType ddLoTargetPduHandleId;
  Cdd_Router_Layer ddLayerType;
}CDD_Router_UpperIfCanTpTxPduInfo;
```

CONST(CDD_Router_UpperIfCanTpTxPduInfo, CDD_ROUTER_CONST) CDD_Router_GaaUpperCanTpIfTxPduInfo[]

The value of ddTxPduHandleId defined in this data refers to the unique HandleID for the PDU as configured in CanTp.
The data is required for transporting the confirmation of delivery for specific PDUs to CanTp module.
Example: When transporting TxConfirmation data, received from CanIf module, to CanTp module
CanTp_TxConfirmation((PduIdType)CDD_Router_GaaUpperCanTpIfTxPduInfo[HandleID].ddTxPduHandleId);

The above HandleID refers to the HandleID of Cdd Router Tx Pdu in CanTp, and matches CddTxPduId called via FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_RouterIf_CanTpTransmit(PduIdType CddTxPduId, P2CONST(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr).
Utilize the MACRO generated under “CanTp IF Tx Upper I-PDU Handles” in the CDD_Router_Cfg.h file.

8.1.12.1 Additional implementation method for CddCanTpLowerLayerTxPdu/CddIfUpperLayerPduRef configuration

The value of ddLoTargetPduHandleId refers to the unique HandleID for the specific PDU, as configured in ‘CddComIfUpperLayerTxPdu.’
When ddLayerType is CddComIfUpperLayer, it means ‘CddComIfUpperLayerTxPdu’ (Cdd Router Tx Pdu of CanIf)

is connected to the CddIfUpperLayerPduRef of Cdd Router Tx Pdu in CanTp.
When using the above configuration, a code to call Transmit function of CanIf module via CDD_RouterIF_CanTpTransmit function with Cdd Router Tx Pdu Id of CanTp may be implemented.

```
FUNC(Std_ReturnType, CDD_ROUTER_CODE) CDD_RouterIF_CanTpTransmit(PduIdType CddTxPduId,
        P2CONST(PduInfoType, AUTOMATIC, CDD_ROUTER_APPL_DATA)PduInfoPtr)
{
  Std_ReturnType Ldd_ReturnValue = E_NOT_OK;

  #if(CDD_ROUTER_MAX_UPPER_CANTP_IF_TX_PDU_COUNT > 0)
  if(CddTxPduId < CDD_ROUTER_MAX_UPPER_CANTP_IF_TX_PDU_COUNT)
  {
    #if(CDD_ROUTER_MAX_LOWER_IF_TX_PDU_COUNT > 0)
    if (CDD_Router_GaaUpperCanTpIfTxPduInfo[CddTxPduId].ddLayerType == CddComIfUpperLayer)
    {
      Ldd_ReturnValue = CanIf_Transmit((PduIdType)(CDD_Router_GaaLowerIfTxPduInfo[
      CDD_Router_GaaUpperCanTpIfTxPduInfo[CddTxPduId].ddLoTargetPduHandleId].ddTxPduHandleId),
PduInfoPtr);
    }
    #endif /* #if(CDD_ROUTER_MAX_LOWER_IF_TX_PDU_COUNT > 0) */
  }
  #endif /* #if(CDD_ROUTER_MAX_UPPER_CANTP_IF_TX_PDU_COUNT > 0) */

  return Ldd_ReturnValue;
}
```

8.1.13 Information on Rx PDU of CanTp module connected with CDD Router

```
/* Structure for Upper Layer CanTp IF Rx PDU */
typedef struct
{
  PduIdType ddRxPduHandleId;
  PduLengthType ddPduLength;
}CDD_Router_UpperCanTpIfRxPduInfo;

CONST(CDD_Router_UpperCanTpIfRxPduInfo, CDD_ROUTER_CONST) CDD_Router_GaaUpperCanTpIfRxPduInfo[]
```

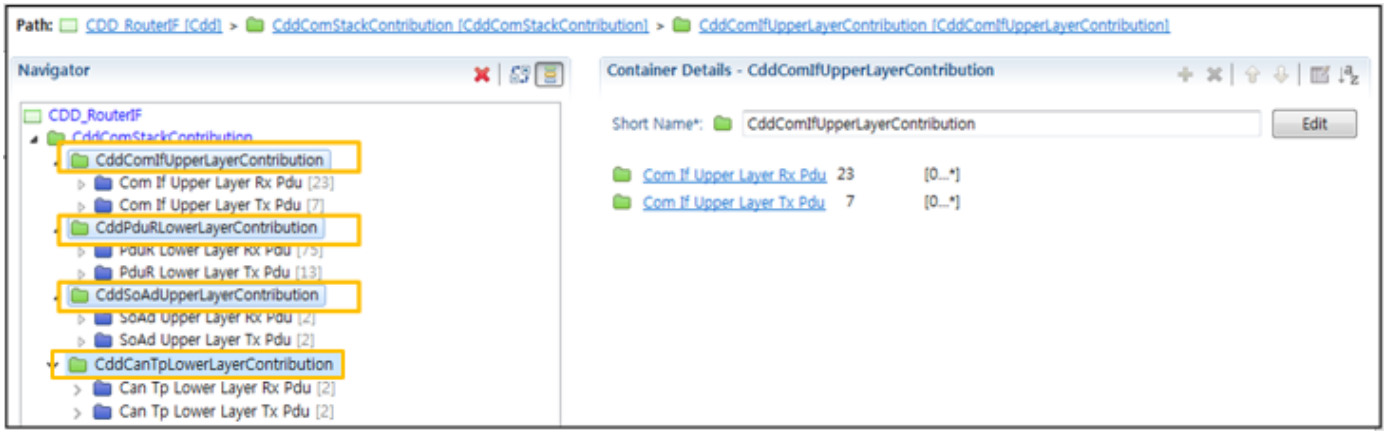
The value of ddRxPduHandleId defined in this data refers to the unique HandleID for the specific PDU, as configured in CanTp.
The data is required for transporting the reception data of specific PDUs to CanTp module.

Example: When transporting the RxIndication data, received from CanIf module, to CanTp module
CanTp_RxIndication((PduIdType)CDD_Router_GaaUpperCanTpIfRxPduInfo[HandleID].ddRxPduHandleId, PduInfoPtr);

The above HandleID refers to the HandleID of PDU as configured in CDD Router, connected to the Rx PDU of CanTp module.
Utilize the MACRO under “CanTp IF Rx Upper I-PDU Handles” in the CDD_Router_Cfg.h file.

8.2 Connection relations with surrounding modules

8.2.1 Example for configuring PDU connection for CDD Router



CddPduRLowerLayerContribution

- Communication configuration related Containers (Tx, Rx) for the lower layer to PduR module

CddCanTpLowerLayerContribution

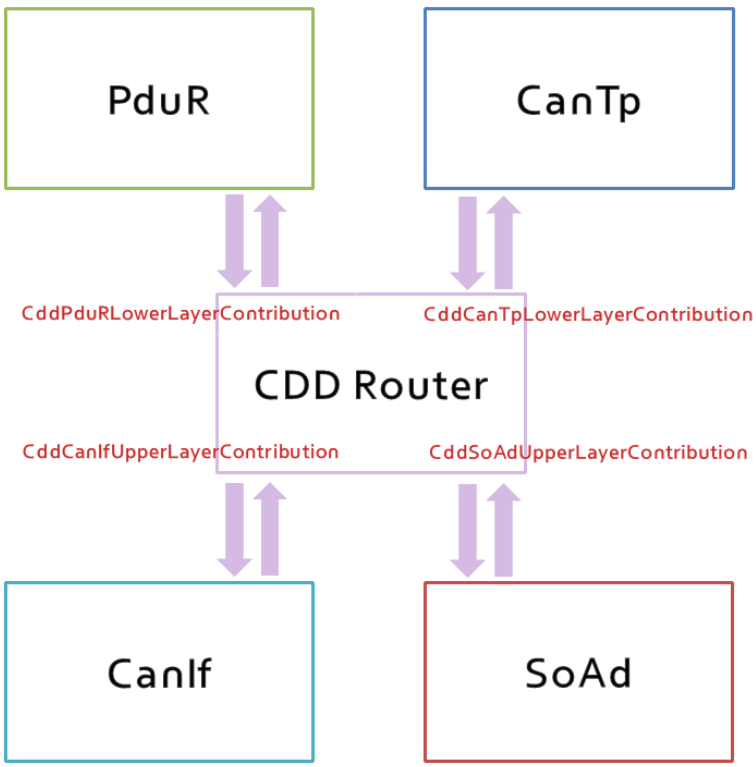
- Communication configuration related Containers (Tx, Rx) for the lower layer to CanTp module

CddComIfUpperLayerContribution

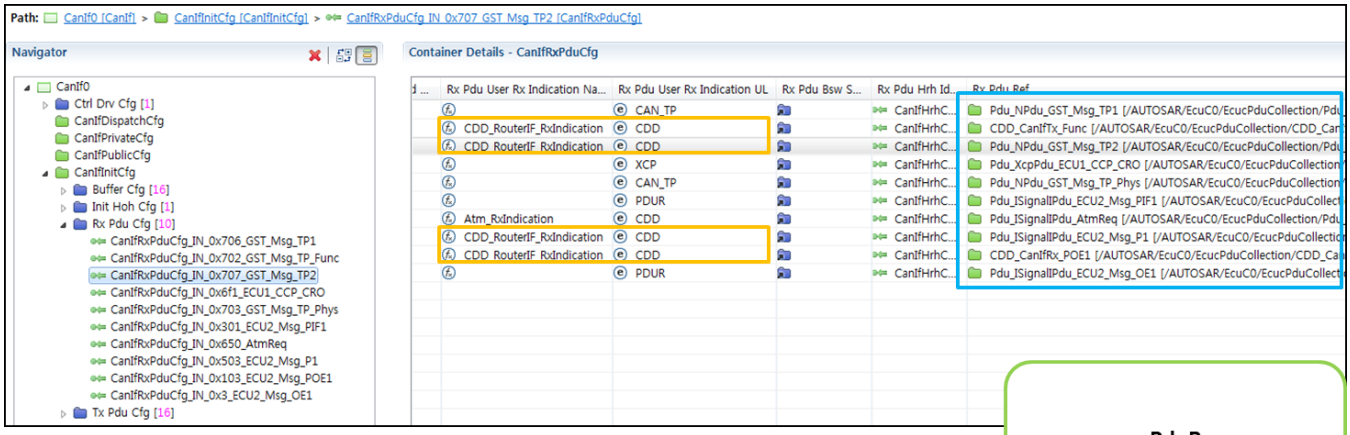
- Communication configuration related Containers (Tx, Rx) for the upper layer to CanIf module

CddSoAdUpperLayerContribution

- Communication configuration related Containers (Tx, Rx) for the upper layer to SoAd module



8.2.2 Example for configuring PDU connection for CanIf module

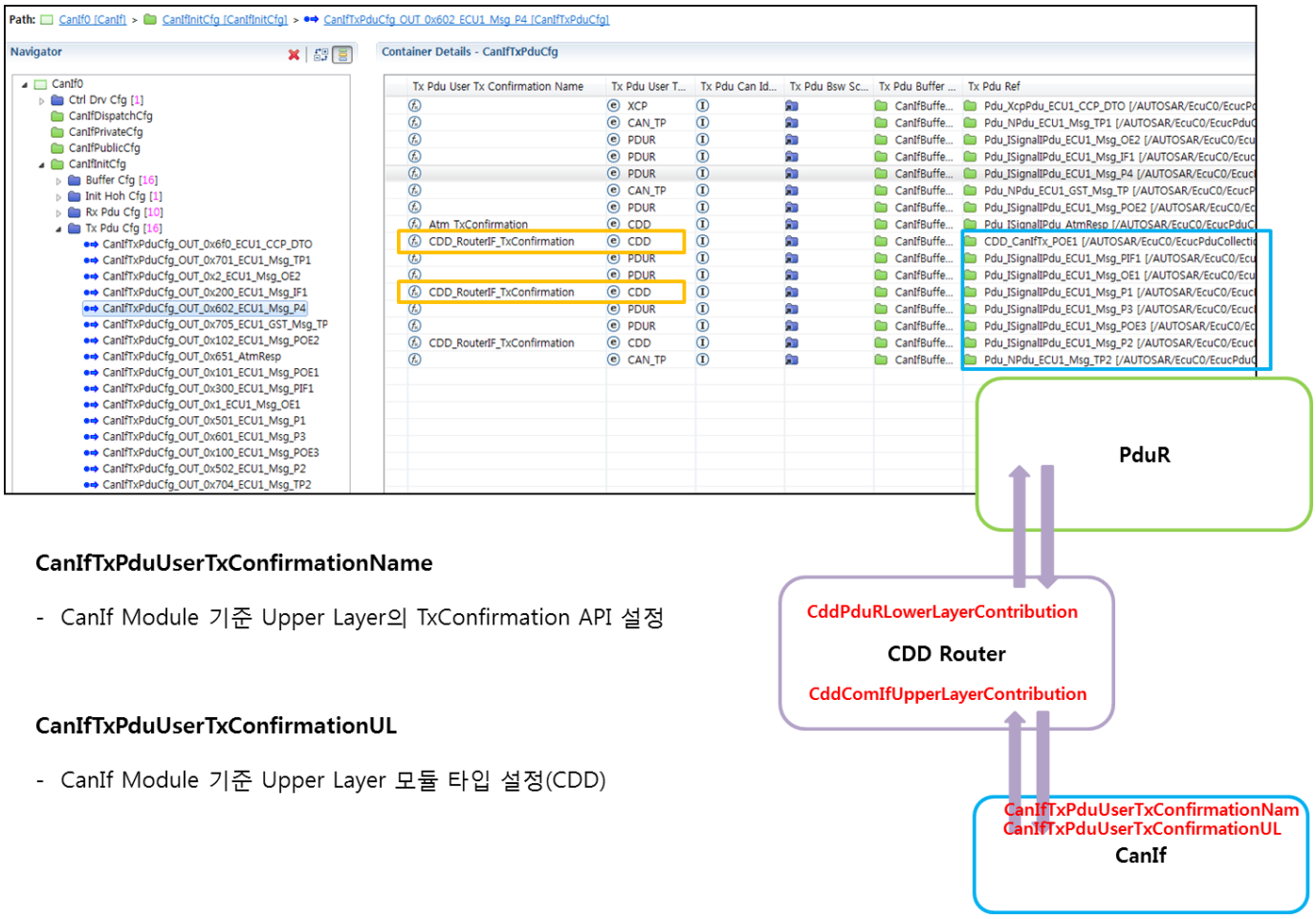


CanIfRxPduUserRxIndicationName

- CanIf Module 기준 Upper Layer의 RxIndication API 설정

CanIfRxPduUserRxIndicationUL

- CanIf Module 기준 Upper Layer 모듈 타입 설정(CDD)



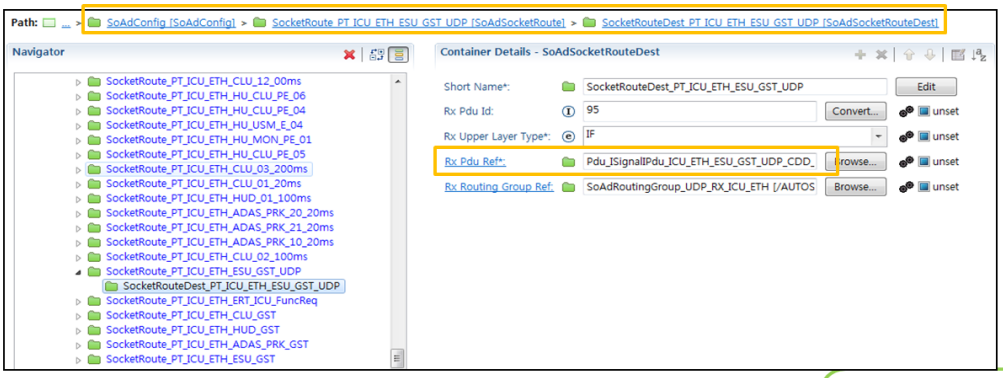
CanIfTxPduUserTxConfirmationName

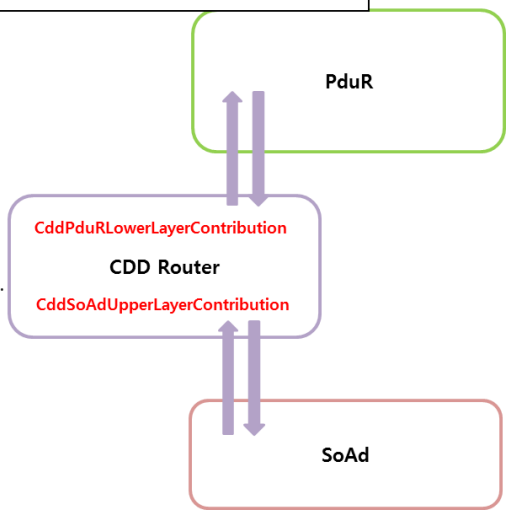
- CanIf Module 기준 Upper Layer의 TxConfirmation API 설정

CanIfTxPduUserTxConfirmationUL

- CanIf Module 기준 Upper Layer 모듈 타입 설정(CDD)

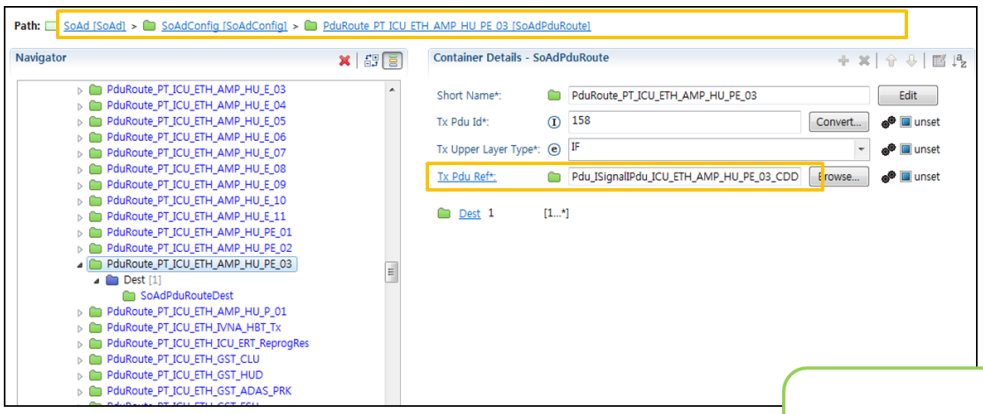
8.2.3 Example for configuring PDU connection for SoAd module

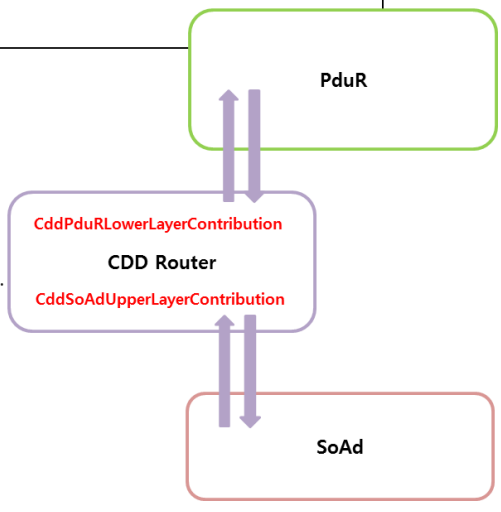




SoAdRxPduRef

- SoAd Module 기준 CDD Router와 Rx 연결관계를 확인하는 설정
- 해당 설정은 PduR과 연결관계를 확인할 때에도 사용한다.
- 동일한 PDU reference를 CDD Router와 PduR에 설정하지 않도록 한다.

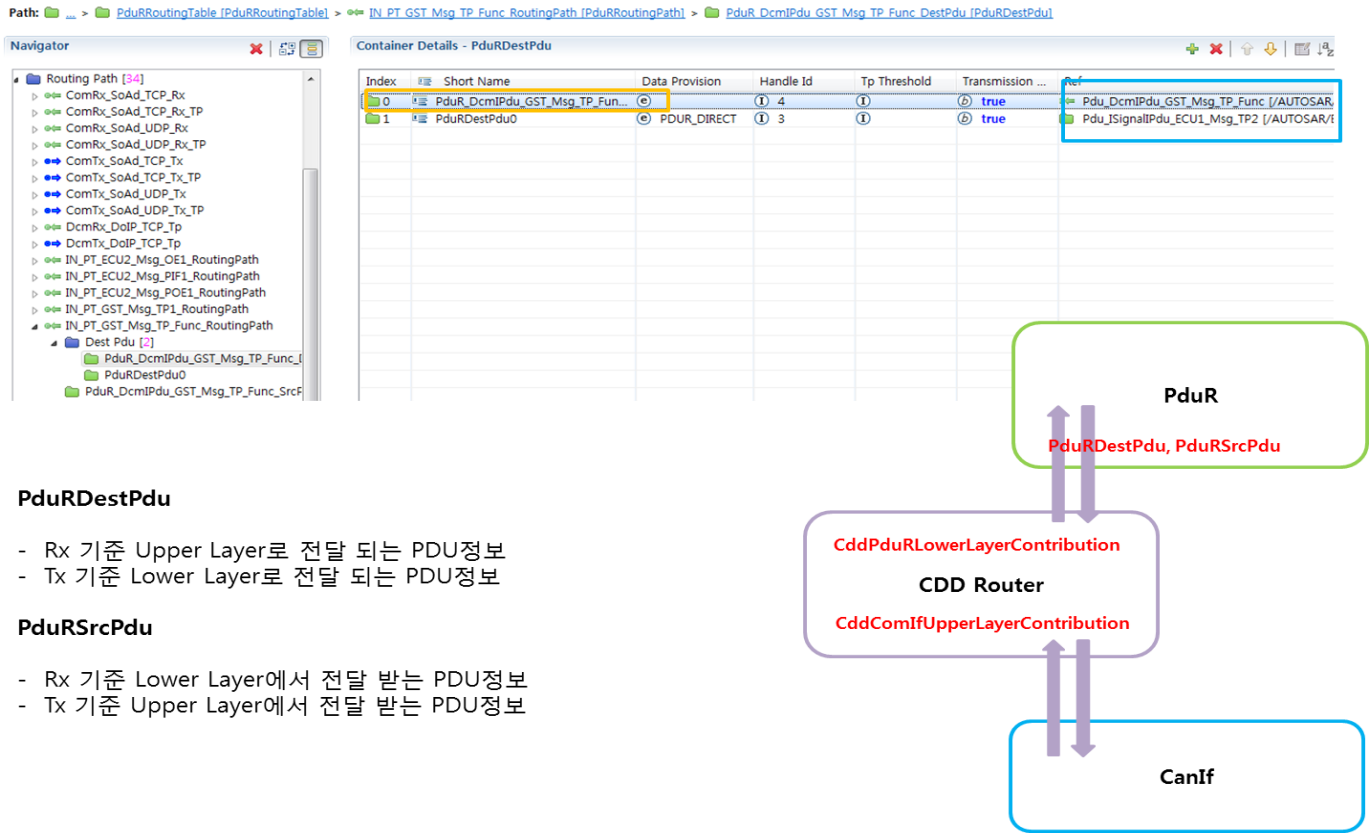




SoAdTxPduRef

- SoAd Module 기준 CDD Router와 Tx 연결관계를 확인하는 설정
- 해당 설정은 PduR과 연결관계를 확인할 때에도 사용한다.
- 동일한 PDU reference를 CDD Router와 PduR에 설정하지 않도록 한다.

8.2.4 Example for configuring PDU connection for PduR module



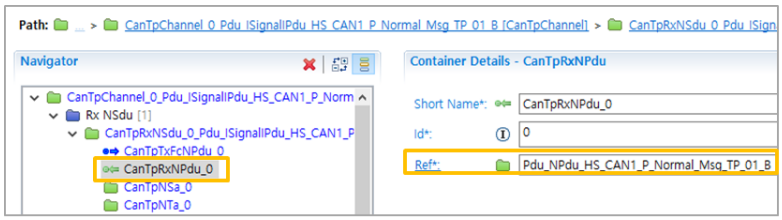
PduRDestPdu

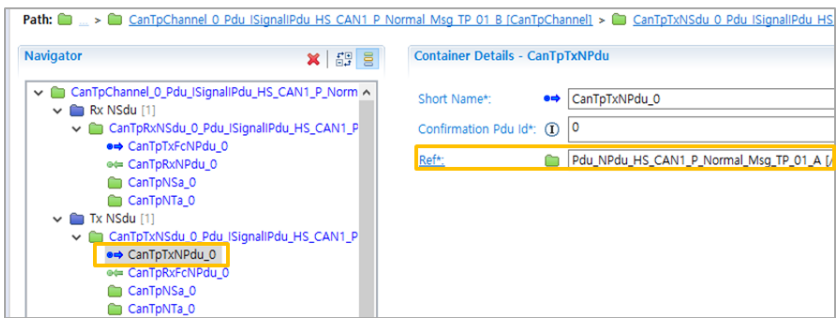
- Rx 기준 Upper Layer로 전달 되는 PDU정보
- Tx 기준 Lower Layer로 전달 되는 PDU정보

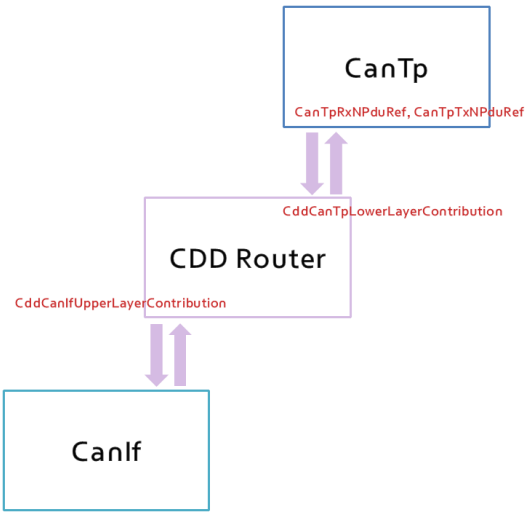
PduRSrcPdu

- Rx 기준 Lower Layer에서 전달 받는 PDU정보
- Tx 기준 Upper Layer에서 전달 받는 PDU정보

8.2.5 Example for configuring PDU connection for CanTp module







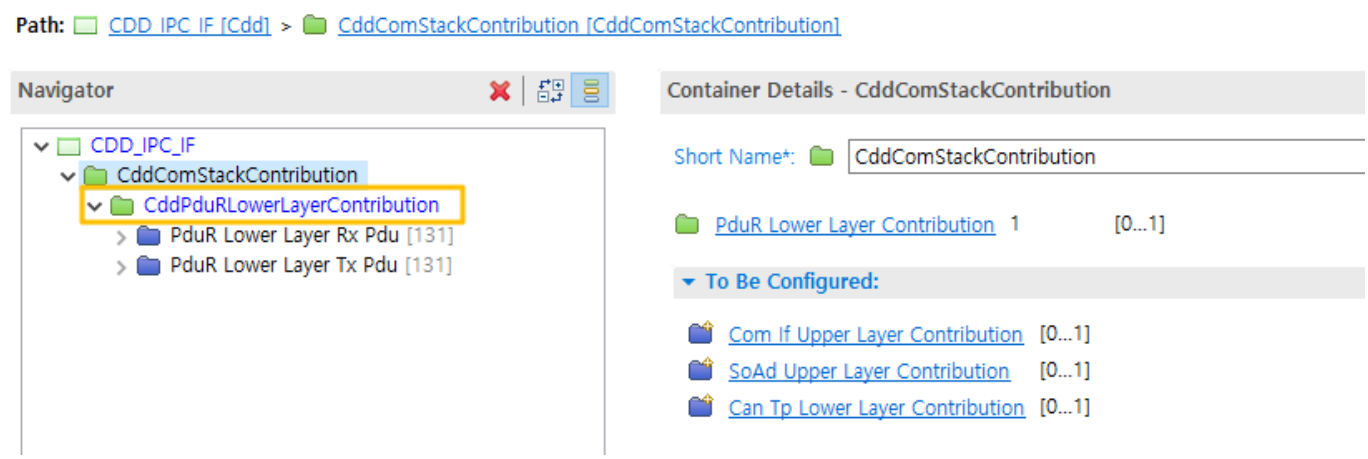
CanTpRxNPduRef

- Configuration for checking Rx connection relationship with CDD Router from CanTp

CanTpTxNPduRef

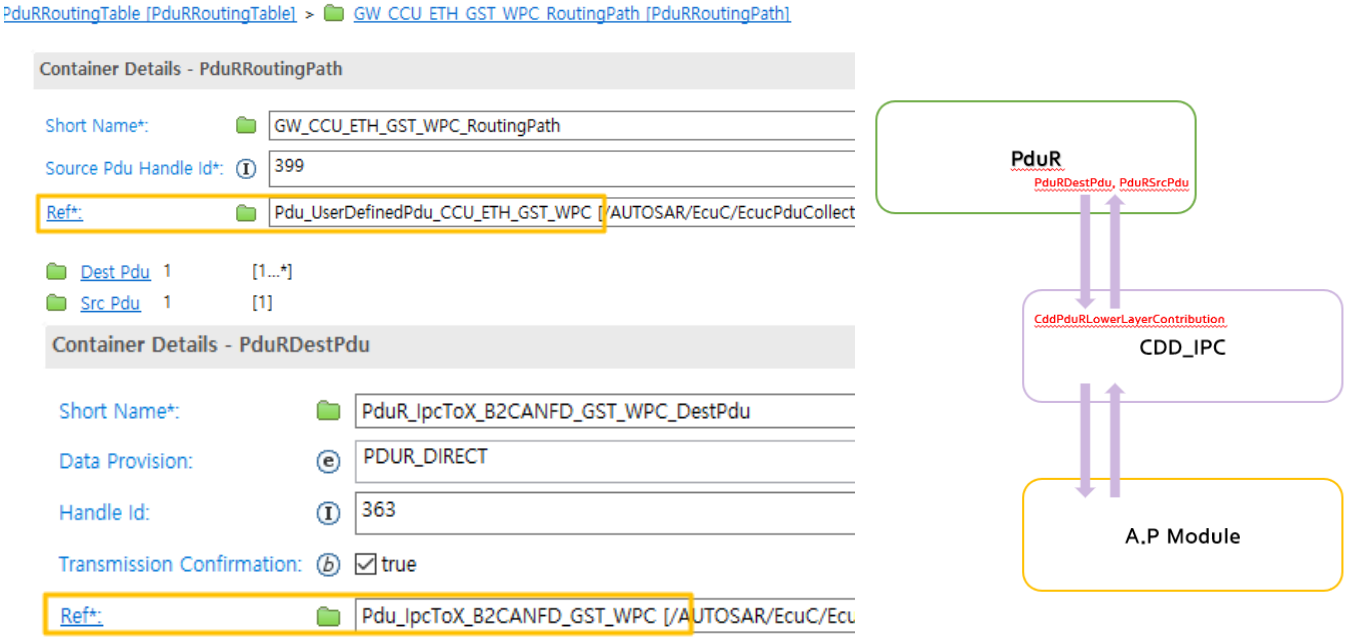
- Configuration for checking Tx connection relationship with CDD Router from CanTp

8.2.6 Example for configuring PDU connection for CDD IPC module



CddPduRLowerLayerContribution

- Communication configuration related Containers (Tx, Rx) for the lower layer to PduR module



PduRDestPdu

- PDU information passed to Upper Layer based on Rx
- PDU information passed to Lower Layer based on Tx

PduRSrcPdu

- PDU information received by Lower Layer based on Rx
- PDU information received by Upper Layer based on Tx

8.3 CDD_Router Related DB Import Configuration Guide

Options for DB Import change depending on the version--before or after 2017.
See PduR Module User Manual for more information.

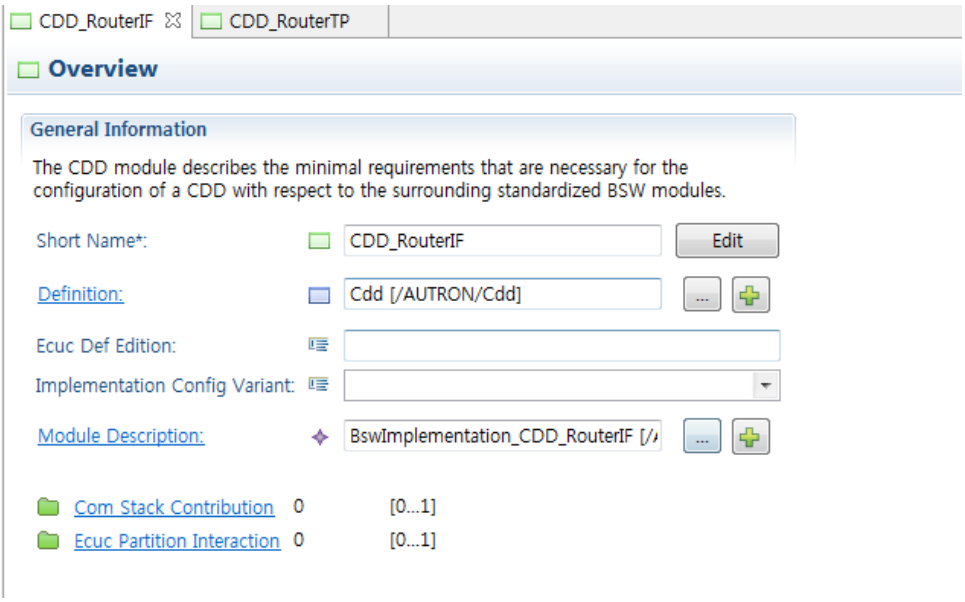
Refer to Confluence Gateway Guide 2.1.4 for IPC import.
https://swpfaq.hyundai-autoever.com/x/s79KAg

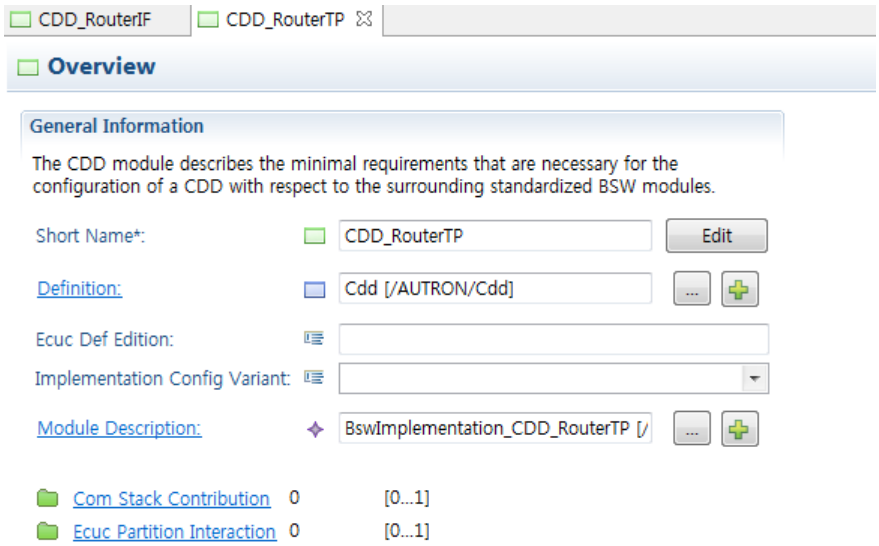
8.4 Integration Guide for CDD_Router Module

8.4.1 Adding Modules

Check Definition and Module Description configuration of the module.
Select the CDD module as defined in
Static_Code₩Modules₩b_autosar_cdd_CDD_Router₩generator₩CDD_Router_ECU_Configuration_PDF.arxml.
(Applies the same for CDD_RouterIF, CDD_RouterTP, CDD_IPC_IF, and CDD_IPC_TP)

Select both CDD_RouterIF and CDD_RouterTP of Bswmd_CDD_Router.arxml.



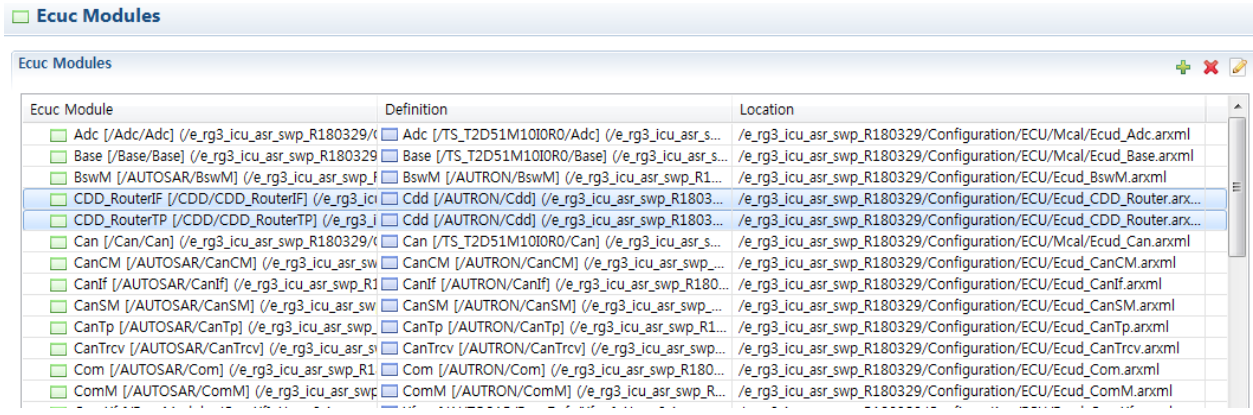


Update MemMap.h file

There is no need to copy if there is a CDD_Router related item in the Memmap.h file currently in use. If not, request to the platform manager.

Add CDD_RouterIF and CDD_RouterTP modules to EcucValueCollection

Module	EcucValueCollection
Configuration file	ECUCD_EcucValueCollection.arxml
Path	EcucValueCollection/Ecuc Modules



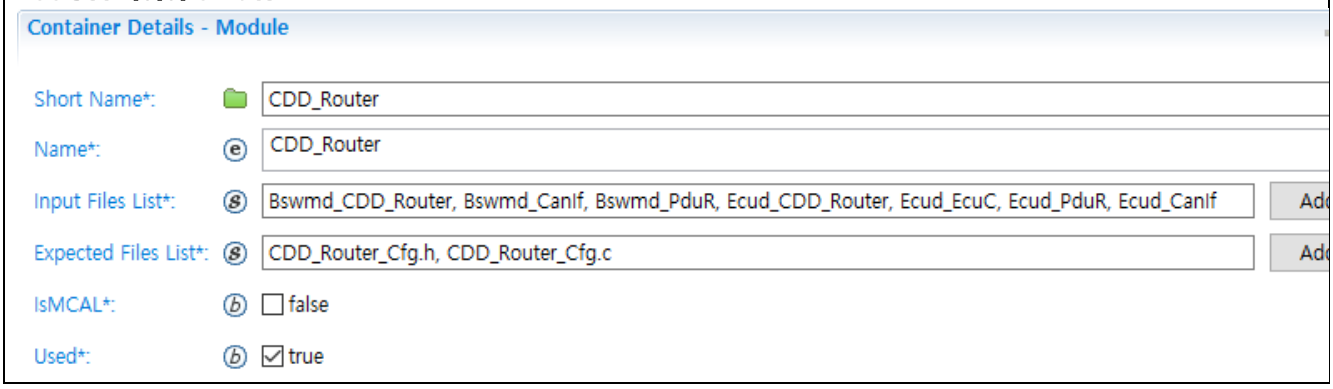
Not required to add EcuMDriverInitItem to EcuM.

Toolset 1.3.3. or older

```
#GenerateCDD_Router ######################################################################
    moduleName = 'CDD_Router'
    expectedFiles = [
        os.path.join('inc', moduleName + '_Cfg.h'),
        os.path.join('src', moduleName + '_Cfg.c'),
    ]
    generateBSW = env.GenerateBSW(
        target=expectedFiles,
        source=[
            Bswmd_CDD_Router,
            Ecud_CanIf, Ecud_PduR, Ecud_EcuC, Ecud_CDD_Router, Ecud_SoAd,
        ],
        BSW_GENERATOR=CDD_Router,
```

```
    )
    generated += [generateBSW]
    Alias('Generate' + moduleName, generateBSW)
    ##############################################################################################
```

Toolset 2.0.0. or later

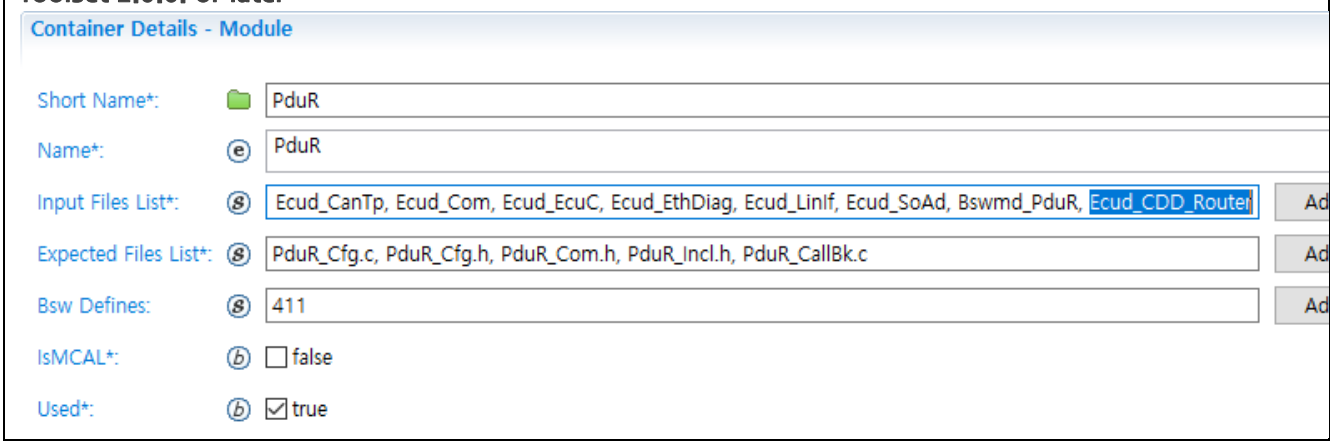


You may remove Ecud_SoAd if there are no modules for SoAd.
Add Ecud_CDD_Router as the generator input value for the associated modules of PduR, CanIf and SoAd.

Toolset 1.3.3. or older

```
# GeneratePduR ###############################################################################
    moduleName = 'PduR'
    expectedFiles = [
        os.path.join('inc', moduleName + '_CanIf.h'),
        os.path.join('inc', moduleName + '_Cfg.h'),
        os.path.join('inc', moduleName + '_Com.h'),
        os.path.join('inc', moduleName + '_Incl.h'),
        os.path.join('inc', moduleName + '_CanTp.h'),
        os.path.join('inc', moduleName + '_Dcm.h'),
        os.path.join('inc', moduleName + '_LinIf.h'),
        os.path.join('src', moduleName + '_CallBk.c'),
        os.path.join('src', moduleName + '_Cfg.c'),
    ]
    generateBSW = env.GenerateBSW(
        target=expectedFiles,
        source=[Bswmd_PduR, Bswmd_Com, Bswmd_CanIf, Bswmd_CanTp, Bswmd_Dcm,
                Bswmd_LinIf, Bswmd_IpduM, Bswmd_CDD_Router, Bswmd_EthDiag,
                Ecud_PduR, Ecud_EcuC, Ecud_Com, Ecud_CanIf,
                Ecud_LinIf, Ecud_Dcm, Ecud_CanTp, Ecud_IpduM, Ecud_SoAd, Ecud_CDD_Router, Ecud_EthDiag,
        ],
        BSW_GENERATOR=PduR,
    )
    generated += [generateBSW]
    Alias('Generate' + moduleName, generateBSW)
    ##############################################################################################
```

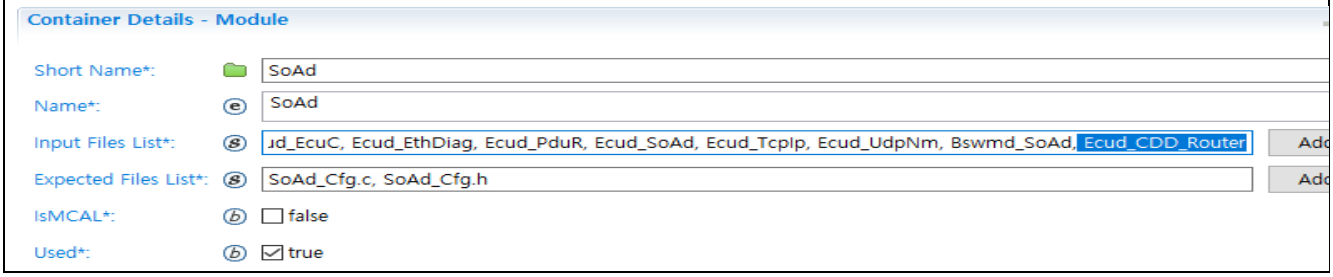
Toolset 2.0.0. or later



Toolset 1.3.3. or older

```
# GenerateSoAd ###################################################################
    moduleName = 'SoAd'
    expectedFiles = [
          os.path.join('inc', moduleName+'_Cfg.h'),
          os.path.join('src', moduleName+'_Cfg.c'),
          ]
    generateBSW =   env.GenerateBSW(
            target=expectedFiles,
            source = [
                Bswmd_SoAd, Bswmd_PduR,
                #Bswmd_DoIP,
                #Bswmd_Sd,
                Ecud_SoAd, Ecud_EcuC, Ecud_Tcpip, Ecud_PduR, Ecud_UdpNm, Ecud_EthDiag, Ecud_CDD_Router,
                #Ecud_DoIP,
                #Ecud_Sd, #Ecud_UdpNm,
                ],
            BSW_GENERATOR = SoAd,
            )
    generated += [generateBSW]
    Alias('Generate' + moduleName, generateBSW)
    ###########################################################################################
```

Toolset 2.0.0. or later

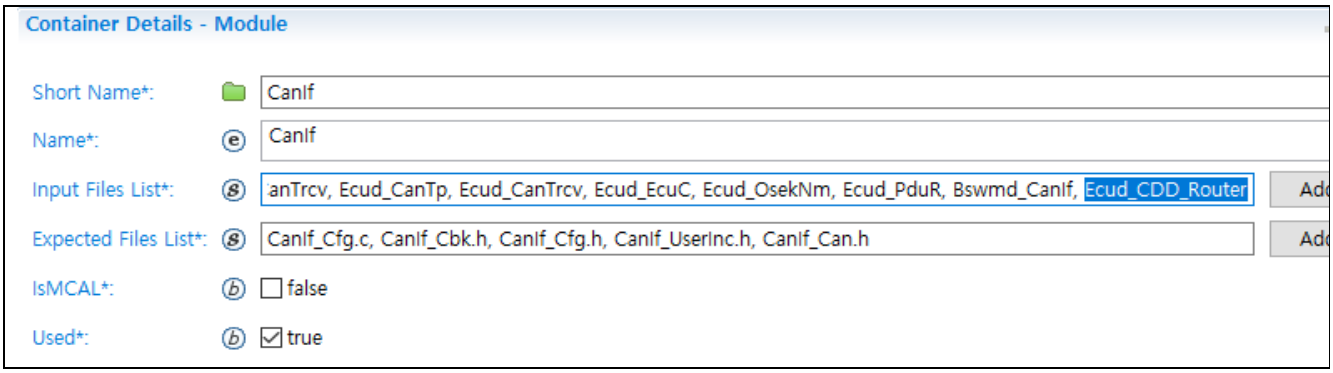


Toolset 1.3.3. or older

```
#                                                                                    GenerateCanIf
#############################################################################
    moduleName = 'CanIf'
    expectedFiles = [
          os.path.join('inc', moduleName + '_Can.h'),
          os.path.join('inc', moduleName + '_Cbk.h'),
          os.path.join('inc', moduleName + '_Cfg.h'),
          os.path.join('inc', moduleName + '_UserInc.h'),
          os.path.join('src', moduleName + '_Cfg.c'),
    ]
    generateBSW = env.GenerateBSW(
          target=expectedFiles,
          source=[
              Bswmd_CanIf, Bswmd_Can, Bswmd_CanTrcv, Bswmd_CDD_Router, #Bswmd_IpduM,
              Ecud_CanIf, Ecud_EcuC, Ecud_CanTp, Ecud_PduR,
              Ecud_Nm, Ecud_OsekNm, Ecud_CanTrcv,
              Ecud_Can, Ecud_CDD_Router, #Ecud_IpduM,
              Ecud_IdsM,
          ],
          BSW_GENERATOR=CanIf,
    )
    generated += [generateBSW]
    Alias('Generate' + moduleName, generateBSW)

#############################################################################################
###
```

Toolset 2.0.0. or later



Add "CDD_Router" to /CanIf/CanIfPublicCfg/CanIfPublicCddHeaderFile for CanIf

Module configuration is automatically harmonized. Refer to the user manual for manual configuration.

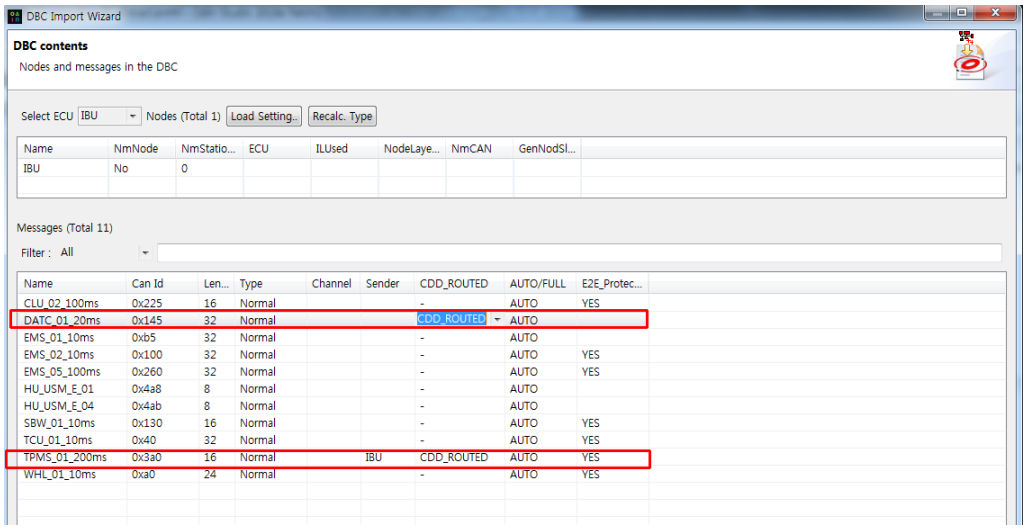
8.4.2 Removing Modules

CDD_Router related content removed from generate.py
Remove "CDD_Router" from /CanIf/CanIfPublicCfg/CanIfPublicCddHeaderFile of CanIf
Remove CDD_Router_IF and CDD_Router_TP from PduR
Remove CDD_Router from Ecuc Modules in EcucValueCollection
Execute Import / Harmonizes

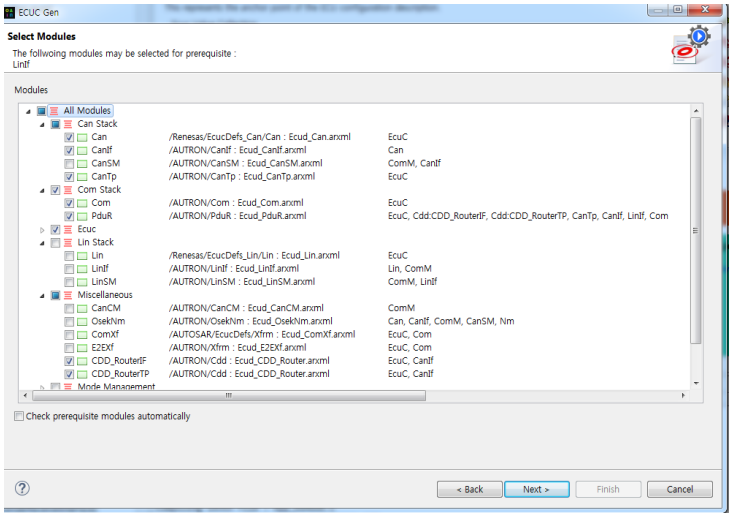
8.4.3 Configuration Guide

Typical message path when using CDD_Router module looks like the following.
Tx Message : COM -> PDUR -> CDD_Router -> CanIf
Rx Message : CanIf -> CDD_Router -> PduR -> COM

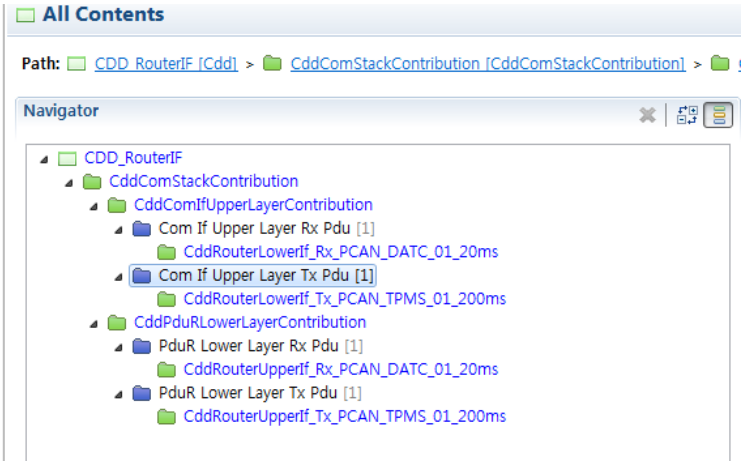
8.4.3.1 Select CDD_ROUTED attribute for Tx and Rx messages selected for DB Import



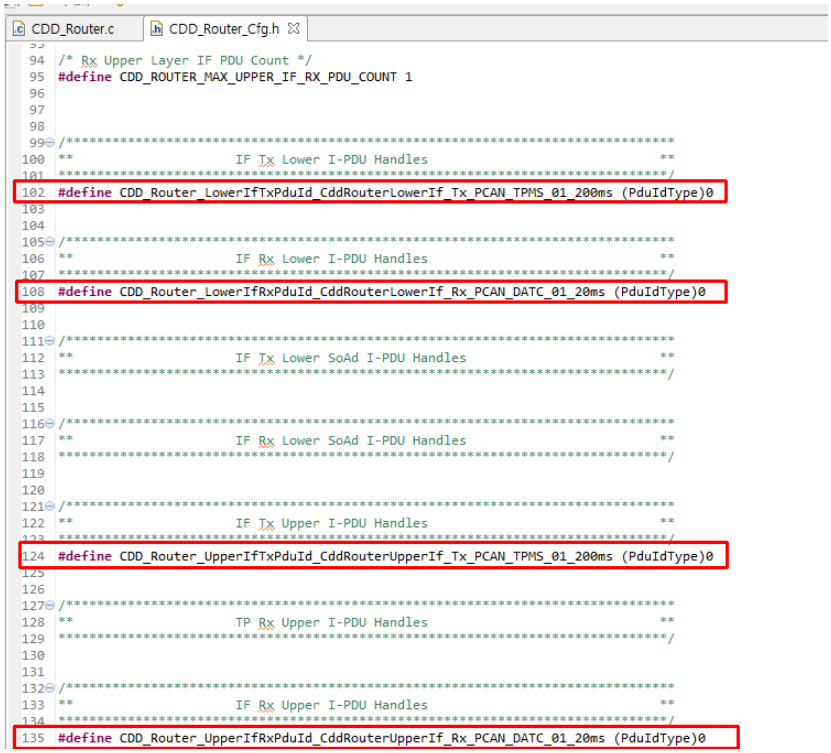
8.4.3.2 Select Can Stack, Com Stack, Ecuc, CDD_RouterIf, and CDD_RouterTP to harmonize



8.4.3.3 Upper Layer and Lower Layer should each have Tx and Rx settings in CDD_RouterIF configuration as the result of harmonizing



8.4.3.4 Should check if the below code is created in CDD_Router_Cfg.h, following generation



CDD_Router.c | CDD_Router_Cfg.h

```
/* Rx Upper Layer IF PDU Count */
#define CDD_ROUTER_MAX_UPPER_IF_RX_PDU_COUNT 1



/**********************************************************************
**                      IF Tx Lower I-PDU Handles                    **
**********************************************************************/
#define CDD_Router_LowerIfTxPduId_CddRouterLowerIf_Tx_PCAN_TPMS_01_200ms (PduIdType)0


/**********************************************************************
**                      IF Rx Lower I-PDU Handles                    **
**********************************************************************/
#define CDD_Router_LowerIfRxPduId_CddRouterLowerIf_Rx_PCAN_DATC_01_20ms (PduIdType)0


/**********************************************************************
**                      IF Tx Lower SoAd I-PDU Handles               **
**********************************************************************/


/**********************************************************************
**                      IF Rx Lower SoAd I-PDU Handles               **
**********************************************************************/


/**********************************************************************
**                      IF Tx Upper I-PDU Handles                    **
**********************************************************************/
#define CDD_Router_UpperIfTxPduId_CddRouterUpperIf_Tx_PCAN_TPMS_01_200ms (PduIdType)0


/**********************************************************************
**                      TP Rx Upper I-PDU Handles                    **
**********************************************************************/


/**********************************************************************
**                      IF Rx Upper I-PDU Handles                    **
**********************************************************************/
#define CDD_Router_UpperIfRxPduId_CddRouterUpperIf_Rx_PCAN_DATC_01_20ms (PduIdType)0
```

8.4.3.5 Build and Test Functionality

1) Perform Tx validation by confirming the below via testing after implementing the application for the signal transfer
Call in the order of Com_Transmit() -> PduR_IfTransmit() -> CDD_RouterIF_Transmit() -> CanIf_Transmit() and check the message from CANoe

2) Transmit a message that contains the signal from CANoe for Rx validation and confirm the below
Call in the order of CanIf_RxIndication() -> CDD_RouterIF_RxIndication() -> PduR_CDD_RouterIFRxIndication() -> Com_RxIndication and check the message from CANoe