ControlDesk

Bus Navigator

For ControlDesk 7.4

Release 2021-A – May 2021

dSPACE

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: http://www.dspace.com/go/locations
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
  Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: http://www.dspace.com/go/supportrequest. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit http://www.dspace.com/go/patches for software updates and patches.

Important Notice

This publication contains proprietary information that is protected by copyright. All rights are reserved. The publication may be printed for personal or internal use provided all the proprietary markings are retained on all printed copies. In all other cases, the publication must not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of dSPACE GmbH.

© 2010 - 2021 by:
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

This publication and the contents hereof are subject to change without notice.

AUTERA, ConfigurationDesk, ControlDesk, MicroAutoBox, MicroLabBox, SCALEXIO, SIMPHERA, SYNECT, SystemDesk, TargetLink and VEOS are registered trademarks of dSPACE GmbH in the United States or other countries, or both. Other brand names or product names are trademarks or registered trademarks of their respective companies or organizations.

Contents

About This Document 9

New Features 11

New Bus Navigator Features (ControlDesk 7.4) 11

Basics and Instructions 13

Introduction to the Bus Navigator 14
- Overview of the Bus Navigator 14
- Basics on Bus Configurations 17
- Bus Configuration Structure in the Bus Navigator Tree 19
- Bus Navigator Demo 24

Basics on Working with the Bus Navigator 34
- How to Display a Bus Configuration with the Bus Navigator 34
- Accessing Bus Communication with a Bus Monitoring Device 37
- Basics on Ethernet PDUs in the Bus Navigator 41
- Activating Communication Matrices for Blockset-Based Bus Communication 44
- Basics on Global Time Synchronization 45

Working with Bus Instruments 48

General Handling of Bus Instruments 48
- How to Configure Layout Names 48
- How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs 50
- How to Configure a Bus Instrument 55

Instruments for Bus Communication Modeled with the Bus Manager 57
- Overview of Available Bus Instrument Types 58
- How to Configure a TX Message/Frame/PDU and Transmit It on the Bus 59
- How to Specify Global Transmission Settings for a CAN or LIN Controller 61

Instruments for Bus Communication Modeled with a Blockset 62
- Overview of Available Bus Instrument Types 63
- How to Configure a TX Message/Frame/PDU and Transmit It on the Bus 64

How to Create a Bus Instrument (Gateway Type) for a CAN Controller Gateway ... 67
How to Specify Global Transmission Settings for a CAN or FlexRay Controller ... 69
How to Display the Error Status for all the RX PDUs of a FlexRay Controller ... 71

Instruments Related to Bus Monitoring Devices ... 73
Overview of Available Bus Instrument Types ... 73
How to Configure a CAN TX Message and Transmit It on the Bus ... 74

Monitoring and Logging Bus Communication ... 76
Basics on Monitoring, Logging, and Filtering Bus Communication ... 77
Dependencies Regarding Monitoring, Logging, and Filtering Bus Communication ... 80
How to Monitor Bus Communication ... 82
How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN) ... 85
Example of Specifying Filters for Bus Monitoring or Logging (CAN, LIN) ... 88
Basics on Filtering Ethernet Bus Communication ... 91
Examples of Filtering Ethernet Bus Communication ... 94
How to Log Raw Bus Communication Data ... 95
How to Display Bus Statistics ... 97

Replaying CAN and Ethernet Bus Communication ... 99
Basics on Replaying CAN and Ethernet Bus Communication ... 99
How to Replay Logged CAN or Ethernet Bus Communication ... 101

Reference Information 103

Bus Instruments ... 104

Bus Instruments (Bus Manager) ... 104

Global Instruments ... 105
Bus Instrument (Global Layout Type) ... 105

CAN Instruments ... 106
Bus Instrument (Inspection Type for CAN) ... 107
Bus Instrument (Manipulation Type for CAN) ... 110
Bus Instrument (RX Type for CAN) ... 114
Bus Instrument (TX Status Type for CAN) ... 119
Bus Instrument (TX Type for CAN) ... 121

LIN Instruments ... 125
Bus Instrument (Inspection Type for LIN) ... 126
Bus Instrument (Manipulation Type for LIN) ... 128

Bus Instrument (RX Type for LIN)..... 131
Bus Instrument (TX Status Type for LIN)..... 134
Bus Instrument (TX Type for LIN)..... 136

Bus Instruments (Blockset)..... 139

CAN Instruments..... 139
Bus Instrument (Gateway Type for CAN)..... 140
Bus Instrument (RX Type for CAN)..... 142
Bus Instrument (TX Status Type for CAN)..... 145
Bus Instrument (TX Type for CAN)..... 147

FlexRay Instruments..... 154
Bus Instrument (Error Status Type for FlexRay)..... 155
Bus Instrument (RX Type for FlexRay)..... 157
Bus Instrument (TX Type for FlexRay)..... 160
Bus Instrument (TX Status Type for FlexRay)..... 166

LIN Instruments..... 169
Bus Instrument (RX Type for LIN)..... 169
Bus Instrument (TX Type for LIN)..... 171

Bus Instruments (Related to Bus Monitoring Devices)..... 174

CAN Instruments..... 174
Bus Instrument (RX Type for CAN)..... 175
Bus Instrument (TX Type for CAN)..... 176

Ethernet Instruments..... 178
Bus Instrument (RX Type for Ethernet)..... 178

LIN Instruments..... 180
Bus Instrument (RX Type for LIN)..... 180

Bus Navigator Commands and Dialogs..... 182
Activate (Database)..... 185
Activate (Logger)..... 186
Activate (Monitor)..... 187
Activate (Replay)..... 188
Add (CAN/Ethernet) Replay..... 189
Add (CAN/LIN/Ethernet) Filter..... 190
Add (CAN/LIN/Ethernet) Logger..... 191
Add (CAN/LIN/Ethernet) Monitor..... 193
Add/Edit Filter Dialog (CAN/LIN)..... 195
Add/Edit Filter Dialog (Ethernet)..... 197
Add/Edit Logger Dialog..... 199
Add/Edit Monitor Dialog..... 201
Add/Edit Replay Dialog..... 203

Bus Navigator ... 204
Bus Navigator Page ... 207
Case-Sensitive Search (Bus Navigator) ... 213
Collapse All ... 213
Configure Column Sets Dialog (Monitoring List) ... 214
Customize Columns (Bus Navigator) ... 215
Decoding View ... 217
Delete (Filter) ... 219
Delete (Logger) ... 220
Delete (Monitor) ... 221
Delete (Replay) ... 222
Delete Column (Monitoring List) ... 223
Edit (Filter) ... 223
Edit (Logger) ... 224
Edit (Monitor) ... 225
Edit (Replay) ... 226
Expand All ... 227
Generate Error Status Layout ... 228
Generate Gateway Layout ... 229
Generate Inspection Layout ... 230
Generate Layout ... 231
Generate Manipulation Layout ... 231
Generate RX Layout ... 232
Generate TX Layout ... 234
Generate TX Status Layout ... 235
Highlight Monitor ... 236
Import Mappings Dialog ... 237
Load All Bus Signals ... 238
Locate Generated Layout ... 239
Monitoring List ... 240
Save ... 249
Set as Favorite Column Set (Monitoring List) ... 249
Start (Logger) ... 250
Start (Monitor) ... 251
Start (Replay) ... 252
Unload All Bus Signals ... 253

Automation 255

Programming ControlDesk Automation ... 256
Automating the Bus Navigator ... 256

Bus Navigator and Bus Instrument Handling..... 260
Bus Navigator and Bus Instrument Handling Interfaces..... 260
Bus Navigator-Related Interfaces..... 260
Bus Instrument-Related Interfaces..... 261

Troubleshooting 263

Synchronizing Bus Monitoring Data with Recorded Data..... 263

Limitations 267

Limitations for the Bus Navigator..... 267

Glossary 269

Index 307

About This Document

Content

This document introduces you to ControlDesk's Bus Navigator.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Special folders

Some software products use the following special folders:

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

```
%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>
```

or

```
%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>
```

Documents folder A standard folder for user-specific documents.

```
%USERPROFILE%\Documents\dSPACE\<ProductName>\
<VersionNumber>
```

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

```
%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\
<ProductName>
```

Accessing dSPACE Help and PDF Files

After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com/go/help.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the icon in dSPACE Help. The PDF opens on the first page.

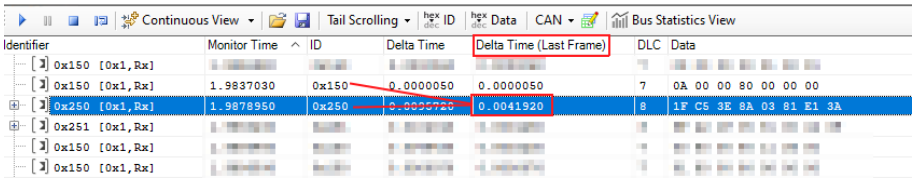
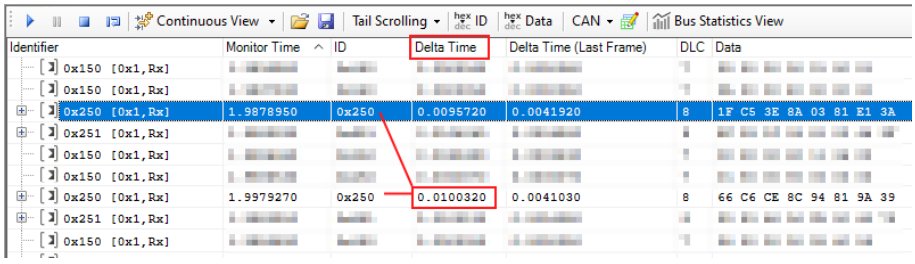
New Features

New Bus Navigator Features (ControlDesk 7.4)

CAN: Display of the delta time (last frame)

In addition to the *delta time*, the ControlDesk Monitoring list now lets you display the *delta time (last frame)* for CAN messages.

The following table compares the two Monitoring list columns:

Column	Description
Delta Time (Last Frame) (New in ControlDesk 7.4; available for CAN)	Displays the difference between the receipt time of a message instance and the receipt time of the previous message instance, *irrespective of the message ID* but *received by the same communication controller*. The following illustration shows an example: 
Delta Time	Displays the difference between the receipt time of a message instance and the receipt time of the previous message instance *with the same ID received by the same communication controller*. The following illustration shows an example: 

For more information, refer to Monitoring List on page 240.

Context menu extensibility of Bus Navigator tree nodes

ControlDesk now lets you extend the context menu of Bus Navigator tree nodes, such as communication clusters or network nodes. You can extend the context menu by using the ControlDesk automation interface.

For general information on extending the context menu of elements, refer to Extending the Context Menu of Elements (ControlDesk Customization 🕮).

Basics and Instructions

Where to go from here

Information in this section

Introduction to the Bus Navigator 14
The Bus Navigator is a ControlDesk component that lets you handle CAN messages, LIN frames, FlexRay PDUs, and Ethernet packets.

Basics on Working with the Bus Navigator 34
You can open instruments via the Bus Navigator to handle the bus communication. The Bus Navigator also lets you monitor, log, and replay bus communication.

Working with Bus Instruments 48
The Bus Navigator provides Bus Instruments to view and/or manipulate the bus communication of a bus configuration.

Monitoring and Logging Bus Communication 76
The Bus Navigator lets you monitor and log bus communication. You can also specify filters to limit the monitored or logged bus communication.

Replaying CAN and Ethernet Bus Communication 99
For CAN and Ethernet bus systems, the Bus Navigator lets you replay logged bus communication.

Introduction to the Bus Navigator

Introduction

The Bus Navigator is a ControlDesk component that lets you handle CAN messages, LIN frames, FlexRay PDUs, and Ethernet packets.

Where to go from here

Information in this section

Overview of the Bus Navigator..... 14
The Bus Navigator lets you display and experiment with bus configurations.

Basics on Bus Configurations..... 17
The bus communication that is displayed in the Bus Navigator can be configured via the Bus Manager or via RTI blocksets, for example.

Bus Configuration Structure in the Bus Navigator Tree..... 19
The Bus Navigator tree displays all the controllers, databases, messages/frames/PDUs etc. for a bus configuration hierarchically.

Bus Navigator Demo..... 24
Demonstrates the ControlDesk Bus Navigator features (monitoring, logging, replay, message transmission). The demo is based on a simulated bus interface.

Overview of the Bus Navigator

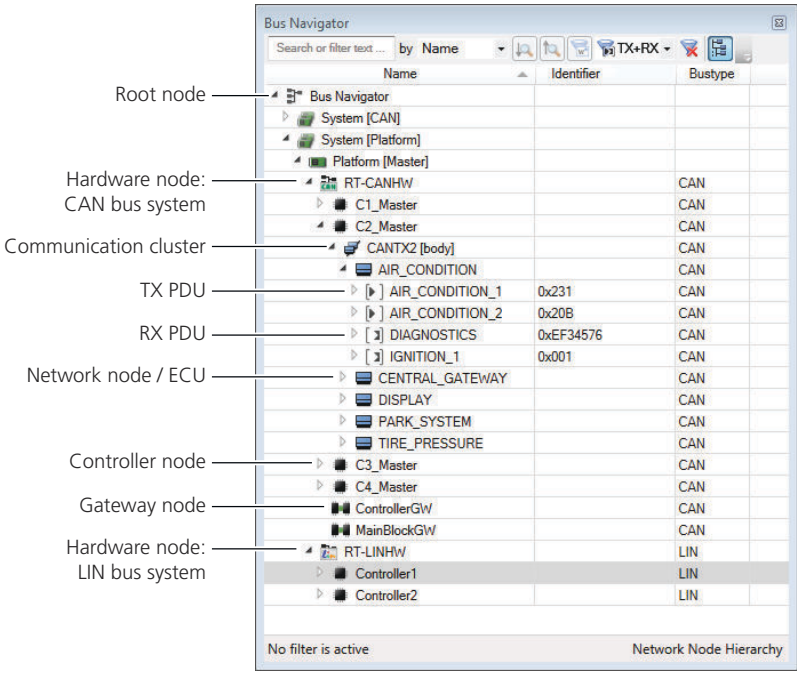
Introduction

The Bus Navigator lets you display and experiment with bus configurations.

Bus configurations

A bus configuration is the configuration of all the controllers, communication clusters, and messages, frames, and PDUs of a specific communication bus such as CAN.

When you add a variable description that references a bus configuration to a platform or a device in the active experiment, the Bus Navigator displays the bus configuration in the Bus Navigator tree. The following illustration shows an example.



Instruments of the Bus Navigator

The Bus Navigator provides Bus Instruments to view and manipulate messages, frames, and PDUs of a bus configuration.

You can create tailor-made Bus Instruments that are automatically placed on a separate layout.

Below is a list of available Bus Instrument types. Bus Instrument types that are only available for certain bus types are named accordingly.

Bus Instrument (Error status type for FlexRay) For FlexRay, the Bus Navigator provides the Bus Instrument (Error status type), which lets you display the error value and the slot status for all the PDUs of a FlexRay communication cluster.

Bus Instrument (Gateway type for CAN) For CAN, the Bus Navigator provides the Bus Instrument (Gateway type) to view gateways and their messages. The Bus Instrument (Gateway type) can display standard CAN messages and messages compliant with the J1939 standard.

Bus Instrument (Global layout type for CAN and LIN) The Bus Instrument (Global layout type) lets you enable or disable the bus communication of a bus configuration fully or partly.

Bus Instrument (Inspection type for CAN and LIN) The Bus Instrument (Inspection type) lets you display information on a PDU that has been configured for inspection purposes.

Bus Instrument (Manipulation type for CAN and LIN) The Bus Instrument (Manipulation type) lets you change the settings of a PDU that has been configured for manipulation purposes.

Bus Instrument (RX type) The Bus Instrument (RX type) lets you display information on RX messages.

Bus Instrument (TX status type for CAN, LIN, and FlexRay) The Bus Instrument (TX status type) lets you enable or disable communication controllers and manipulate all their TX messages at the same time.

Bus Instrument (TX type for CAN, LIN, and FlexRay) The Bus Instrument (TX type) lets you manipulate and transmit TX messages.

Monitoring, logging, and filtering bus communication

The Bus Navigator lets you monitor and log bus communication. Depending on the model basis and the bus type, you can do this via a dSPACE platform or a bus monitoring device in connection with either a PC-based interface or a bus interface of a dSPACE platform.

You can reduce the amount of monitored/logged data by specifying filters. The filters can apply to the entire system, a specific bus system, or a specific controller. Each filter can contain one or more filter items.

For more information on monitoring, logging, and filtering bus communication, refer to Monitoring and Logging Bus Communication on page 76.

Replaying logged bus communication

For CAN and Ethernet bus systems, the Bus Navigator lets you replay the raw data of logged bus communication. You can also monitor the data that is being replayed.

For more information, refer to Replaying CAN and Ethernet Bus Communication on page 99.

Features specific to CAN

Free raw messages Free raw messages are TX and RX messages that are independent of the database file. They always consist of 8 (CAN) or 64 (CAN FD) 8-bit signals. You can completely modify free raw messages, including their identifiers and message lengths. You can also change the ID format of a free raw message. The selected format affects the number of raw bytes to be sent for the message, and specifies the message's bit rate switch (BRS) value (CAN FD only).

Frames with no specified sender and receiver For bus communication that is accessed via the CAN Bus Monitoring device, ControlDesk lets you import and display CAN frames for which no sender and receiver is specified in the communication matrix. The frames are displayed in the Bus Navigator tree under the DS_UnknownSender and DS_UnknownReceiver network nodes. You can create the Bus Instrument (TX type) and the Bus Instrument (RX type) for them.

Support of CAN FD protocol The Bus Navigator supports the CAN FD (CAN with Flexible Data Rate) protocol on the following dSPACE platforms or devices:

- MicroAutoBox platform
- MicroAutoBox III platform
- DS1006 Processor Board platform
- DS1007 PPC Processor Board platform

- SCALEXIO platform
- VEOS platform
- CAN Bus Monitoring device in connection with one of the following interfaces:
  - CAN interface of a DCI-CAN2 or the DCI-CAN/LIN1 interface
  - CAN interface of a SCALEXIO platform
  - CAN interface of VEOS platform

Compared with the classic CAN protocol, CAN FD comes with an increased bandwidth for serial communication.

Support of global time synchronization The Bus Navigator lets you display and manipulate CAN PDUs that contain time synchronization information.

Related topics

Basics

Basics on Bus Configurations....17
Basics on Global Time Synchronization....45
Basics on Monitoring, Logging, and Filtering Bus Communication....77
Basics on Replaying CAN and Ethernet Bus Communication....99
Bus Configuration Structure in the Bus Navigator Tree....19
Working with Bus Instruments....48

HowTos

How to Display a Bus Configuration with the Bus Navigator....34

Basics on Bus Configurations

Introduction

The bus communication that is displayed in the Bus Navigator can be configured via the Bus Manager or via RTI blocksets, for example.

Bus configurations in simulation applications

For dSPACE platforms such as SCALEXIO or VEOS, the bus configuration is part of the simulation application running on that platform. Depending on the specific simulation platform, the bus configuration can be modeled with one of the following tools:

Modeling Tool	Simulation Application Types	Simulation Platform
Bus Manager in ConfigurationDesk[1)]	RTA	▪ SCALEXIO ▪ MicroAutoBox III
	OSA	VEOS
Bus Manager (stand-alone)[2)]	OSA	VEOS

Modeling Tool	Simulation Application Types	Simulation Platform
RTI CAN MultiMessage Blockset[3)]	RTA	▪ SCALEXIO ▪ MicroAutoBox II ▪ MicroLabBox ▪ DS1006-/DS1007-based systems
FlexRay Configuration Blockset[4)]	RTA	▪ MicroAutoBox III ▪ SCALEXIO
RTI FlexRay Configuration Blockset[4)]	RTA	▪ MicroAutoBox II ▪ DS1006-/DS1007-based systems
RTI LIN MultiMessage Blockset[5)]	RTA	▪ SCALEXIO ▪ MicroAutoBox II ▪ DS1006-/DS1007-based systems

1) Refer to Workflow for Using the Bus Manager and Configuring Bus Communication for Real-Time Applications (ConfigurationDesk Bus Manager Implementation Guide 🕮).
2) Refer to Workflow for Configuring Bus Communication and Generating Bus Simulation Containers (Bus Manager (Stand-Alone) Implementation Guide 🕮).
3) Refer to Basics on Working with the RTI CAN MultiMessage Blockset (RTI CAN MultiMessage Blockset Reference 🕮).
4) Refer to dSPACE FlexRay Configuration Package (FlexRay Configuration Tool Guide 🕮).
5) Refer to How to Create a Model Using the RTI LIN MultiMessage Blockset (RTI LIN MultiMessage Blockset Reference 🕮).

Variable descriptions When you build a simulation application, a variable description file that references the related bus configuration is generated. Each time you build the application again, a new variable description file is generated. You can adapt the bus configurations displayed in the Bus Navigator by reloading the variable description file in the **Project** controlbar. Refer to Reload (Variable Description) (ControlDesk Variable Management 🕮).

Related topics

Basics

Bus Configuration Structure in the Bus Navigator Tree..... 19

HowTos

How to Display a Bus Configuration with the Bus Navigator..... 34

Bus Configuration Structure in the Bus Navigator Tree

Introduction

The bus configuration is structured hierarchically in the Bus Navigator tree. The tree displays all the controllers, databases, messages/frames/PDUs etc. for a bus configuration. Depending on the bus configuration, the following nodes can be displayed.

Node symbols

Every element of the bus configuration displayed in the Bus Navigator tree is visualized by a main symbol. The main symbol can be supplemented by one or more property symbols.

Main Symbol	Node	Possible Property Symbols	Combination Examples
	Bus Navigator (root node)	-	-
	System	-	-
	Platform/Monitoring device	-	-
	Hardware node: CAN bus system	-	-
	Hardware node: Ethernet bus system	-	-
	Hardware node: FlexRay bus system	-	-
	Hardware node: LIN bus system	-	-
	Controller	-	-
	Communication cluster	Enabled	Enabled communication cluster
	Network node	Monitored	Monitored network
	Ethernet socket	Server Client	Server socket
	Ethernet Channel	-	-
	FlexRay channel	-	-
	Message/frame/PDU	TX RX TX/RX Not received by the ECU	TX/RX message/frame/PDU
	PDU	Static TX Static RX Static TX/RX (red) Dynamic TX (red) Dynamic RX	Static sync TX PDU

Main Symbol	Node	Possible Property Symbols	Combination Examples
		Startup sync Sync Monitored Not received by the ECU	
/	Container IPDU (TX/RX)	-	-
/	Multiplexed IPDU (TX/RX)	-	-
/	Secured IPDU	-	-
	Sub-PDU/Part of a multiplexed IPDU	Static TX Static RX Static TX/RX (red) Dynamic TX (red) Dynamic RX Startup sync Sync Monitored Not received by the ECU	Static RX Sub-PDU, not received by the ECU
	Signal	Static TX Static RX (red) Dynamic TX (red) Dynamic RX Belonging to a startup sync PDU Belonging to a sync PDU Monitored (gray) Mode dependent (red) Mode manager/multiplexer switch Not received by the ECU	Monitored static sync TX signal
	Signal group	-	-
	Global time domain	TX RX	Time master of the global time domain
	Gateway	Enabled	Enabled gateway
	Filter	-	-
	Monitor	Started	Started monitor
/	Logger (active / inactive)	Started	

Main Symbol	Node	Possible Property Symbols	Combination Examples
/	Replay (active / inactive)	Started	Started logger Started replay

Bus Navigator

The Bus Navigator node is the root node of a bus configuration in the Bus Navigator tree and displays the system(s) used.

- Platform, SP-System: A single processor system is used.
- Platform, MP-System: A multiprocessor system is used.
- Monitoring device: A PC-based interface is used.

System

The system node contains the processor node(s) to which a bus configuration is loaded.

Platform/Monitoring device

The platform/device node displays the bus configuration that is loaded to the platform/device. It contains the hardware node.

Hardware node: CAN bus system

This node contains all the CAN controllers of a bus configuration.

Hardware node: Ethernet bus system

This node contains all the Ethernet controllers.

Hardware node: FlexRay bus system

This node contains all the FlexRay controllers of a bus configuration.

Hardware node: LIN bus system

This node contains all the LIN controllers of a bus configuration.

Controller

Each controller of a bus configuration or monitoring system is displayed in a controller node.

The controller node of a bus configuration contains all the relevant communication matrices and configurations of the controller used for RX and TX messages, frames, and PDUs. The controller connects the related platform or device to the bus.

Communication cluster	A communication cluster represents a communication network of network nodes that are connected to the same physical channels and share the same bus protocol and address range. The name of the related communication matrix is shown in brackets.
Network node	The network node represents the communication parts of an ECU that are relevant for one communication cluster.
Ethernet socket	An Ethernet socket represents the endpoint of Ethernet communication between ECUs.
Ethernet channel	This node contains all the Ethernet PDUs that are sent and received by a network node of an Ethernet controller. An Ethernet network node can contain several Ethernet channels. Multiple PDUs can be sent and received over each channel.
FlexRay channel	This node contains all the FlexRay PDUs that are sent and received by an ECU that is connected to a FlexRay controller. A FlexRay network node can contain several FlexRay channels. Multiple PDUs can be sent and received over each channel.
Message/frame/PDU	A communication matrix contains messages/frames/PDUs. In a bus configuration based on a simulation application, the number and configuration of messages/frames/PDUs depends on the communication matrix and the configuration of the related system model. You can view and/or manipulate messages/frames/PDUs using the Bus Navigator: ▪ RX messages/frames/PDUs are received by an ECU. You can view the messages/frames/PDUs. ▪ TX messages/frames/PDUs are transmitted by an ECU. You can view, configure, and transmit TX messages/frames/PDUs. ▪ TX/RX messages/frames/PDUs are both TX and RX messages/frames/PDUs.
Container IPDU	A Container IPDU contains one or more other IPDUs (i.e., contained IPDUs). When a container IPDU is mapped to a frame, all its contained IPDUs are included in that frame as well.
Multiplexed IPDU	A multiplexed IPDU is an IPDU that consists of one dynamic part, a selector field, and one optional static part. Multiplexing is used to transport varying ISignal IPDUs via the same bytes of a multiplexed IPDU.

Secured IPDU	A secured IPDU is an IPDU that secures the payload of another PDU (i.e., authentic IPDU) for secure onboard communication (SecOC). The secured IPDU contains the authentication information that is used to secure the authentic IPDU's payload. To use secure onboard communication, SecOC support must be enabled in the bus configuration. For more information, refer to: ▪ Bus Manager in ConfigurationDesk: Implementing Secure Onboard Communication in Executable Applications (ConfigurationDesk Bus Manager Implementation Guide 🕮) ▪ Bus Manager (stand-alone): Implementing Secure Onboard Communication in Executable Applications (Bus Manager (Stand-Alone) Implementation Guide 🕮)
Signal	A signal that is defined for a message/frame/PDU. The signals can be manipulated via the TX Bus Instrument of a TX or TX/RX message/frame/PDU. Mode signals and their mode-dependent signals can be distinguished by their icons. For details on mode signals and mode-dependent signals, refer to Bus Instrument (TX Type for CAN) on page 147.
Signal group	A signal group refers to a set of signals that must always be kept together. It is used to guarantee the atomic transfer of AUTOSAR composite data types.
Global time domain	A single ECU can belong to several global time domains at once and it can have a different role in each domain, i.e., it can be a time master in one domain while being a time slave in another. The role of an ECU in a given global time domain is indicated by the ▸ and ⌟ property symbols.
Gateway (only for CAN)	When you exchange messages between two CAN buses using the RTICANMM Gateway block in the Simulink model, the gateway node is available in the Bus Navigator tree. If configured in RTICANMM, you can open a gateway layout to exclude messages from gatewaying, for example.
Filter	Filters are used to define filtered bus monitoring or data logging. You can add filter nodes to system, hardware, controller, and replay nodes.
Monitor	You can monitor the raw and physical data of the bus communication. To monitor the bus communication, a monitor node can be added to the Bus Navigator tree. The monitored bus communication depends on the hierarchy level of the Bus Navigator tree you add a monitor node to. You can define filtered and unfiltered monitoring: ▪ For filtered monitoring, the monitor node must be added to a filter node.

- For unfiltered monitoring, the monitor node can be added to system, hardware, or controller nodes.

Logger

To log the raw data of messages or frames, a logger node can be added to the Bus Navigator tree. The logged raw data depends on the hierarchy level of the Bus Navigator tree you add a logger node to. You can define filtered and unfiltered data logging:

- For filtered data logging, the logger node must be added to a filter node.
- For unfiltered data logging, the logger node can be added to system, hardware, or controller nodes.

Replay

To replay the data logged by a logger node, a replay node can be added to a controller node in the Bus Navigator tree. The replayed data depends on the log file that is loaded for replay. To visualize the replay, you can start monitoring simultaneously.

To replay only selected parts of a logged bus communication, you can add a filter node to the replay node.

Extending the context menu of Bus Navigator tree nodes

ControlDesk lets you extend the context menu of Bus Navigator tree nodes such as communication clusters and network nodes. You can extend the context menu by using the ControlDesk automation interface.

For general information on extending the context menu of elements, refer to Extending the Context Menu of Elements (ControlDesk Customization 🕮).

Related topics

Basics

Extending the Context Menu of Elements (ControlDesk Customization 🕮)

HowTos

How to Display a Bus Configuration with the Bus Navigator..... 34

Bus Navigator Demo

Opening demo projects

For instructions on opening demo projects, refer to Opening a demo project (ControlDesk Introduction and Overview 🕮).

Description of the demo project

The *BusNavigatorDemo* project contains the *Monitoring, Logging, Replay* experiment. The experiment allows you to use the bus communication features

(monitoring, logging, replay, message transmission) of ControlDesk. The demo is based on a simulated bus interface.

Required products and modules Working with this demo requires:

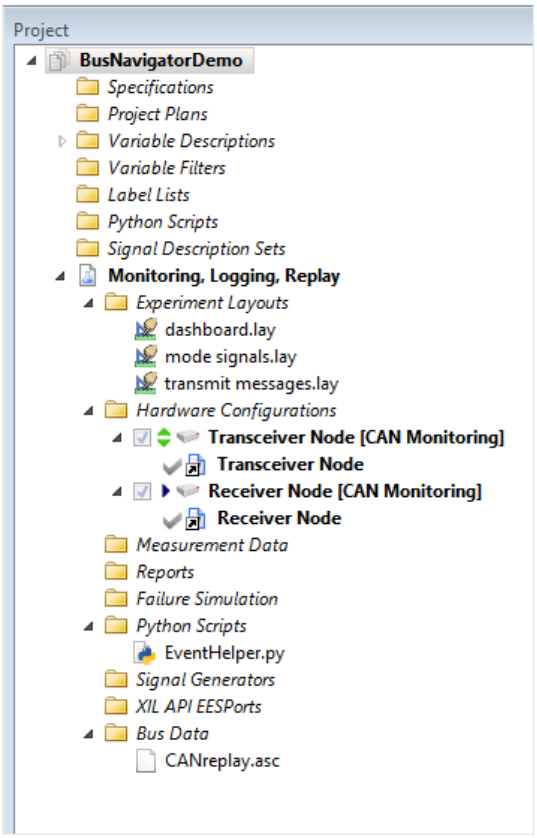
- *ControlDesk*
- *ControlDesk Bus Navigator Module*

Demo setup The *Monitoring, Logging, Replay* experiment contains two CAN Bus Monitoring devices, which represent the CAN controllers of two ECUs interconnected via CAN:

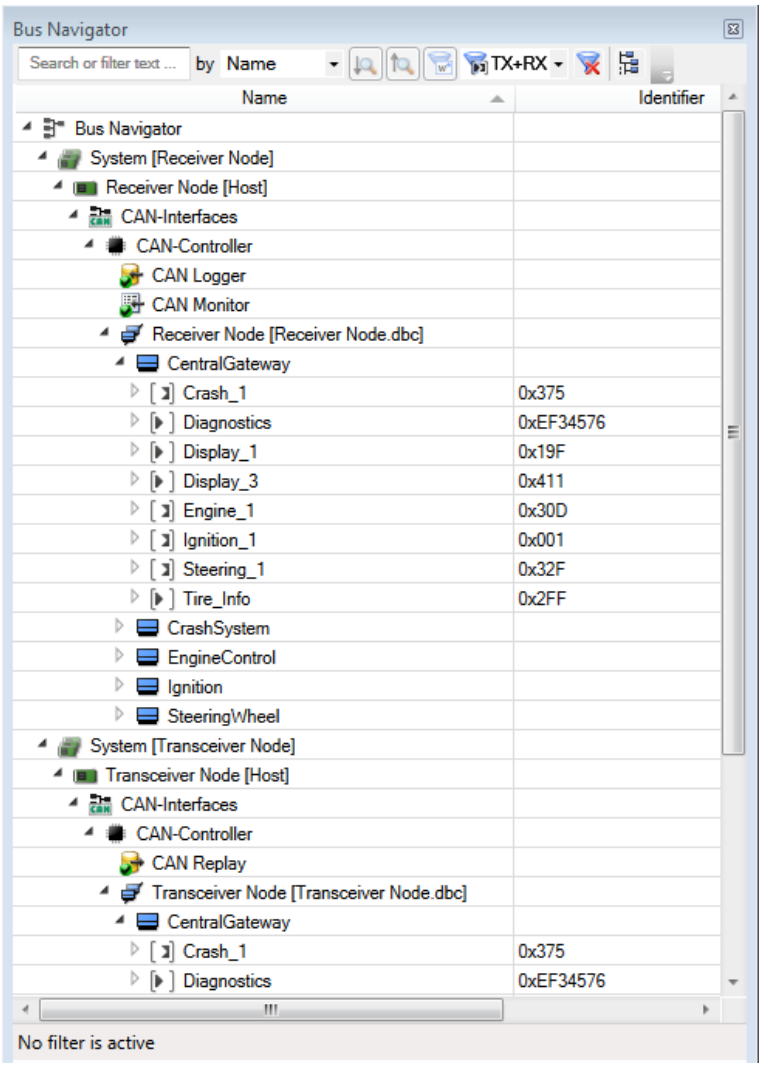
- The *Transceiver Node* device represents the CAN controller of the transmitting ECU.
- The *Receiver Node* device represents the CAN controller of the receiving ECU.

The communication between both devices is based on a virtual CAN bus, so no real CAN bus interface is required.

The illustration below shows the demo setup in the ControlDesk Project controlbar:



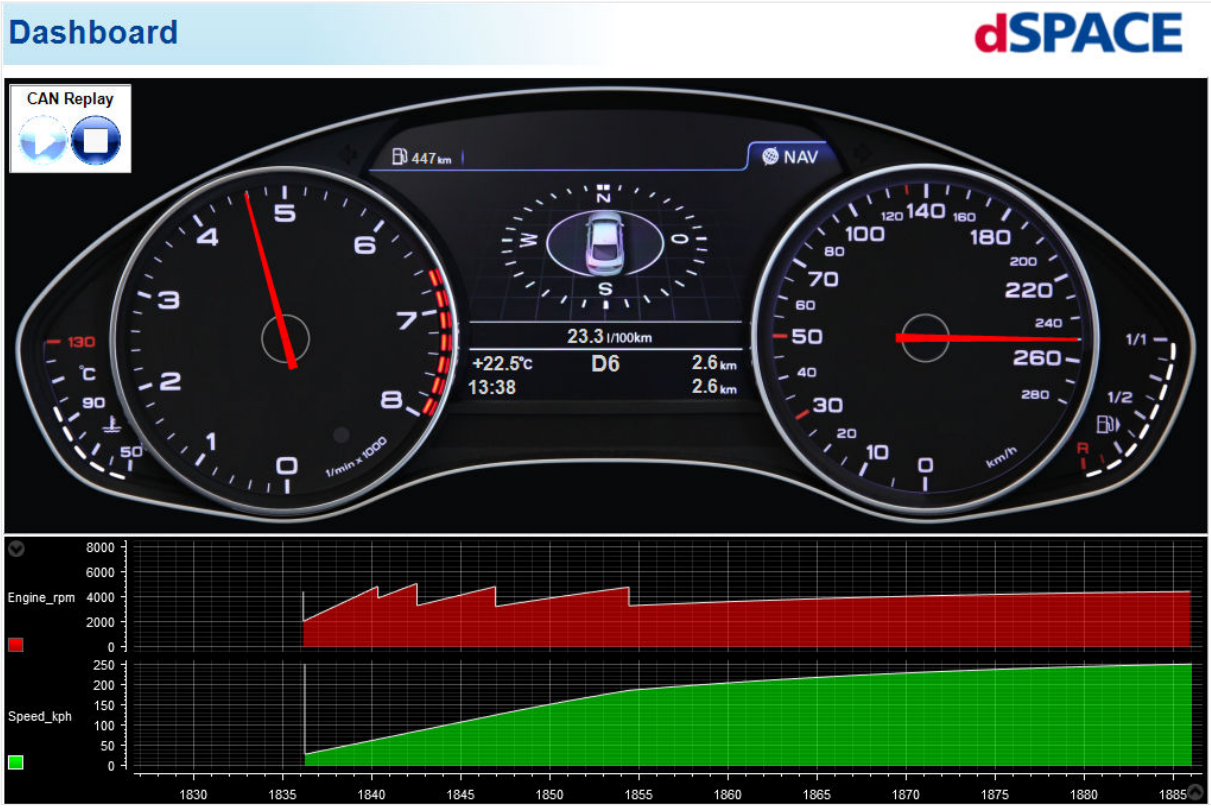
The illustration below shows the related CAN bus configuration in the ControlDesk Bus Navigator:



Demo layouts The *Monitoring, Logging, Replay* experiment contains the following layouts:

- *dashboard* layout

  Contains instruments that visualize signals of the `Display_1`, `Display_3` and `Engine_1` messages.



Note

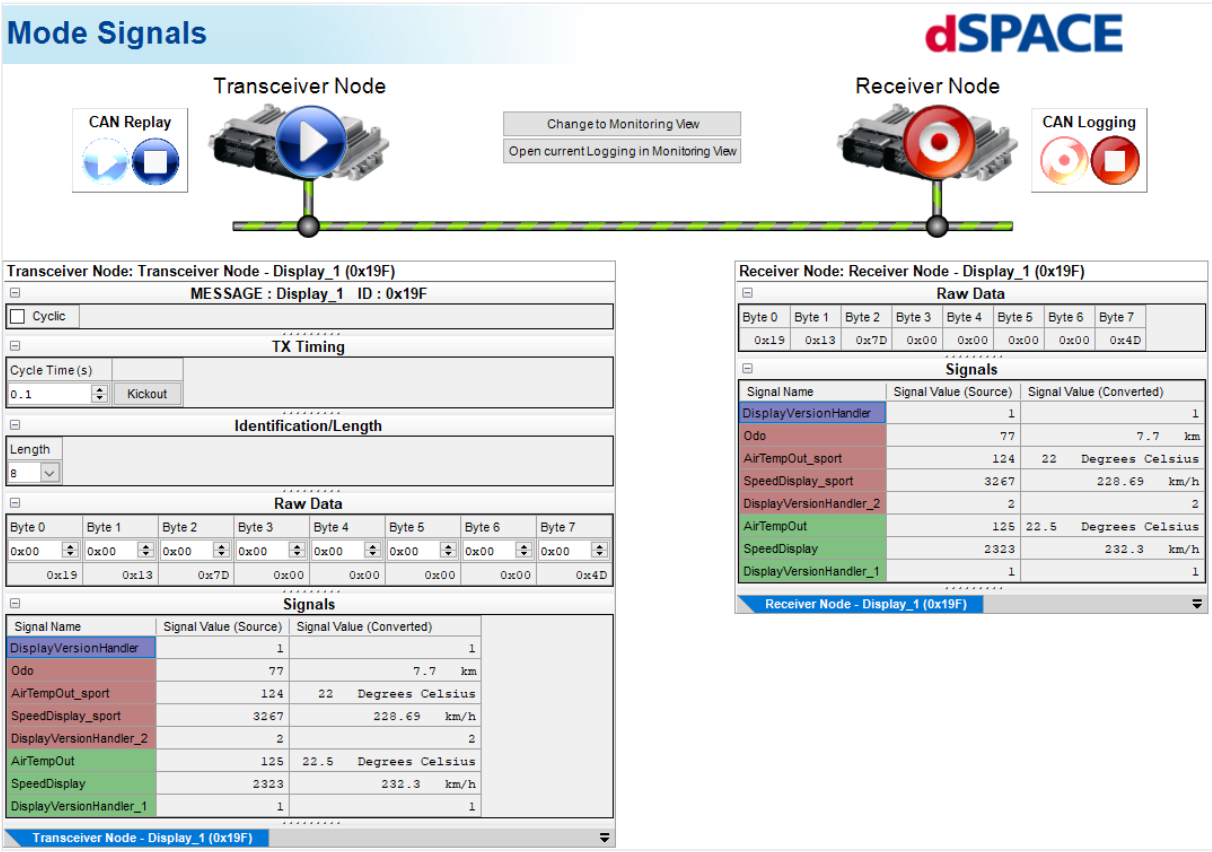
For replaying, monitoring and logging CAN data, online calibration must be started. To visualize CAN data in the Time Plotter on the *dashboard* layout, you also have to start measuring.

Tip

The LayoutAndInstrumentHandling demo shows how to create the dashboard instrument displayed above via tool automation.
Refer to Tool Automation Demos (refer to Tool Automation Demos (ControlDesk Automation 🕮)).

- *mode signals* layout

  Contains instruments that visualize the `Display_1` message.



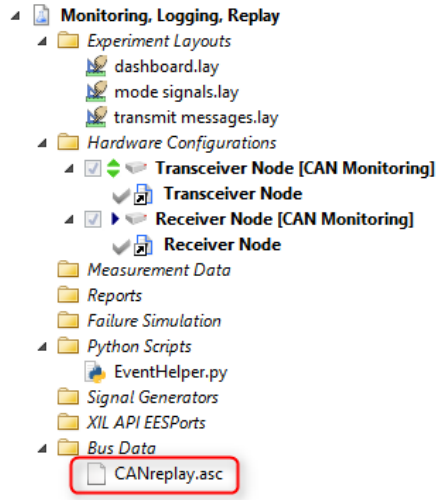
- *transmit messages* layout

  For details on using this layout, see Transmitting and receiving CAN messages on page 32.

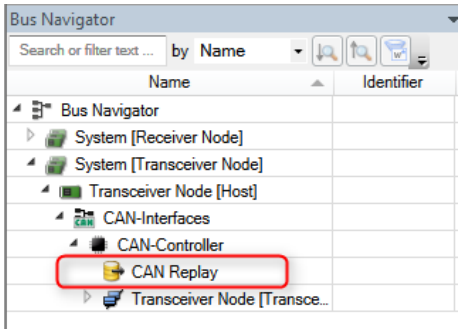
Event handling to activate the monitoring list The *BusNavigatorDemo* project provides an event handling example for activating a monitoring list. For details, refer to Activating a Layout by Using Events (ControlDesk Automation 🕮).

Replaying CAN data

The *Monitoring, Logging, Replay* experiment contains the `CANreplay.asc` replay file:



The replay file is assigned to the CAN Replay node of the Transceiver Node:



To start replaying CAN messages, perform the following steps:

1. On the Home ribbon, click Status Control – Go Online to start online calibration.

   The *Monitoring, Logging, Replay* experiment contains an event handler that performs the following steps when you start online calibration:

   - It activates and starts CAN replay via the Transceiver Node.
   - It starts CAN monitoring and CAN logging via the Receiver Node.

Tip

You can also start replay as follows:

- Click in the *dashboard* or *mode signals* layout.

  OR
- Select Start Replay from the context menu of the CAN Replay node in the Bus Navigator.

To stop replaying CAN messages, perform the following steps:

1. On the **Home** ribbon, click **Status Control – Go Offline** to stop online calibration.

   The experiment contains an event handler that performs the following steps when you stop online calibration:

   - It stops CAN replay.
   - It stops CAN monitoring and CAN logging.

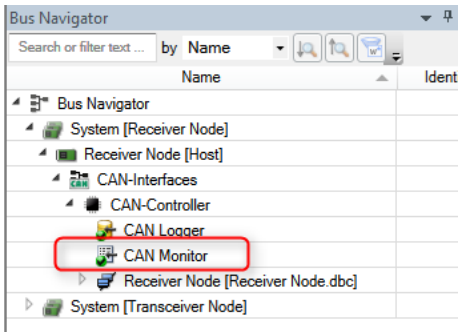
Tip

You can also stop replay as follows:

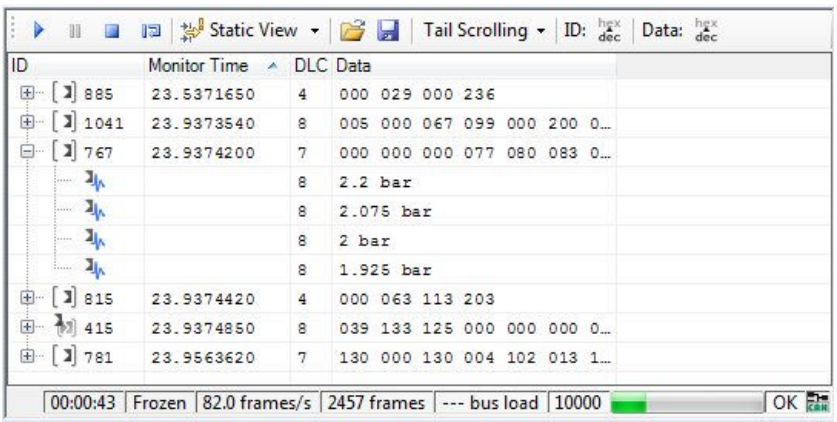
- Click in the *dashboard* or *mode signals* layout.
  OR
- Clear **Start Replay** from the context menu of the **CAN Replay** node in the **Bus Navigator**.

Monitoring CAN data

The **Receiver Node** contains the **CAN Monitor** node:



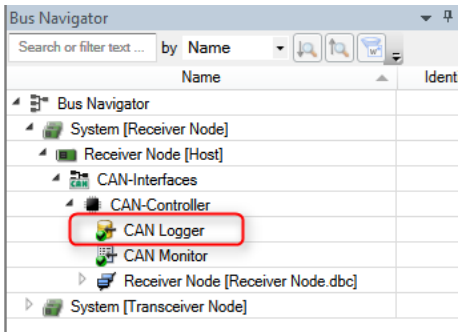
During CAN data replay and when monitoring is started, the **CAN Monitor** monitoring list displays the data of all the CAN messages monitored by the **Receiver Node** (unfiltered monitoring).



Monitoring is started when you start online calibration. You can also start monitoring by clicking in the monitoring list.

Logging CAN data

The Receiver Node contains the CAN Logger node:

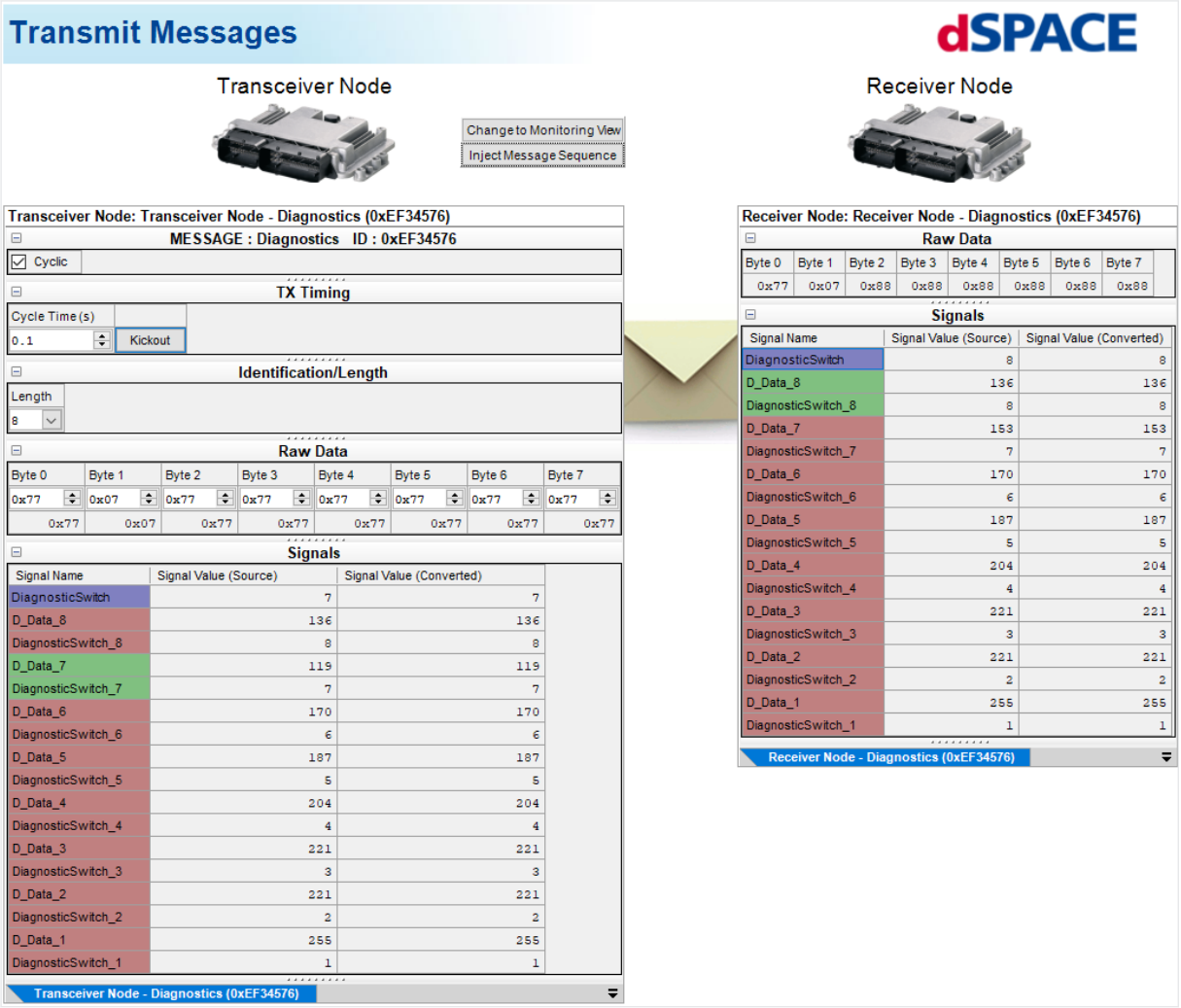


The CAN Logger node logs CAN data to the `CANlogging.asc` file.

Logging is started when you start online calibration. You can also start logging by selecting Start Logger from the context menu of the CAN Logger node in the Bus Navigator.

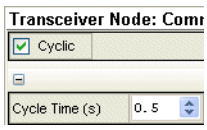
Transmitting and receiving CAN messages

The *transmit messages* layout lets you transmit the Diagnostics CAN message from the *Transceiver Node* device to the *Receiver Node* device.



To transmit and receive CAN messages, perform the following steps:

1. On the Home ribbon, click Status Control – Go Online to start online calibration.
2. Then perform one of the following steps:
   - To transmit the message once, click Kickout in the Bus Instrument (TX Type for CAN).
   - To transmit the message cyclically, specify the desired cycle time and activate Cyclic in the Bus Instrument (TX Type for CAN).



- To transmit a sequence of messages with changing signals, click Inject Message Sequence in the layout.

Related topics

Basics

Monitoring and Logging Bus Communication.....76
Overview of the Bus Navigator.....14
Working with Bus Instruments.....48
Working with the Bus Manager Demo (ConfigurationDesk Bus Manager Implementation Guide 🕮)

Basics on Working with the Bus Navigator

Introduction

You can open instruments via the Bus Navigator to handle the bus communication. The Bus Navigator also lets you monitor, log, and replay bus communication.

Where to go from here

Information in this section

How to Display a Bus Configuration with the Bus Navigator.....34
To display a bus configuration with the Bus Navigator in ControlDesk.

Accessing Bus Communication with a Bus Monitoring Device.....37
You can use a bus monitoring device to access bus communication.
A bus monitoring device lets you monitor, log, and replay bus communication. It also enables you to access single bus messages by means of bus instruments.

Basics on Ethernet PDUs in the Bus Navigator.....41
ControlDesk provides the Ethernet Bus Monitoring Device for accessing Ethernet communication between ECUs.
The Bus Navigator lets you display the Ethernet socket hierarchy for the connected ECUs.
You can create bus instruments to access PDU instances.

Activating Communication Matrices for Blockset-Based Bus Communication.....44
CAN or LIN bus communication modeled with an RTI blockset can contain different communication variants of one controller, but only one can be active at a time. You can switch between the variants during run time.

Basics on Global Time Synchronization.....45
The Bus Navigator lets you display and manipulate CAN PDUs that contain time synchronization information.

How to Display a Bus Configuration with the Bus Navigator

Objective

To display a bus configuration with the Bus Navigator in ControlDesk.

Preconditions

The Bus Navigator tree is only displayed if **Load Bus Navigator configurations automatically** is enabled in the **Bus Navigator** page of the **ControlDesk Options** dialog.

Method

To display a bus configuration with the Bus Navigator

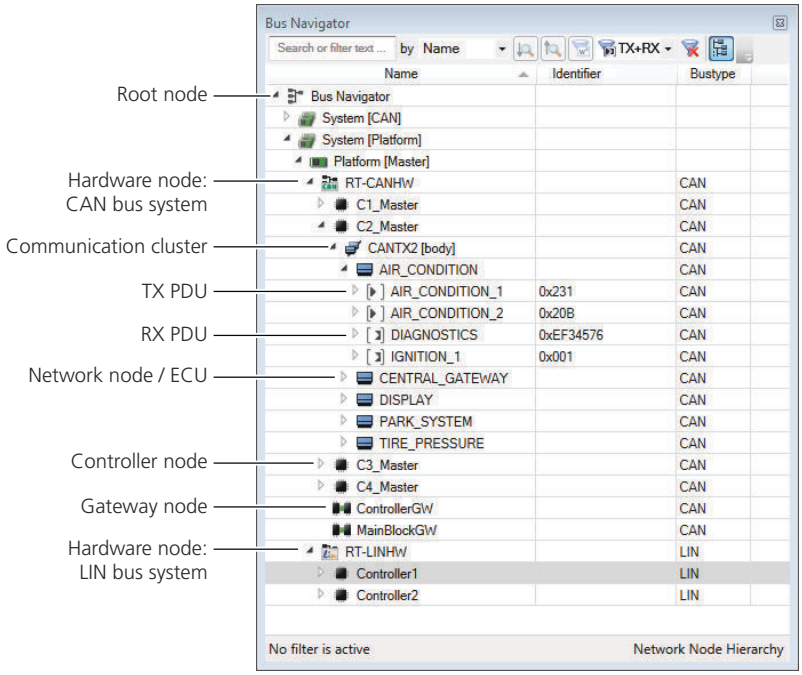
1 Add a platform and an SDF file that references a bus configuration to an experiment in ControlDesk.

– or –

Add a bus monitoring device and a variable description file that describes the bus configuration to an experiment in ControlDesk.

2 On the View ribbon, click Controlbar – Switch Controlbars – Bus Navigator.

ControlDesk opens the Bus Navigator. It displays the bus configuration you added in step 1 in a tree. Refer to the following illustration:



3 Configure the display of the bus configuration as required:

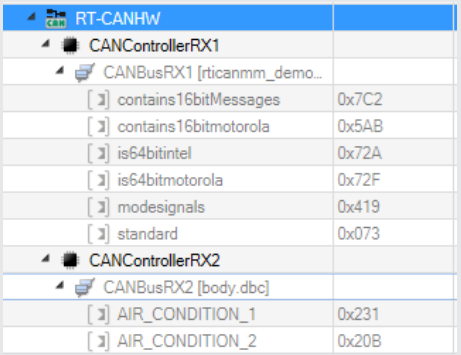
- Right-click the column header and select Customize Columns to change the displayed columns. For details, refer to Customize Columns (Bus Navigator) on page 215.
- Click a column in the column header to sort the tree elements alphabetically in ascending or descending order.

Result

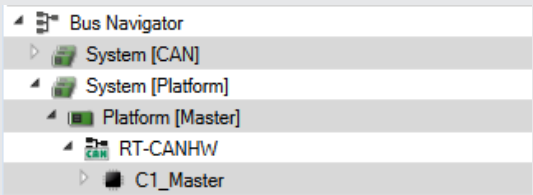
You displayed a bus configuration with the Bus Navigator.

Tip

- If the bus communication was modeled with the RTI CAN MultiMessage Blockset, the LIN MultiMessage Blockset, or the FlexRay Configuration Package, you can unload bus signals from the bus configuration. PDUs with unloaded signals are displayed in gray.



  Unloading bus signals is useful if a bus configuration contains many bus signals which may cause memory exceptions. Refer to Unload All Bus Signals on page 253.
- You can specify alternating row colors for the tree on the **Variables** page of the **ControlDesk Options** dialog. Refer to Variables Page (ControlDesk Variable Management 🕮).



  The setting is also valid for variable lists in other controlbars, such as the **Variables** controlbar.
- You can search for bus elements or filter the tree. Use the toolbar above the items or start an incremental search. Refer to Bus Navigator on page 204.

Next steps

You can use the Bus Navigator to access bus communication via Bus Instruments or bus monitoring:

- For viewing and configuring messages, frames, and PDUs of the bus communication, create Bus Instruments according to your requirements. For exemplary instructions, refer to How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs on page 50.
- For monitoring bus communication, add a monitor node to the Bus Navigator tree. Refer to How to Monitor Bus Communication on page 82.

You can extend the context menu of Bus Navigator tree nodes using the ControlDesk automation interface. Refer to Extending the Context Menu of Elements (ControlDesk Customization 🕮).

Related topics

Basics

Accessing Bus Communication with a Bus Monitoring Device..... 37
Basics on Bus Configurations..... 17
Bus Configuration Structure in the Bus Navigator Tree..... 19
Extending the Context Menu of Elements (ControlDesk Customization 🕮)

HowTos

How to Add a Platform/Device to an Experiment (ControlDesk Platform Management 🕮)

References

Bus Navigator..... 204

Accessing Bus Communication with a Bus Monitoring Device

Introduction

You can use a bus monitoring device to access bus communication.

A bus monitoring device lets you monitor, log, and replay bus communication. It also enables you to access single bus messages by means of bus instruments.

Use cases

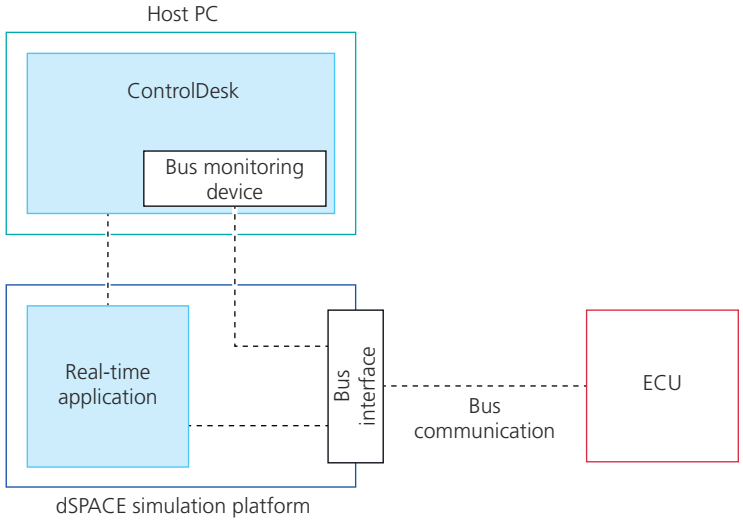
A bus monitoring device lets you access the bus communication of an ECU.

The ECU communicates with the simulation platform using the corresponding bus channels.

There are three ways to access bus communication with a bus monitoring device:

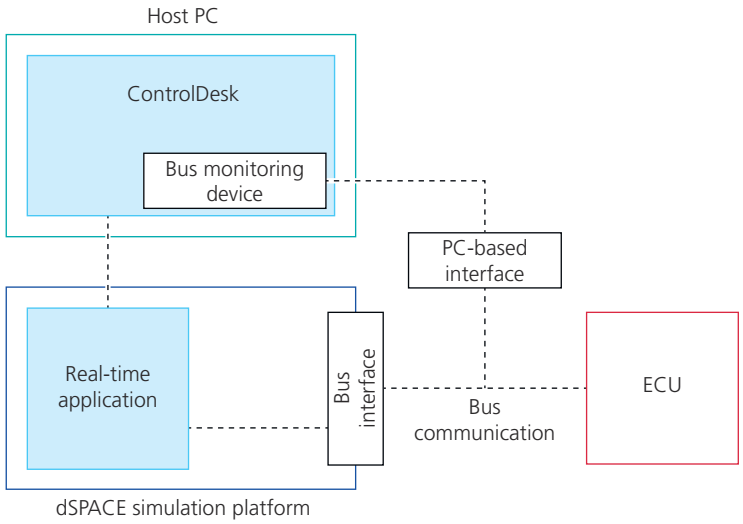
- Using a bus monitoring device in connection with a bus interface of a dSPACE simulation platform, i.e., SCALEXIO or a MicroAutoBox III.

Refer to the following illustration.



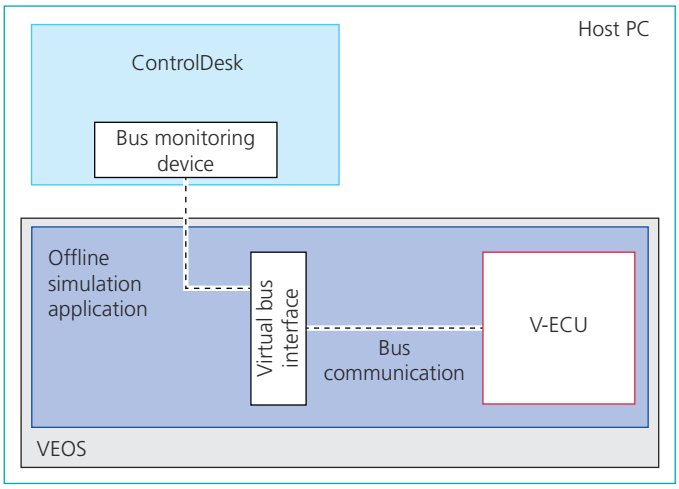
- Using a bus monitoring device in connection with a PC-based interface, such as a DCI-CAN/LIN1 or a DCI-CAN2.

  Refer to the following illustration.



- Using a bus monitoring device in connection with a virtual bus interface of an offline simulation application (OSA) in VEOS.

Refer to the following illustration.



Application-independent access to the bus communication

When you run a real-time application on a dSPACE simulation system, you can use a bus monitoring device to access the bus communication *independently* of the real-time application. The device interprets the bus communication via a variable description file that you can load to it. Therefore, you can make changes to the communication matrix without having to rebuild the real-time application and you can also observe bus messages that have not been configured with the Bus Manager or a dSPACE blockset.

For information on supported variable description file types, refer to the following pages:

- CAN Bus Monitoring Device (ControlDesk Platform Management 🕮)
- Ethernet Bus Monitoring Device (ControlDesk Platform Management 🕮)
- LIN Bus Monitoring Device (ControlDesk Platform Management 🕮)

Accessing single bus messages using bus instruments

You can access *single* bus messages using bus instruments.

For instructions on how to create bus instruments to handle RX and TX messages, refer to How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs on page 50.

For instructions on how to configure and transmit a CAN TX message, refer to How to Configure a CAN TX Message and Transmit It on the Bus on page 74.

Replaying logged bus communication

You can use the CAN and Ethernet Bus Monitoring devices to replay logged bus communication. Refer to Basics on Replaying CAN and Ethernet Bus Communication on page 99.

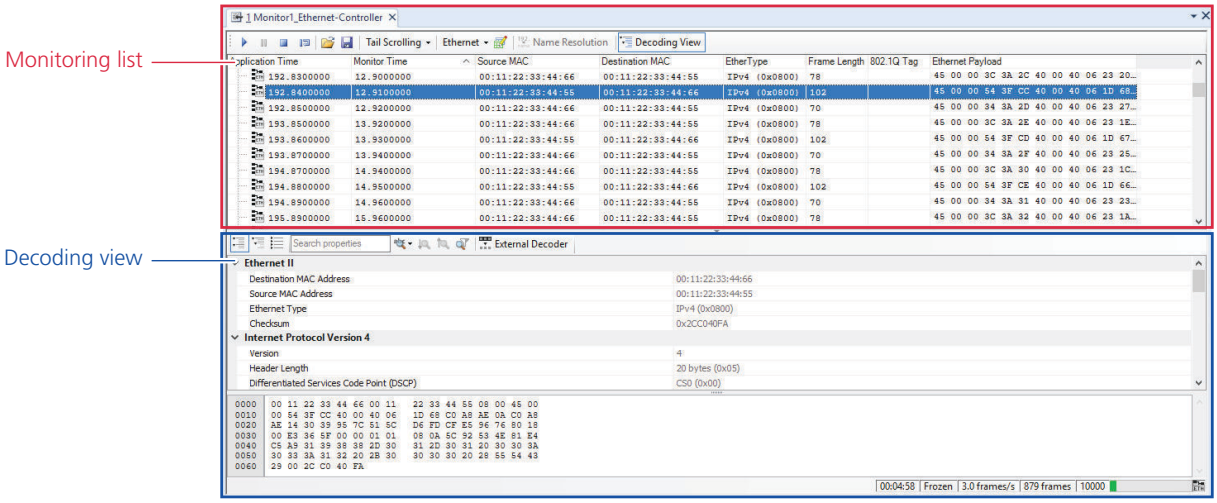
Sending CAN messages

You can also use the CAN Bus Monitoring device to send CAN messages that are imported from a variable description file.

To enable this feature, open the ControlDesk **Options** dialog and go to **Variables – Import Options – Bus Import Options**. Under **Variable handling**, select **Generate raw byte variables for each CAN message**. For more information on this option, refer to Variables Page (ControlDesk Variable Management 🕮).

Ethernet decoding view

For Ethernet monitoring, the monitoring list provides the decoding view, which displays detailed information on a selected Ethernet frame. Refer to the following illustration.



For more information, refer to Decoding View on page 217.

Related topics

Basics

Basics on Ethernet PDUs in the Bus Navigator.....41
Basics on Replaying CAN and Ethernet Bus Communication.....99
Dependencies Regarding Monitoring, Logging, and Filtering Bus Communication.....80

HowTos

How to Configure a CAN Bus Monitoring Device (ControlDesk Platform Management 🕮)
How to Configure a LIN Bus Monitoring Device (ControlDesk Platform Management 🕮)
How to Configure an Ethernet Bus Monitoring Device (ControlDesk Platform Management 🕮)
How to Monitor Bus Communication.....82
How to Replay Logged CAN or Ethernet Bus Communication.....101

References

Decoding View.....217
Monitoring List.....240

Options Command (ControlDesk User Interface Handling 🕮)

Basics on Ethernet PDUs in the Bus Navigator

Introduction

ControlDesk provides the Ethernet Bus Monitoring Device for accessing Ethernet communication between ECUs.

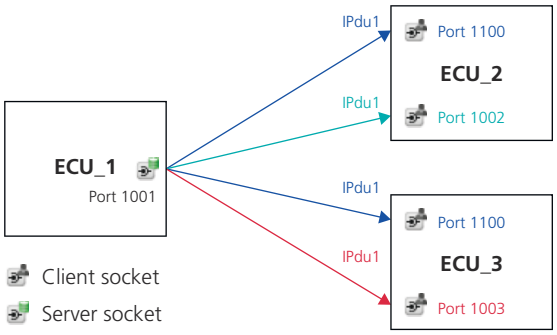
The Bus Navigator lets you display the Ethernet socket hierarchy for the connected ECUs.

To display information on single PDUs, you can create bus instruments.

Use case

In the example use case shown in the following illustration, ECU_1 sends one instance of IPdu1 from port 1001 to ECU_2 and ECU_3 via unicast. The PDU is received on port 1002 and port 1003, respectively.

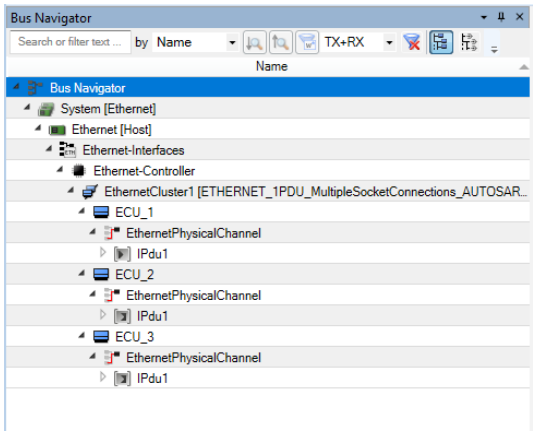
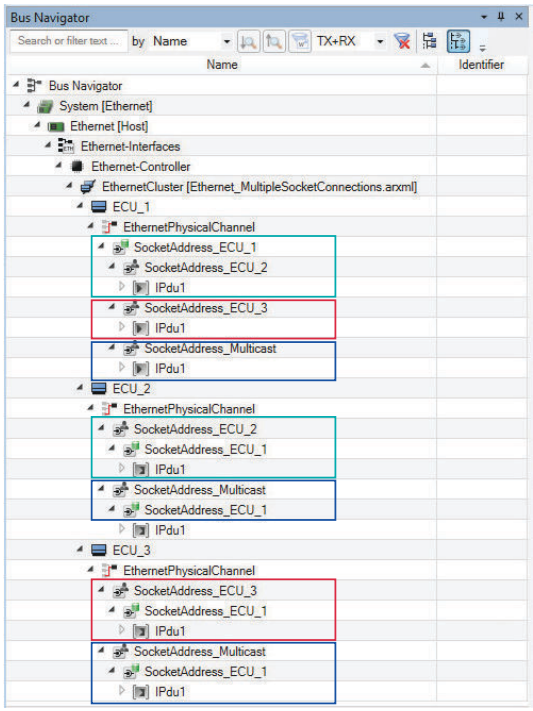
ECU_1 also sends an instance of IPdu1 to the other ECUs via multicast, which is assigned to port 1100.



Ethernet socket hierarchy

The Bus Navigator tree provides two view configurations for displaying Ethernet communication between ECUs. You can select the view configuration by using the Show Socket Hierarchy (🌳) button.

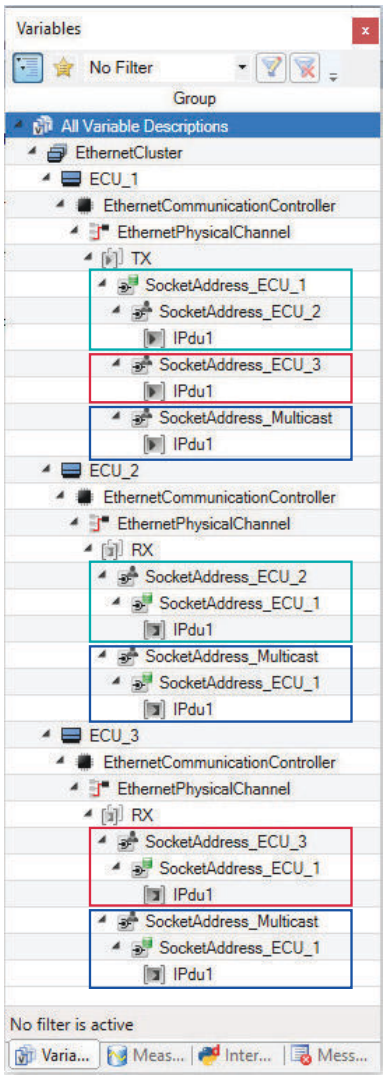
The following table compares the two view configurations:

	Show Socket Hierarchy Disabled	**Show Socket Hierarchy Enabled**
Bus Navigator tree display		
PDU instances in the tree	PDU instances cannot be distinguished. IPdu1 is listed only once for each ECU.	Each instance of the PDU is displayed in the context of the client and server sockets. The coloring of the instances of IPdu1 corresponds to the schematic under Use case on page 41.
Bus instrument creation[1)]	ControlDesk creates a bus instrument with one tab for each instance of the selected PDU. The tab header displays information on the sender and receiver address.	ControlDesk creates a bus instrument for the selected PDU instance.
Layout creation for signals of PDUs	You cannot use drag & drop to create a layout for a signal of a PDU.	You can drag a signal of a PDU to the working area to create a layout with an instrument.

[1)] You can create the Bus Instrument (RX Type for Ethernet) for each PDU that is displayed in the Bus Navigator tree.

Ethernet PDUs in the Variables controlbar

The following illustration shows the corresponding display in the Variables controlbar.



Tool automation

You can generate the Bus Instrument (RX Type for Ethernet) via tool automation. However, you cannot access Ethernet sockets via tool automation.

For more information on generating bus instruments via tool automation, refer to Automating the Bus Navigator on page 256.

Related topics

Basics

Bus Configuration Structure in the Bus Navigator Tree..... 19

Dependencies Regarding Monitoring, Logging, and Filtering Bus Communication..... 80

HowTos

How to Monitor Bus Communication..... 82

References

Bus Instrument (RX Type for Ethernet)..... 178

Activating Communication Matrices for Blockset-Based Bus Communication

Introduction

CAN or LIN bus communication modeled with an RTI blockset can contain different communication variants of one controller. You can switch between the variants during run time. This means that a CAN/LIN controller can have several communication matrices as child nodes in the Bus Navigator tree, but only one can be active at a time. You can activate a communication matrix via the Bus Navigator or the real-time model.

Activating a communication matrix via the real-time model

To activate or deactivate a communication matrix, the real-time model contains the Variation variable in the RTICANMM/RTILINMM ControllerSetup block (refer to RTICANMM ControllerSetup (RTI CAN MultiMessage Blockset Reference 🕮)/RTILINMM ControllerSetup (RTI LIN MultiMessage Blockset Reference 🕮)). The variable is used to switch between the RTICANMM/RTILINMM MainBlocks. You can find it below BusSystems/<bus type>/ControllerSetup_Blocks/<controller_name> in the Variables controlbar of ControlDesk.

To control the variation from within the real-time model, the Variation inport of the RTICANMM/RTILINMM ControllerSetup block must be activated. This is the case if the variable BusSystems/<bus type>/ControllerSetup_Blocks/<controller name>/Variation_Switch is `0`.

Activating a communication matrix via the Bus Navigator

You can change the active variant of a CAN or LIN controller via the Bus Navigator. To activate a communication matrix, select Activate Database from the context menu of the corresponding communication matrix node in the Bus Navigator tree.

Switching the communication matrix via the Bus Navigator deactivates the Variation inport of the RTICANMM/RTILINMM ControllerSetup block in the Simulink model.

Tip

If you want to control the CAN/LIN controller variation from the Simulink model again, you must reactivate the Variation inport of the RTICANMM/RTILINMM ControllerSetup block by setting the Variation_Switch variable to `0`.

Related topics

References

Activate (Database)..... 185

Basics on Global Time Synchronization

Introduction

The concept of global time synchronization was introduced and standardized by AUTOSAR as a means of providing and distributing synchronized times across all ECUs in a vehicle.

The Bus Navigator lets you display and manipulate CAN PDUs that contain time synchronization information.

Basic concepts of global time synchronization

Time bases A time base is a unique time entity which is characterized by the progression of time, ownership of the time base, and reference to the physical world.

There are two types of time bases:

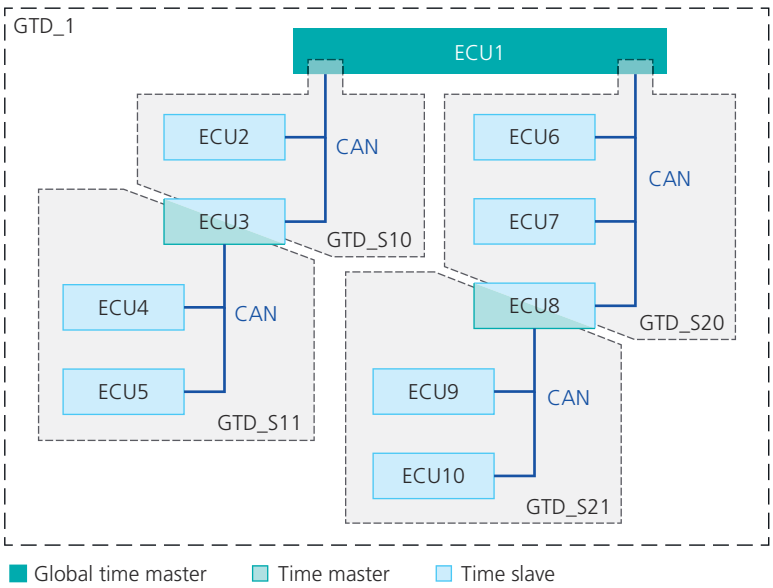
- Synchronized time bases: A synchronized time base is a time base which is synchronized with other time bases at different processing entities. These synchronized time bases constitute a global time.
- Offset time bases: An offset time base is a time base which depends on a particular synchronized time base relative to which it holds an offset value.

Global time domains All components (e.g., nodes and communication systems) that are linked to a synchronized time base constitute a global time domain. A global time domain can contain several time subdomains, i.e., components that belong to the same time base but different communication buses.

Time master, slave, and gateway Global time synchronization is based on a master-slave principle. Each global time domain has a time master which acts as the source for the time base, distributing the time base cyclically via the bus systems to a set of time slaves. Each time slave has its own time base instance, i.e., local instance of the time.

A time gateway receives the time base as a slave and distributes it to a set of time slaves as a time master. This set of time slaves and master then constitutes a time subdomain.

The following schematic illustrates the distribution of a global time in a bus network with four communication clusters and five global time domains. *GTD_S10* and *GTD_S20* are subdomains of *GTD_1*. *GTD_S11* and *GTD_S21* are subdomains of *GTD_S10* and *GTD_S20*, respectively, where ECU3 and ECU8 act as time gateways. All global time domains use the same domain identifier.



Time synchronization messages Exchanging time via CAN requires two messages: a time synchronization (SYNC) message and a follow-up (FUP) message. Whenever a time slave receives a new time synchronization message from the time master, the time slave extrapolates the time until the next synchronization with the time master. The FUP message completes each synchronization process. SYNC and FUP messages are transmitted using the same CAN identifier.

Time synchronization messages can be CRC-secured or unsecured. The following illustrations show both cases together with the corresponding FUP messages. The CRC-secured case is at top. In this case, only one user byte is available, while the unsecured case provides three user bytes (two as part of the SYNC message, one as part of the FUP message).

SYNC:

Message type	CRC	Time domain	E2E sequence counter	User byte 0	Seconds
8 bits	*8 bits*	*4 bits*	*4 bits*	*8 bits*	*32 bits*
Byte 0	Byte 1	Byte 2		Byte 3	Byte 4 ... Byte 7

MSB LSB

FUP:

Message type	CRC	Time domain	E2E sequence counter	Reserved	SGW (1 bit)	Overflow of seconds (2 bits)	Nanoseconds
8 bits	*8 bits*	*4 bits*	*4 bits*	*5 bits*			*32 bits*
Byte 0	Byte 1	Byte 2		Byte 3			Byte 4 ... Byte 7

MSB LSB

SYNC:

Message type	User byte 1	Time domain	E2E sequence counter	User byte 0	Seconds
8 bits	*8 bits*	*4 bits*	*4 bits*	*8 bits*	*32 bits*
Byte 0	Byte 1	Byte 2		Byte 3	Byte 4 ... Byte 7

MSB LSB

FUP:

Message type	User byte 2	Time domain	E2E sequence counter	Reserved	SGW (1 bit)	Overflow of seconds (2 bits)	Nanoseconds
8 bits	*8 bits*	*4 bits*	*4 bits*	*5 bits*			*32 bits*
Byte 0	Byte 1	Byte 2		Byte 3			Byte 4 ... Byte 7

MSB LSB

The information in the SYNC and FUP messages can be used for validation, i.e., to check whether the time synchronization has been executed in accordance with the transport protocol.

Configuration For more information on configuring time synchronization, refer to the following documentation:

- Bus Manager:
  - Bus Manager in ConfigurationDesk: Bus Configuration Features Available for Global Time Domains (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Bus Configuration Features Available for Global Time Domains (Bus Manager (Stand-Alone) Implementation Guide 🕮)
- RTI CAN MultiMessage Blockset:
  - Block Description (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)

Related topics

References

Bus Instrument (RX Type for CAN)..... 114
Bus Instrument (TX Type for CAN)..... 121
Bus Instrument (TX Type for CAN)..... 147

Working with Bus Instruments

Introduction

The Bus Navigator provides Bus Instruments to view and/or manipulate the bus communication of a bus configuration.

Where to go from here

Information in this section

General Handling of Bus Instruments.....48

Instruments for Bus Communication Modeled with the Bus Manager.....57

Instruments for Bus Communication Modeled with a Blockset.....62

Instruments Related to Bus Monitoring Devices.....73

General Handling of Bus Instruments

Where to go from here

Information in this section

How to Configure Layout Names.....48
To configure the naming of layouts that are generated automatically when you create a Bus Instrument.

How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs.....50
To create Bus Instruments for handling single RX and TX messages/frames/PDUs.

How to Configure a Bus Instrument.....55
To configure a Bus Instrument.

How to Configure Layout Names

Objective

To configure the naming of layouts that are generated automatically when you create a Bus Instrument.

Basics

The Bus Navigator uses a default macro to define unique names for the layouts that are generated automatically when you create a Bus Instrument via a context menu of the Bus Navigator tree.

You can configure the naming of the layouts by specifying a macro containing typed text and/or predefined macros.

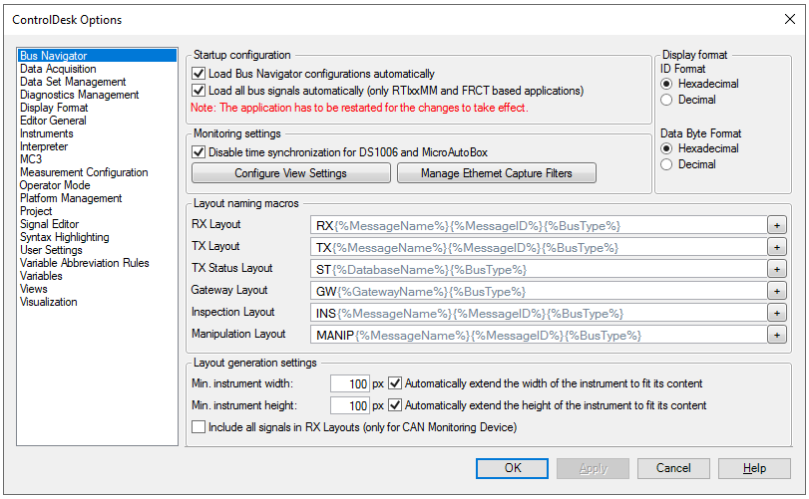
Method

To configure layout names

1 On ControlDesk's File ribbon, click Options.

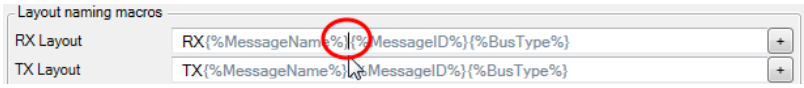
ControlDesk opens the ControlDesk Options dialog.

2 In the ControlDesk Options dialog, select the Bus Navigator page.



The Bus Navigator page displays the predefined macros for the layout names of the Bus Navigator instruments.

3 Place the cursor in the edit field of the layout whose name you want to configure.



Note

Make sure that you place the cursor outside of the braces of a predefined macro.

4 To configure the layout name, type text in the edit field and/or click the Browse button to select a predefined macro from the list.

5 Click OK to confirm your settings.

Result

When you generate a new layout via the Bus Navigator tree, it is named according to the macro you specified.

Note

Layouts that are currently open or already saved in the Experiment layouts folder in the Project controlbar are not renamed.

Related topics

References

Bus Navigator Page..... 207

How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs

Objective

To create the following Bus Instruments for handling single RX and TX messages/frames/PDUs:

- A Bus Instrument (RX type) for an RX or TX/RX message/frame/PDU
- A Bus Instrument (TX type) for a TX or TX/RX message/frame/PDU

Restriction

For a bus configuration related to an Ethernet Bus Monitoring device or a LIN Bus Monitoring device, you can only generate a Bus Instrument (RX type).

Precondition

The experiment must contain a bus configuration related to one or more of the following devices:

- CAN controller
- FlexRay controller
- LIN controller
- Ethernet controller of an Ethernet Bus Monitoring device

The bus configuration must include at least one message, frame, or PDU that has been configured for simulation purposes.

Method

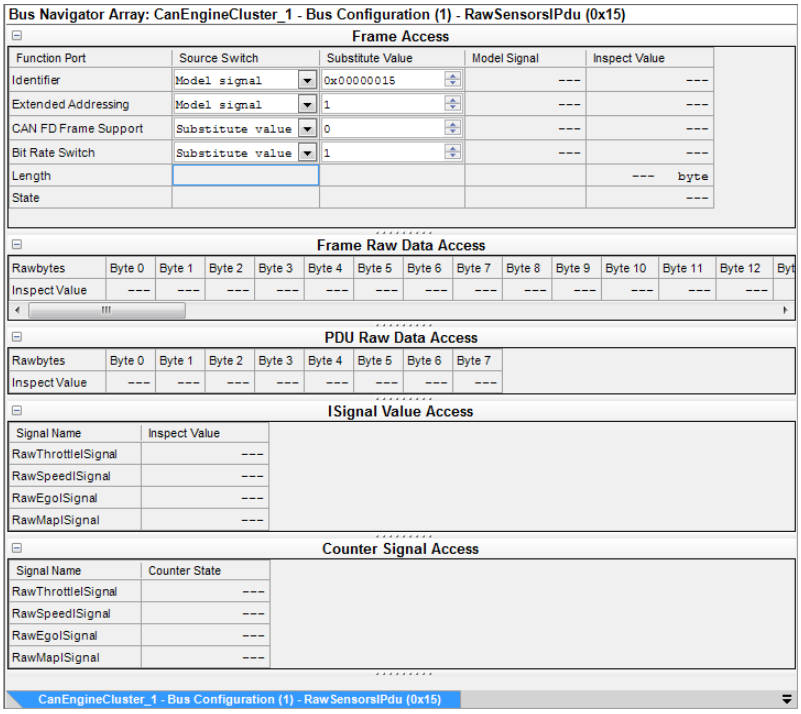
To create Bus Instruments to handle RX and TX messages/frames/PDUs

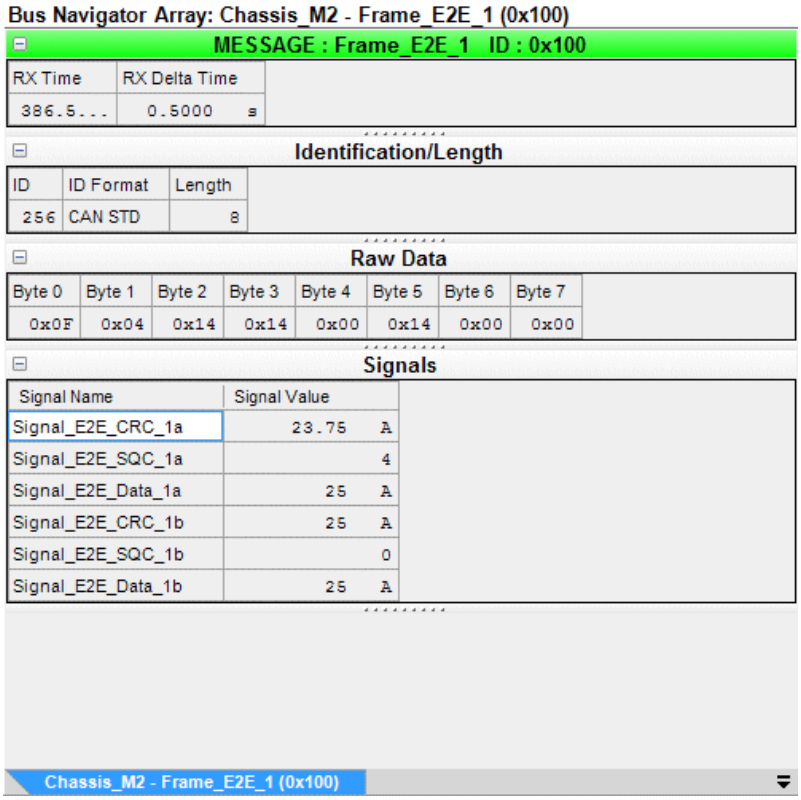
1 From the Bus Navigator tree, drag an RX message/frame/PDU node onto an empty area of an existing layout.

– Or –

From the context menu of an RX or TX/RX message/frame/PDU node, select **Generate RX Layout** to create a new layout containing a Bus Instrument (RX type).

ControlDesk opens a Bus Instrument (RX type) for the selected message/frame/PDU, see the following examples. The first example shows a Bus Instrument (RX type) for bus communication modeled with the Bus Manager. The second example shows a Bus Instrument (RX type) for bus communication modeled with a blockset.





Tip

You can change the default size of a newly created Bus Instrument in the ControlDesk Options dialog. Refer to Bus Navigator Page on page 207.

2 From the Bus Navigator tree, drag a TX or TX/RX message/frame/PDU node onto an empty area of an existing layout.

– Or –

From the context menu of an RX or TX/RX message/frame/PDU node, select Generate TX Layout to create a new layout containing a Bus Instrument (TX type) on it.

ControlDesk opens a Bus Instrument (TX type) for the selected message/frame/PDU, see the following examples. The first example shows a Bus Instrument (TX type) for bus communication modeled with the Bus Manager. The second example shows a Bus Instrument (TX type) for bus communication modeled with a blockset.

Bus Navigator Array: CanEngineCluster_1 - Bus Configuration (1) - FuelRateEnginelPdu (0x14)

PDU Enable Access

Source Switch	Substitute Value	Model Signal	Inspect Value
Model signal	☑ Enable	No conversion	No conversion

PDU Cyclic Timing Control

Function Port	Source Switch	Substitute Value	Model Signal	Inspect Value
Period	Model signal	0	---	---
Offset	Model signal	0	---	---

PDU Trigger Access

Source Switch	Substitute Value	Model Signal	Inspect Value
Model signal	☐ Trigger	No conversion	No conversion

PDU Raw Data Access

Source Switch		
Model signal		
Rawbytes	Byte 0	Byte 1
Substitute Value	0x00	0x00
Model Signal	---	---
Inspect Value	---	---

ISignal Value Access

Signal Name	Source Switch	Substitute Value	Model Signal	Inspect Value
FuelRateEnginelSignal	Model signal	0	---	---

CanEngineCluster_1 - Bus Configuration (1) - FuelRateEnginelPdu (0x14)

Bus Navigator Array: Chassis_M1 - IGNITION_1 (0x1)

MESSAGE : IGNITION_1 ID : 0x1

☑ GlobalEnable ☑ GlobalEnableTX ☑ ECUEnable ☑ Enable ☑ Cyclic

CRC

CRC	CRC Alg.	CRC Dyn. Alg.	CRC Dyn.Cnt.
☑	CRConByte7_f...	CRConByte7	0

TX Timing

Cycle Time (s)	Delay Time	Time Out	
1	0.000 s	0	Kickout

RX Times

RX Time	RX Delta Time
4841....	1.0000 s

Identification/Length

Message ID	XTD ID Format	Length	Countdown	Dyn. Length
1	☐	8	0	8

Raw Data

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x07	0x00	0x00	0x00	0x00	0x00	0x70	0x78

Signals

Signal Name	Source	Constant	Signal Value
IgnitionKey		7	7
Ignition_on_no_Start		☐	0
EngineStart		☐	0
MC_Ignition_1	Counter	0	7
CRC_Ignition_1		255	120

Chassis_M1 - IGNITION_1 (0x1)

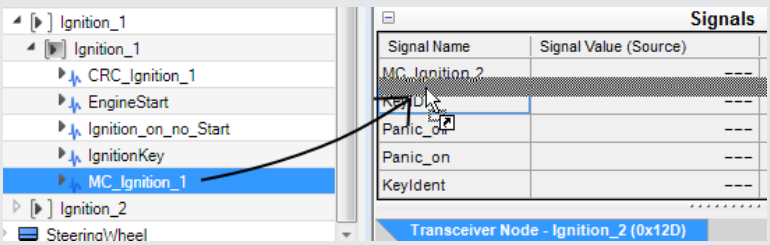
Result

- The Bus Instrument (RX type) and the Bus Instrument (TX type) opened.
- The instruments are tailor-made and the appearance of the instruments depends on the bus configuration(s) available in the loaded variable configuration file. The instruments display information on messages/frames/PDUs and their signals.
- If you modeled the bus communication using the Bus Manager and the bus configuration file contains various bus configurations for a message or frame, ControlDesk creates instrument tabs for each bus configuration.
- Signals contained in the message/frame/PDU are displayed in a list of signal names and signal values. If a signal is not connected to its corresponding variable in the system model, the signal value entry is crossed out.
- You can open several RX Bus Instruments and TX Bus Instruments for the same message/frame/PDU at the same time.

Tip

There are several ways to create or configure an instrument by dragging items from the Bus Navigator tree to the layout (multiselection is possible):

- Dragging a message/frame/PDU to an empty area of an existing layout adds a new Bus Instrument to the layout (as shown above).
- Dragging a message/frame/PDU to an existing Bus Instrument adds a new tab to the instrument.
- Dragging a signal onto an existing Bus Instrument (or another appropriate instrument, such as a Variable Array) adds the signal to the instrument. The signal and the Bus Instrument must have the same bus type.



- Dragging a signal onto an empty area of the layout opens the instrument list and lets you select an instrument (for example a Variable Array) to be created for the signal. Multiselection of signals is possible. The signal is always connected to the RX variable.

If you right-click when dragging the element, a list opens for you to select the action to perform.

To delete signals from an instrument, select them and press **`Delete`**. Multiselection is possible.

For details on the Bus Instrument (RX type) and the Bus Instrument (TX type), refer to Bus Instruments on page 104.

Next steps

- To display an active bus communication in the instruments, you must start online calibration.

- A Bus Instrument (TX type) can be used to configure and transmit messages/frames/PDUs. Refer to
  - How to Configure a TX Message/Frame/PDU and Transmit It on the Bus on page 59 (bus communication modeled with the Bus Manager)
  - How to Configure a TX Message/Frame/PDU and Transmit It on the Bus on page 64 (bus communication modeled with a blockset)
  - How to Configure a CAN TX Message and Transmit It on the Bus on page 74 (CAN Bus Monitoring device)

Related topics

References

Generate RX Layout..... 232
Generate TX Layout..... 234

How to Configure a Bus Instrument

Objective

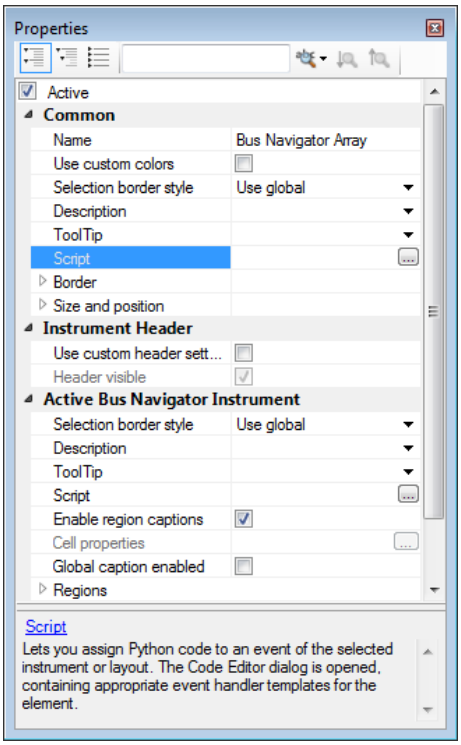
To configure a Bus Instrument.

Method

To configure a Bus Instrument

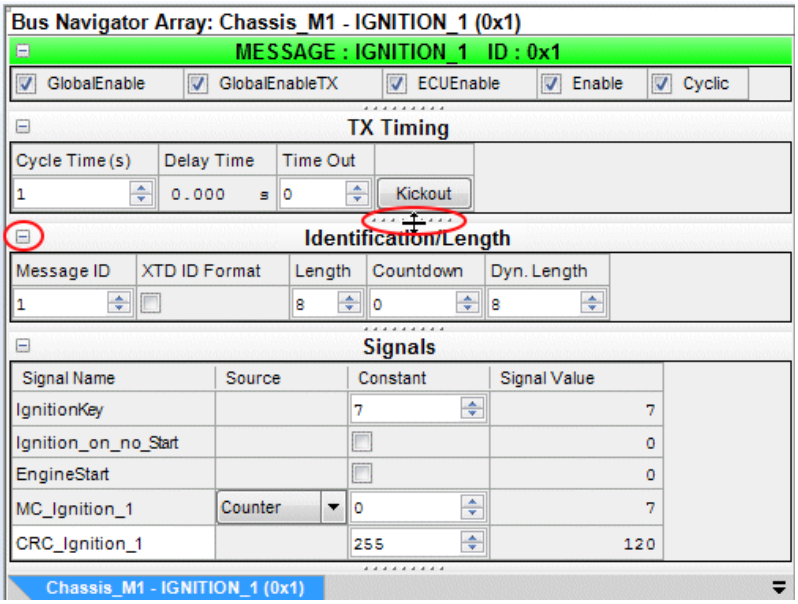
1 From the context menu of the Bus Instrument, select Instrument Properties.

ControlDesk activates the **Properties** controlbar and displays the properties of the selected Bus Instrument.



2 Select properties and configure them.

3 In the Bus Instrument, use the splitter bars and buttons to expand or collapse regions as shown in the following illustration.



Result

You configured the Bus Instrument.

Tip

- Via the buttons below the title bar of the **Properties** controlbar, you can:
  - Expand or collapse categories and properties
  - Search and filter the controlbar's content
- On the **Layouting** ribbon, click **Instruments – Assign Properties** to assign the property settings of shared properties from the currently selected instrument to the next one you select.

Related topics

References

Bus Instruments..... 104

Instruments for Bus Communication Modeled with the Bus Manager

Where to go from here

Information in this section

Overview of Available Bus Instrument Types..... 58
The Bus Navigator provides different Bus Instrument types to handle CAN and LIN bus communication modeled with the Bus Manager.

How to Configure a TX Message/Frame/PDU and Transmit It on the Bus..... 59
To configure a TX message/frame/PDU and transmit it on the bus via the Bus Instrument (TX type).

How to Specify Global Transmission Settings for a CAN or LIN Controller..... 61
To specify basic transmission settings globally for all the TX PDUs of a CAN or LIN communication matrix via the Bus Instrument (TX status type).

Overview of Available Bus Instrument Types

Introduction

The Bus Navigator provides different Bus Instrument types to handle CAN and LIN bus communication modeled with the Bus Manager.

Instrument types

The following table provides an overview of all the Bus Instrument types that are available for bus communication modeled with the Bus Manager.

Bus Instrument Type	Function	Bus Types
Global Layout	To enable and disable the bus communication of a bus configuration fully or partly in one step.	CAN, LIN (refer to Bus Instrument (Global Layout Type) on page 105)
Inspection	To display information on PDUs that have been configured for inspection purposes with the Bus Manager.	CAN (refer to Bus Instrument (Inspection Type for CAN) on page 107)
		LIN (refer to Bus Instrument (Inspection Type for LIN) on page 126)
Manipulation	To display and change the settings of PDUs that have been configured for manipulation purposes with the Bus Manager.	CAN (refer to Bus Instrument (Manipulation Type for CAN) on page 110)
		LIN (refer to Bus Instrument (Manipulation Type for LIN) on page 128)
RX	To display information on RX messages/frames/PDUs.	CAN (refer to Bus Instrument (RX Type for CAN) on page 114)
		LIN (refer to Bus Instrument (RX Type for LIN) on page 131)
TX	To display, configure and transmit TX messages/frames/PDUs.	CAN (refer to Bus Instrument (TX Type for CAN) on page 121)
		LIN (refer to Bus Instrument (TX Type for LIN) on page 136)
TX status	To display and configure the transmission options of TX messages/frames/PDUs.	CAN (refer to Bus Instrument (TX Status Type for CAN) on page 119)
		LIN (refer to Bus Instrument (TX Status Type for LIN) on page 134)

Related topics

References

Bus Instruments (Bus Manager)..... 104

How to Configure a TX Message/Frame/PDU and Transmit It on the Bus

Objective

To configure a TX message/frame/PDU and transmit it on the bus via the Bus Instrument (TX type).

Preconditions

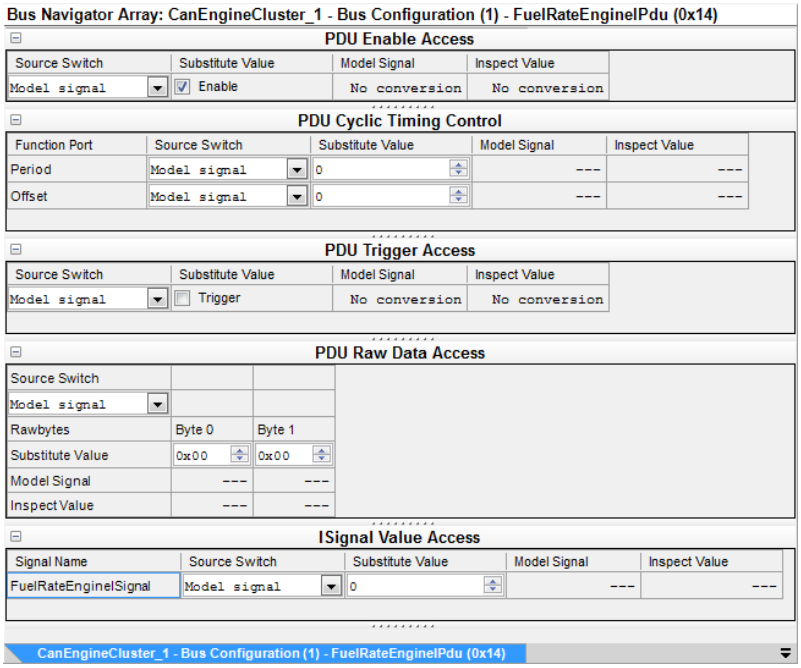
You have created a Bus Instrument (TX type) for the TX message/frame/PDU. Refer to How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs on page 50.

Method

To configure a TX message/frame/PDU and transmit it on the bus

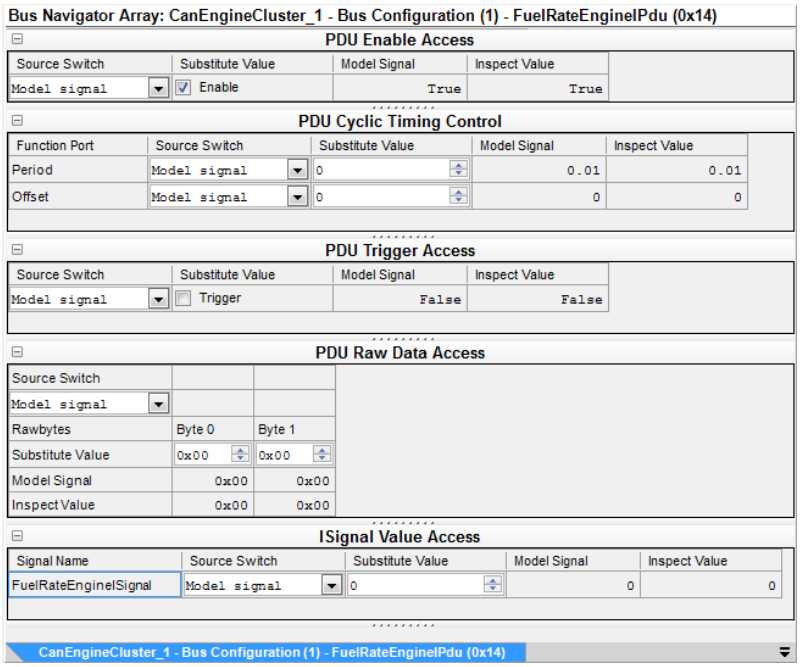
1 Open or select a layout that contains a Bus Instrument (TX type) for the relevant message/frame/PDU and select the instrument or instrument tab.

The following illustration shows an example.



2 On the Home ribbon, click Status Control – Go Online to start online calibration and to activate the instrument.

The Bus Instrument (TX type) displays the settings specified in the bus configuration for the selected message/frame/PDU. Refer to the following example.



3 Set **Source Switch** to **Substitute Value** for the settings you want to manipulate.

4 Change the settings according to your requirements, for example:
   - Select **Enable** in the **PDU Enable Access** region to enable transmission of the PDU.
   - In the **PDU Cyclic Timing Control** region, change the period and the offset in the corresponding **Substitute Value** fields.
   - Change the value of the raw bytes in the corresponding **Substitute Value** fields.

Tip

The available manipulation options depend on the settings specified in the Bus Manager. For an overview of all available options, refer to:

- Bus Instrument (TX Type for CAN) on page 121
- Bus Instrument (TX Type for LIN) on page 136

Result

- The Bus Instrument (TX type) displays configurable settings of the TX message/frame/PDU.
- You configured the TX message/frame/PDU:
  - You enabled its transmission.
  - You changed transmission settings.

Related topics

HowTos

How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs..... 50

References

Bus Instrument (TX Type for CAN)..... 121
Bus Instrument (TX Type for LIN)..... 136

How to Specify Global Transmission Settings for a CAN or LIN Controller

Objective

To specify basic transmission settings globally for all the TX PDUs of a CAN or LIN communication matrix via the Bus Instrument (TX status type).

Basics of transmission states

In the Bus Instrument (TX type) for a *single* TX message, you can enable and configure the transmission of a specific TX message.

To select *multiple* TX messages for transmission and clear others for the communication matrix of a CAN or LIN controller at the same time, you can use the Bus Instrument (TX status type).

Method

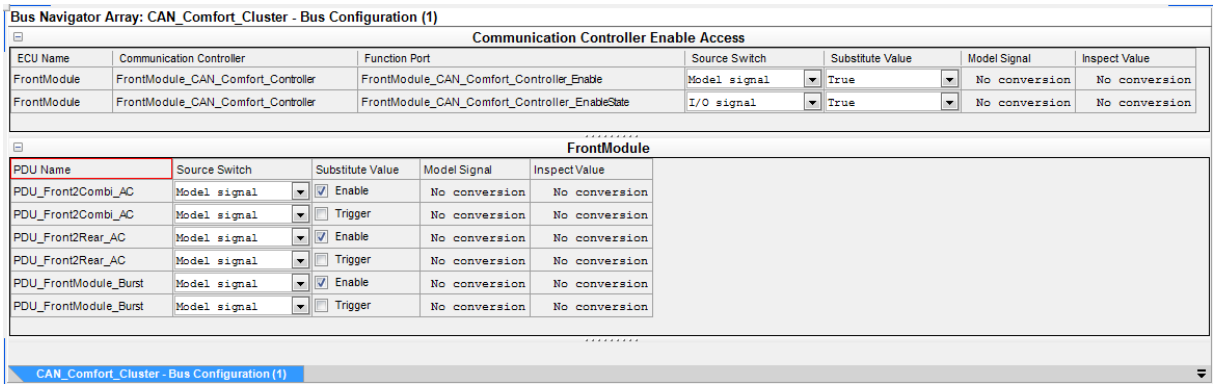
To specify global transmission settings for a CAN or LIN controller

1 From the Bus Navigator tree, drag a CAN or LIN communication cluster node to an empty area of an existing layout.

– Or –

From the context menu of a CAN or LIN communication cluster node, select **Generate TX Status Layout** to create a new layout with a Bus Instrument (TX status type).

2 ControlDesk opens the Bus Instrument (TX status type) for the related communication matrix. Refer to the following example for a CAN controller.



Tip

Only a limited number of layouts can be open at one time. If you encounter problems while opening Bus Instruments, reduce the number of open layouts.

3 On the **Home** ribbon, click **Status Control – Go Online** to start online calibration and to activate the instrument.

4 Set **Source Switch** to **Substitute Value** for the communication controller function ports and PDUs you want to manipulate.

5 Set **Substitute Value** to **True** to enable a communication controller.

6 Select/clear **Enable** in the **Substitute Value** column to enable/disable the transmission of a PDU. Select **Trigger** in the **Substitute Value** column to enable triggered transmission of a PDU.

Tip

The actual available manipulation options depend on the settings specified in the Bus Manager. For an overview of all the available options, refer to:

- Bus Instrument (TX Status Type for CAN) on page 119
- Bus Instrument (TX Status Type for LIN) on page 134

Result

You specified global transmission settings for a CAN or LIN controller.

Related topics

References

Bus Instrument (TX Status Type for CAN)..... 119
Bus Instrument (TX Status Type for LIN)..... 134

Instruments for Bus Communication Modeled with a Blockset

Where to go from here

Information in this section

Overview of Available Bus Instrument Types..... 63
The Bus Navigator provides different Bus Instrument types to handle bus communication modeled with a blockset.

How to Configure a TX Message/Frame/PDU and Transmit It on the Bus.....64
To configure a TX message/frame/PDU and transmit it on the bus via the Bus Instrument (TX type).

How to Create a Bus Instrument (Gateway Type) for a CAN Controller Gateway.....67
To create a Bus Instrument (Gateway type) for a CAN controller gateway.

How to Specify Global Transmission Settings for a CAN or FlexRay Controller.....69
To specify basic transmission settings globally for all the TX messages/PDUs of a CAN or FlexRay controller via the Bus Instrument (TX status type).

How to Display the Error Status for all the RX PDUs of a FlexRay Controller.....71
You can use the Bus Instrument (Error status type for FlexRay) to display the error status for all the PDUs of a FlexRay controller.

Overview of Available Bus Instrument Types

Introduction

The Bus Navigator provides different Bus Instrument types to handle bus communication modeled with a blockset.

Instrument types

The following table provides an overview of all the Bus Instrument types that are available for bus communication modeled with a blockset.

Bus Instrument Type	Function	Bus Types
Error Status	To display the error value and the slot status for all the PDUs of a FlexRay communication cluster.	FlexRay (refer to Bus Instrument (Error Status Type for FlexRay) on page 155)
Gateway	To specify whether messages are gatewayed between two CAN buses and change the specified gateway filters during run time.	CAN (refer to Bus Instrument (Gateway Type for CAN) on page 140)

Bus Instrument Type	Function	Bus Types
RX	To display information on RX messages/frames/PDUs.	CAN (refer to Bus Instrument (RX Type for CAN) on page 142)
		LIN (refer to Bus Instrument (RX Type for LIN) on page 169)
		FlexRay (refer to Bus Instrument (RX Type for FlexRay) on page 157)
TX	To display, configure and transmit TX messages/frames/PDUs.	CAN (refer to Bus Instrument (TX Type for CAN) on page 147)
		LIN (refer to Bus Instrument (TX Type for LIN) on page 171)
		FlexRay (refer to Bus Instrument (TX Type for FlexRay) on page 160)
TX status	To display and configure the transmission options of TX messages/frames/PDUs.	CAN (refer to Bus Instrument (TX Status Type for CAN) on page 145)
		FlexRay (refer to Bus Instrument (TX Status Type for FlexRay) on page 166)

Related topics

References

Bus Instruments (Blockset)..... 139

How to Configure a TX Message/Frame/PDU and Transmit It on the Bus

Objective

To configure a TX message/frame/PDU and transmit it on the bus via the Bus Instrument (TX type).

Preconditions

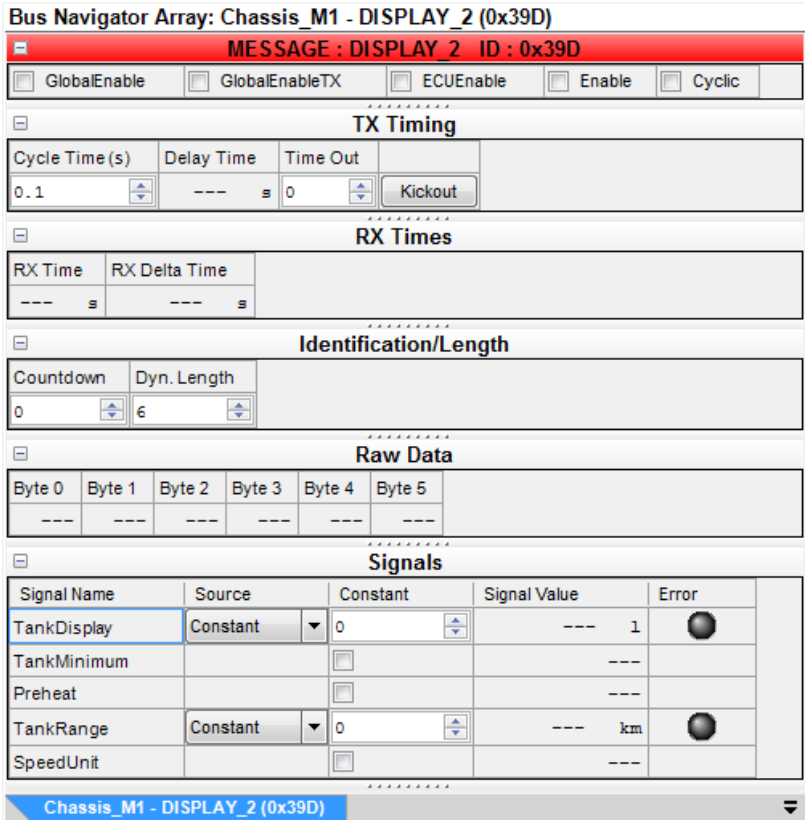
You have created a Bus Instrument (TX type) for the TX message/frame/PDU. Refer to How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs on page 50.

Method

To configure a TX message/frame/PDU and transmit it on the bus

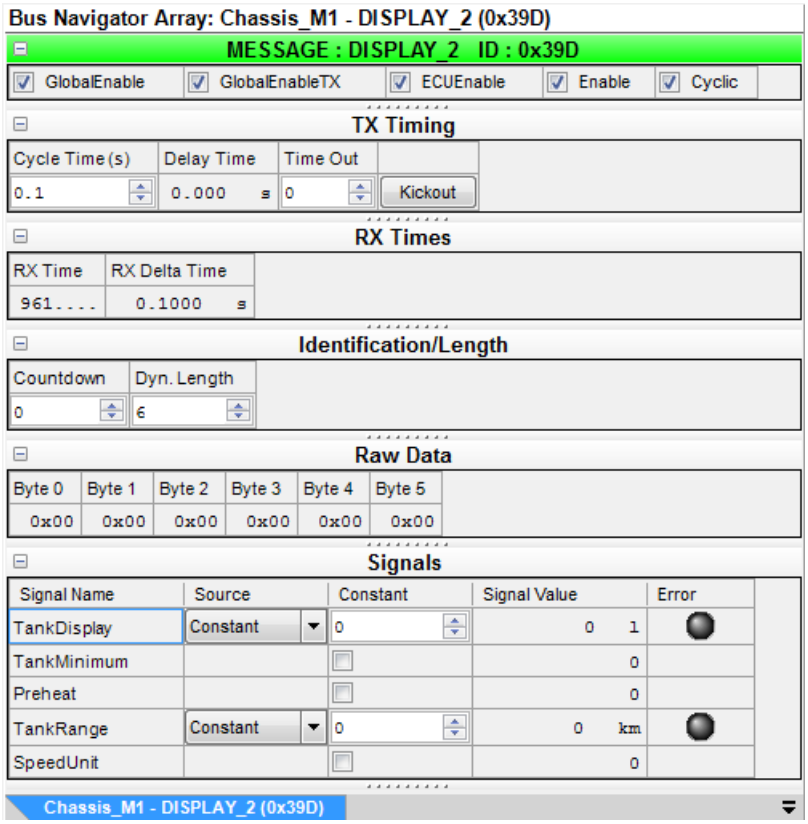
1 Open or select a layout that contains a Bus Instrument (TX type) for the relevant message/frame/PDU and select the instrument or instrument tab.

The following illustration shows an example.



2 On the Home ribbon, click Status Control – Go Online to start online calibration and to activate the instrument.

The Bus Instrument (TX type) displays the settings specified in the bus configuration for the selected message, see the following example.



3 For a bus configuration that is based on a real-time application, the header color of the instrument indicates whether the bus communication or a specific message is active or inactive:

If the header is red, select the available enable options. The possible displayed enable options are GlobalEnable, GlobalEnableTX, ECU, and Enable.

Tip

If the header remains red, the message/frame/PDU is configured to be enabled via ControlDesk *and* model input. In this case, the message/frame/PDU must be enabled in the real-time model, too.

4 If the value of RX Time does not change, select Cyclic and specify a Cycle Time.

Only the running RX time indicates that the message/frame/PDU is transmitted on the bus.

Tip

- If Cyclic is not displayed, it is not enabled to be set via ControlDesk.
- If the RX time is still not running, cyclic message/frame/PDU transmission must be enabled via the real-time model, too.

5 To transmit the message/frame/PDU once immediately on the bus, click **Kickout**.

Tip

If **Kickout** is not displayed, it is not enabled to be set via ControlDesk.

6 Manipulate the signals of the message/frame/PDU, for example, specify a **Constant** value or select a manipulation option from the **Source** list.

Tip

The actual available manipulation options depend on the settings specified in the real-time model. For an overview of all the available options, refer to:

- Bus Instrument (TX Type for CAN) on page 147
- Bus Instrument (TX Type for LIN) on page 171
- Bus Instrument (TX Type for FlexRay) on page 160

Result

- The Bus Instrument (TX type) displays information on the TX message/frame/PDU.
- You configured the TX message/frame/PDU:
  - You specified its transmission.
  - You specified the manipulation of signal values by specifying the constant values and/or selecting signal manipulation options.
- If the message/frame/PDU cannot be transmitted on the bus, the message/frame/PDU header is displayed in red.

Related topics

HowTos

How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs..... 50

References

Bus Instrument (TX Type for CAN)..... 147
Bus Instrument (TX Type for FlexRay)..... 160
Bus Instrument (TX Type for LIN)..... 171

How to Create a Bus Instrument (Gateway Type) for a CAN Controller Gateway

Objective

To create a Bus Instrument (Gateway type) for a CAN controller gateway.

Basics on gatewaying in the Bus Navigator

If a gateway is configured in the real-time model using the RTICANMM Gateway block, the gateway is displayed as a node on the same level as the processor node in the Bus Navigator tree. You can specify as many gateways as you like in the real-time model. Active gateways are labeled .

Two types of gateway nodes are possible:

- The CAN controller gateway with the connected controllers displayed in the gateway name.

  A CAN controller gateway is always active. You can always open layouts for this gateway.
- The MainBlock gateway with the connected MainBlocks displayed in the gateway name.

  A MainBlock gateway is active when the connected MainBlocks are active.

You can use the Bus Instrument (Gateway type) to exclude messages from being exchanged between two CAN buses.

Preconditions

- The real-time model must contain at least one CAN controller with at least one configured CAN message.
- Gatewaying must be specified in the real-time model. Refer to Block Description (RTICANMM Gateway) (RTI CAN MultiMessage Blockset Reference 🕮).
- A real-time platform must be added to the experiment, and a corresponding SDF file, generated for a real-time model created with the RTI CAN MultiMessage Blockset must be added to the platform. Refer to How to Add a Platform/Device to an Experiment (ControlDesk Platform Management 🕮).

Method

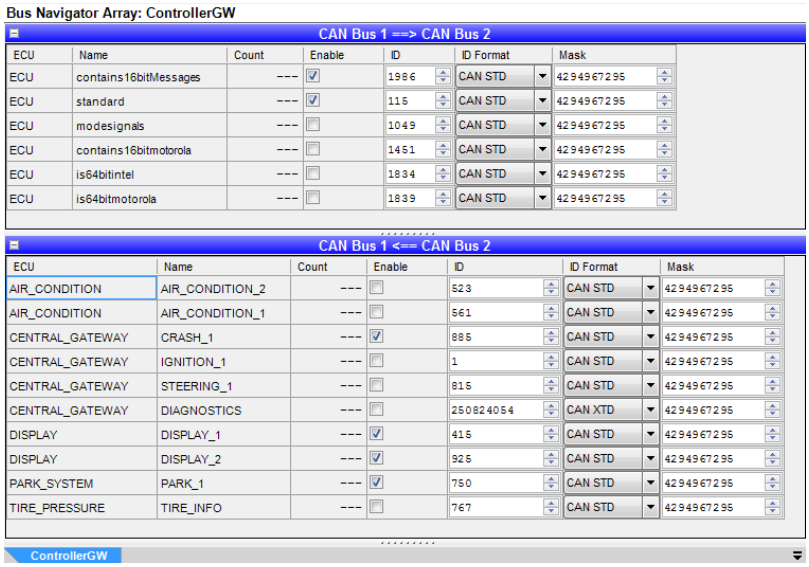
To create a Bus Instrument (Gateway type) for a CAN controller gateway

1. From the Bus Navigator tree, drag a gateway node onto an empty area of an existing layout.

   – Or –

   From the context menu of a gateway node, select **Generate Gateway Layout** to create a new layout containing a Bus Instrument (Gateway type).

ControlDesk opens the Bus Instrument (Gateway type), see the following example.



Tip

Only a limited number of layouts can be open at one time. If you encounter problems while opening Bus Instruments, reduce the number of open layouts.

2 On the **Home** ribbon, click **Status Control – Go Online** to start online calibration and activate the instrument.

Result

The Bus Instrument (Gateway type) displays the gateway direction and message filters for gatewaying. Via the message filters, you can manipulate the gateway settings defined in the RTICANMM Gateway block, exclude messages from being gatewayed, etc.

Related topics

References

Bus Instrument (Gateway Type for CAN)..... 140
RTICANMM Gateway (RTI CAN MultiMessage Blockset Reference 🕮)

How to Specify Global Transmission Settings for a CAN or FlexRay Controller

Objective

To specify basic transmission settings globally for all the TX messages/PDUs of a CAN or FlexRay controller via the Bus Instrument (TX status type).

Basics of transmission states

In the Bus Instrument (TX type), you can enable and configure the transmission of *a specific TX message*. However, you can also enable some global transmission settings: the communication of the controller and the transmission of all TX messages.

To select *several TX messages* for transmission and clear others for the communication matrix of a controller at the same time, you must specify their transmission states in the Bus Instrument (TX status type).

Precondition

The experiment must contain a bus configuration with at least one CAN or FlexRay controller and one configured CAN or FlexRay TX message. Refer to How to Display a Bus Configuration with the Bus Navigator on page 34.

Method

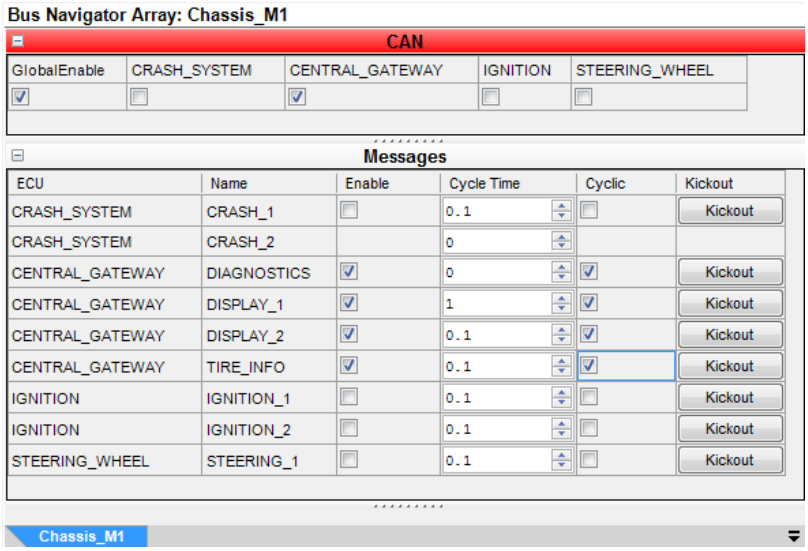
To specify global transmission settings for a CAN or FlexRay controller

1 From the Bus Navigator tree, drag a CAN or FlexRay communication cluster node onto an empty area of an existing layout.

– Or –

From the context menu of a CAN or FlexRay communication cluster node, select **Generate TX Status Layout** to create a new layout and place a Bus Instrument (TX status type) on it.

2 ControlDesk opens the Bus Instrument (TX status type) for the related communication matrix, see the following example.



Tip

Only a limited number of layouts can be open at one time. If you encounter problems while opening Bus Instruments, reduce the number of open layouts.

3 Start online calibration to activate the instrument.

4 If cleared, select **GlobalEnable** to enable bus communication for the controller and all TX messages.

   Bus communication is only possible if **GlobalEnable** is selected.

5 Select or clear ECUs (such as CENTRAL_GATEWAY in the example above) to enable or disable ECUs and all their messages.

6 Select or clear **Enabled** for the messages whose transmission you want to enable or disable.

7 Specify a **Cycle time** and select **Cyclic** for messages you want to transmit cyclically.

8 Click **Kickout** for messages you want to transmit once immediately.

Tip

The actual available manipulation options depend on the settings specified in the real-time model. For an overview of all the available options, refer to Bus Instrument (TX Status Type for CAN) on page 145 or Bus Instrument (TX Status Type for FlexRay) on page 166.

Result

You specified the global transmission settings for a CAN or FlexRay controller.

Related topics

References

Bus Instrument (TX Status Type for CAN)..... 145
Bus Instrument (TX Status Type for FlexRay)..... 166

How to Display the Error Status for all the RX PDUs of a FlexRay Controller

Objective

You can use the Bus Instrument (Error status type for FlexRay) to display the error status for all the PDUs of a FlexRay controller.

Precondition

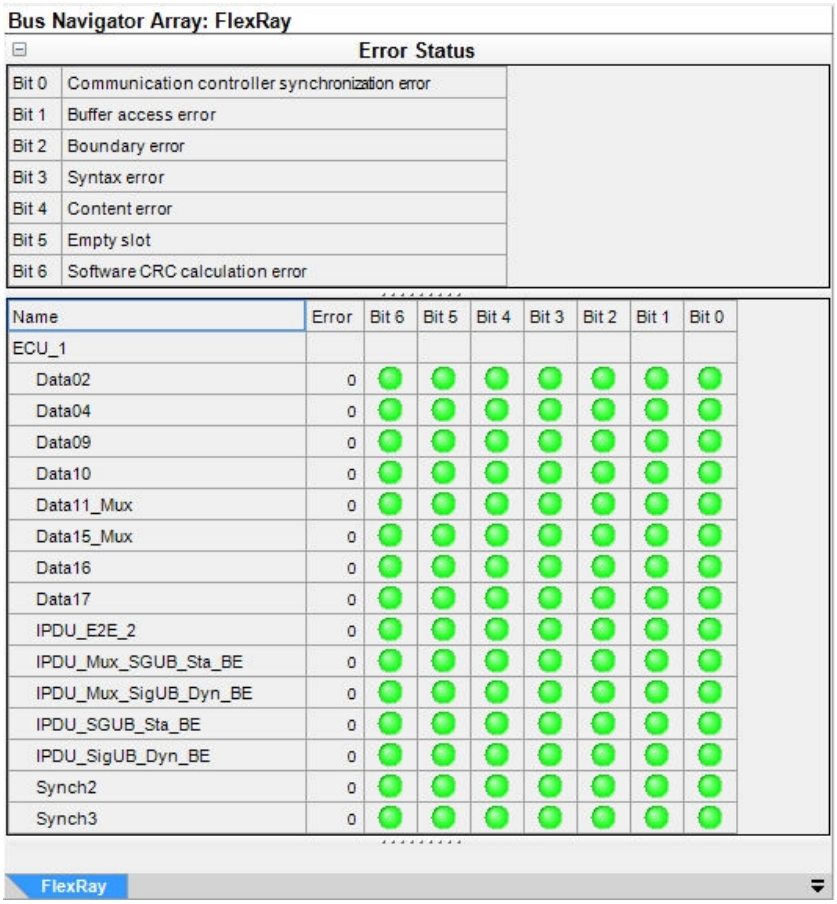
The experiment must contain a bus configuration with at least one FlexRay controller and at least one configured PDU.

Method

To display the error status for all the RX PDUs of a FlexRay controller

1 From the context menu of a FlexRay communication matrix node, select **Generate Error Status Layout** to create a new layout containing a Bus Instrument (Error status type).

2 ControlDesk opens the Bus Instrument (Error status type) for the selected communication matrix, see the following example.



Tip

Only a limited number of layouts can be open at one time. If you encounter problems while opening Bus Instruments, reduce the number of open layouts.

3 Start online calibration to activate the instrument.

Result

The Bus Instrument (Error status type for FlexRay) displays the error value and an LED for each error bit for all the RX and RX/TX PDUs of the selected FlexRay communication matrix.

Related topics

References

Bus Instrument (Error Status Type for FlexRay)..... 155

Instruments Related to Bus Monitoring Devices

Where to go from here

Information in this section

Overview of Available Bus Instrument Types.....73
The Bus Navigator provides different Bus Instrument types to handle CAN, LIN and Ethernet bus communication using a bus monitoring device.

How to Configure a CAN TX Message and Transmit It on the Bus.....74
To configure a CAN TX message and transmit it on the bus via the Bus Instrument (TX type for CAN).

Overview of Available Bus Instrument Types

Introduction

The Bus Navigator provides different Bus Instrument types to handle CAN, LIN and Ethernet bus communication using a bus monitoring device.

Instrument types

The following table provides an overview of all the Bus Instruments that are available for accessing bus communication via a bus monitoring device.

Bus Instrument Type	Function	Bus Types
RX	To display information on RX messages, frames, or PDUs.	CAN (refer to Bus Instrument (RX Type for CAN) on page 175)
		Ethernet (refer to Bus Instrument (RX Type for Ethernet) on page 178)
		LIN (refer to Bus Instrument (RX Type for LIN) on page 180)
TX	To display, configure and transmit TX messages, frames, or PDUs.	CAN (refer to Bus Instrument (TX Type for CAN) on page 176)

Related topics

HowTos

How to Configure a CAN TX Message and Transmit It on the Bus.....74

References

Bus Instruments (Related to Bus Monitoring Devices).....174

How to Configure a CAN TX Message and Transmit It on the Bus

Objective

To configure a CAN TX message and transmit it on the bus via the Bus Instrument (TX type for CAN).

Preconditions

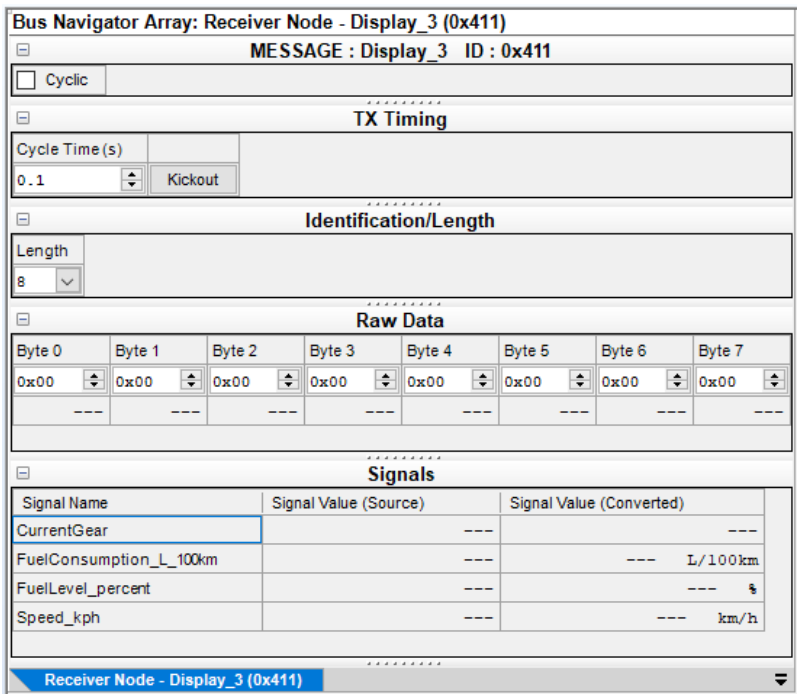
- You have created a Bus Instrument (TX type for CAN) for the TX message. Refer to How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs on page 50.
- You have activated the Import Options – Bus Import Options – Generate raw byte variables for each CAN message via the Variables page of the ControlDesk Options dialog.

Method

To configure a CAN TX message and transmit it on the bus

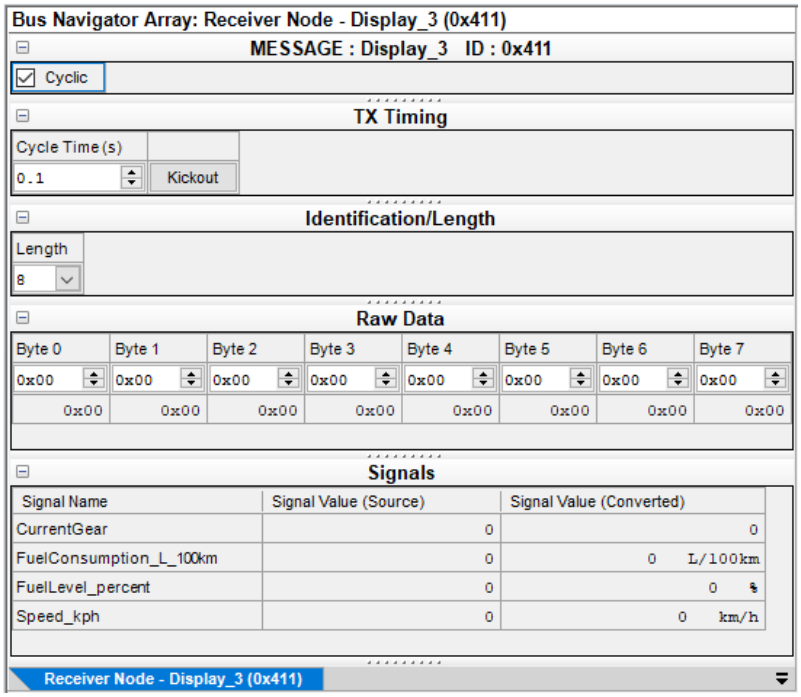
1 Open or select a layout that contains a Bus Instrument (TX type for CAN) for the relevant message and select the instrument or instrument tab.

The following illustration shows an example.



2 On the Home ribbon, click Status Control – Go Online to start online calibration and to activate the instrument.

The Bus Instrument (TX type for CAN) displays values in the Raw Data and Signals regions. Refer to the following illustration.



If no values are displayed in the Raw Data and Signals regions, select Cyclic and specify a Cycle Time.

3 To transmit the message once immediately on the bus, click Kickout.

4 Configure the settings of the message according to your requirements, for example, change the message length via the Identification/Length region.

Result

- The Bus Instrument (TX type for CAN) displays information on the TX message.
- You configured the TX message:
  - You specified its transmission.
  - You configured the message settings according to your requirements.

Related topics

Basics

Bus Instrument (TX Type for CAN)..... 176

HowTos

How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs..... 50

Monitoring and Logging Bus Communication

Introduction

The Bus Navigator lets you monitor and log bus communication. You can also specify filters to limit the monitored or logged bus communication.

Where to go from here

Information in this section

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
You can monitor, log, and filter bus communication by adding monitor, logger, and filter nodes to the Bus Navigator tree.

Dependencies Regarding Monitoring, Logging, and Filtering Bus Communication.....80
Monitoring, logging, and filtering bus communication depends on the bus type and on the interface you use.

How to Monitor Bus Communication.....82
To perform bus monitoring.

How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN).....85
To specify a filter for monitoring and/or logging selected bus communication data.

Example of Specifying Filters for Bus Monitoring or Logging (CAN, LIN).....88
Showing some examples of specifying filters for bus monitoring or logging.

Basics on Filtering Ethernet Bus Communication.....91
You can reduce the amount of data captured by the Ethernet Bus Monitoring device by specifying a filter for Ethernet bus communication.

Examples of Filtering Ethernet Bus Communication.....94
Some examples show you how to define Ethernet filter expressions according to the Berkeley Packet Filter syntax.

How to Log Raw Bus Communication Data.....95
To log raw bus communication data.

How to Display Bus Statistics.....97
To enable the display of bus statistics during bus monitoring.

Basics on Monitoring, Logging, and Filtering Bus Communication

Introduction

You can monitor, log, and filter bus communication by adding monitor, logger, and filter nodes to the Bus Navigator tree.

Note

You are responsible for ensuring that ControlDesk's monitoring and logging function is used only in accordance with the intended purpose and the applicable legal regulations.
Refer to:

- Important Information when Monitoring and Logging Ethernet Traffic (ControlDesk Introduction and Overview 🕮)
- Important Information on Using an External Decoder for Interpreting Ethernet Raw Data (ControlDesk Introduction and Overview 🕮)

Levels of monitoring, logging, and filtering

You can add monitor, logger, and filter nodes to different hierarchy levels of the Bus Navigator tree depending on the part of the bus communication that you want to access. Refer to the following table.

Hierarchy Level[1)]	Symbol	Accessed Part of Bus Communication
System node		Bus communication of the entire system.
Hardware node		Bus communication of all the controller nodes that belong to the selected hardware node.
Controller node		Bus communication of the selected controller node.[2)]

1) Refer to Bus Configuration Structure in the Bus Navigator Tree on page 19.
2) For blockset-based CAN or LIN bus communication, a controller can have several communication matrices (refer to Activating Communication Matrices for Blockset-Based Bus Communication on page 44). In this case, the bus communication of the *active* communication matrix is accessed.

Example The illustration in Monitoring raw and physical data on page 78 shows an example of monitor nodes that have been added to a controller node. In this case, adding the monitor nodes to higher hierarchy levels does not change the amount of data that is monitored because each system node contains only one hardware node with one controller node.

Monitoring bus communication

When you add a monitor node ( ) to the Bus Navigator tree, ControlDesk opens a monitoring list in the working area.

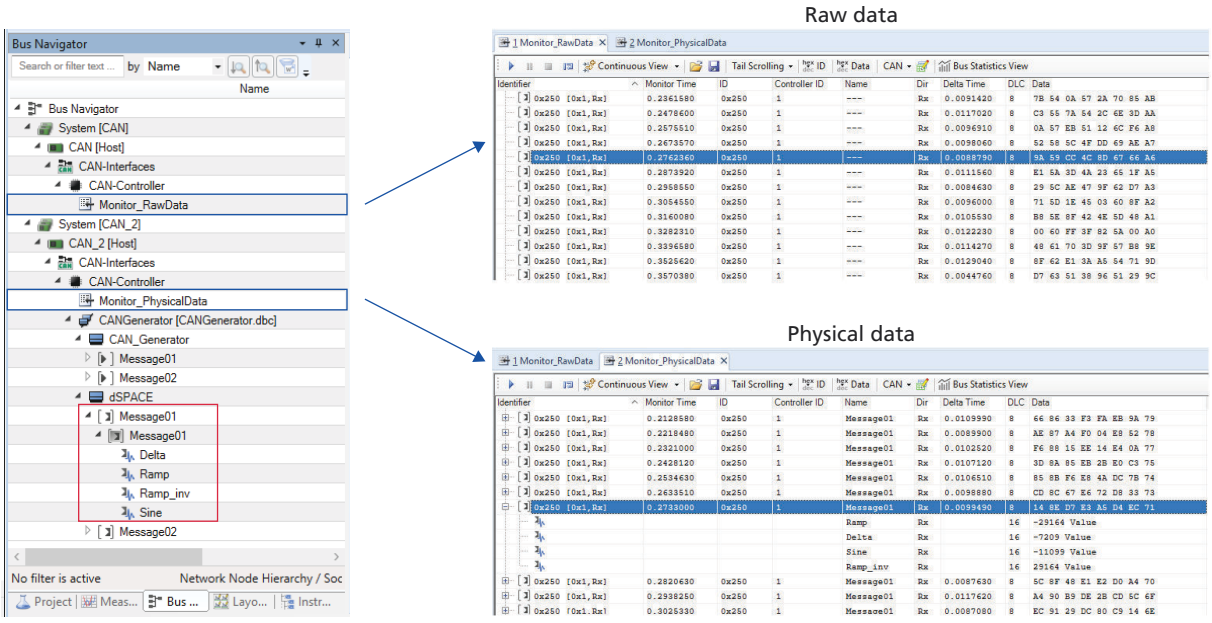
The monitoring list displays data from the monitored bus communication either raw or converted to a physical format. This depends on whether you added a *variable description file* to the device that performs the monitoring.

Monitoring raw and physical data If you are using a bus monitoring device and you want to display *physical* data in the monitoring list or use the functions of a Bus Instrument, you must add a variable description file to the device. Without a variable description file, you can only monitor the *raw* data from the bus communication.

A variable description file can provide message names and organize the message data into signals with physical values.

The illustration below compares the monitoring of raw and physical data from the `CANGenerator` (refer to CANGenerator Program (ControlDesk Introduction and Overview 🕮)) demo program. The data is monitored via two CAN Bus Monitoring devices: The CAN device monitors the data without a variable description file. Therefore, the corresponding monitoring list `Monitor_RawData` displays only raw data.

In the case of the CAN_2 device, the `CANGenerator.dbc` variable description file was loaded to it, which converts the messages into signals with physical values, as displayed in the `Monitor_PhysicalData` monitoring list.



For an overview of supported variable description file formats, refer to Variable Descriptions Supported by ControlDesk (ControlDesk Variable Management 🕮).

Bus statistics For CAN, CAN FD, and LIN bus monitoring, you can display bus statistics in the monitoring list. Depending on your setup, you can also use a logger (refer to Logging bus communication on page 79) to save the bus statistics together with the logged data.

Activating monitor nodes To control the starting behavior of monitor nodes at different hierarchy levels, you can activate and deactivate them separately. If you start one monitoring node, all the activated monitoring nodes at lower levels

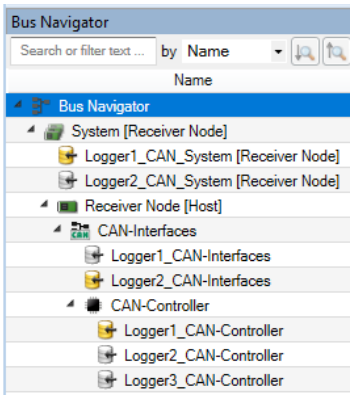
are also started. If you want to prevent a lower-level monitor node from being started, you must deactivate it.

Logging bus communication

You can add logger nodes (⇥) to different hierarchy levels of the Bus Navigator tree to log the corresponding parts of the *raw* bus communication in a file.

For more information on configuring a logger, refer to Add/Edit Logger Dialog on page 199.

Activating logger nodes You can add multiple logger nodes to each system, hardware, controller, or filter node. However, only one logger can be active at a time for each of these nodes. Therefore, activating one logger automatically deactivates all other loggers belonging to the same node. Refer to the following illustration, where active logger nodes are yellow.



Replaying logged bus communication For CAN and Ethernet bus communication, you can add replay nodes (⇥) to controller nodes of the Bus Navigator tree to replay logged bus communication. Refer to Basics on Replaying CAN and Ethernet Bus Communication on page 99.

Filtering bus communication

You can add filter nodes (▼) to different hierarchy levels of the Bus Navigator tree to filter the corresponding parts of the bus communication. You can then add a monitor or a logger node to the filter node to monitor or log only the bus communication that passes the filter.

In addition, you can add a filter node to a CAN replay node to replay only selected parts of a logged bus communication.

Related topics

Basics

Basics on Replaying CAN and Ethernet Bus Communication.....99
Bus Configuration Structure in the Bus Navigator Tree.....19

Dependencies Regarding Monitoring, Logging, and Filtering Bus Communication..... 80

HowTos

How to Display Bus Statistics..... 97

References

Activate (Logger)..... 186
Activate (Monitor)..... 187
Add (CAN/Ethernet) Replay..... 189
Add (CAN/LIN/Ethernet) Filter..... 190
Add (CAN/LIN/Ethernet) Logger..... 191
Add (CAN/LIN/Ethernet) Monitor..... 193
Decoding View..... 217
Monitoring List..... 240

Dependencies Regarding Monitoring, Logging, and Filtering Bus Communication

Introduction

Monitoring, logging, and filtering bus communication depends on the bus type and on the interface you use. The following sections provide an overview for each bus type.

CAN

Interface	**Monitoring** **Logging** **Filtering**
Platforms: ▪ DS100x-based system ▪ MicroLabBox ▪ MicroAutoBox II ▪ SCALEXIO[1)]	✓
Monitoring device in connection with a PC-based interface[2)]	✓
Monitoring device in connection with a bus interface of SCALEXIO, MicroAutoBox III, or VEOS[3)]	✓

1) Not for bus communication modeled with the Bus Manager.
2) CAN FD is supported by the dSPACE DCI-CAN2 or DCI-CAN/LIN1 interfaces and various interfaces of Vector Informatik, such as VN16xx, VN5610, and VN8900. For general information on supported interfaces, refer to Supported CAN Interfaces (ControlDesk Platform Management).
3) You can also display and log bus statistics.

Ethernet

Interface	Monitoring Logging Filtering
Platforms: ▪ DS100x-based system ▪ MicroLabBox ▪ MicroAutoBox II ▪ MicroAutoBox III ▪ SCALEXIO	–
Monitoring device in connection with a PC-based interface[1)]	✓
Monitoring device in connection with a bus interface of SCALEXIO, MicroAutoBox III, or VEOS	✓

1) For information on supported interfaces, refer to Supported Ethernet Interfaces (ControlDesk Platform Management).

Ethernet decoding The Bus Navigator supports Ethernet decoding according to the SOME/IP-SD protocol and Ethernet standard protocols, such as IPv4, TCP, and UDP. However, decoding according to a variable description is not supported.

FlexRay

Monitoring is not supported for FlexRay bus communication.

LIN

Interface	Monitoring Logging Filtering
Platforms: ▪ DS100x-based system ▪ MicroLabBox ▪ MicroAutoBox II ▪ SCALEXIO[1)]	✓
Monitoring device in connection with a PC-based interface[2)]	✓
Monitoring device in connection with a bus interface of SCALEXIO[3)], MicroAutoBox III, or VEOS[3)]	✓

1) Not for bus communication modeled with the Bus Manager.
2) For information on supported interfaces, refer to Supported LIN Interfaces (ControlDesk Platform Management).
3) You can also display and log bus statistics.

Related topics

Basics

Accessing Bus Communication with a Bus Monitoring Device.....37
Basics on Filtering Ethernet Bus Communication.....91

Basics on Replaying CAN and Ethernet Bus Communication.....99

HowTos

How to Display Bus Statistics.....97
How to Log Raw Bus Communication Data.....95
How to Monitor Bus Communication.....82
How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN).....85

How to Monitor Bus Communication

Objective

To perform bus monitoring.

Monitor node settings

When you add a monitor node, you can specify its settings before you start monitoring.

You can specify the following monitor node settings:

- Monitor name: Editable name of the monitor.
- Ring buffer settings: When the monitored data exceeds a specified number of messages/frames/PDUs, monitoring can either stop or continue and overwrite old monitor data.
- Capture settings: Editable update rate.

Depending on the bus type, additional settings might be displayed. Refer to Add/Edit Monitor Dialog on page 201.

Preconditions

You must have an experiment with a bus configuration.

Method

To monitor bus communication

1. For unfiltered monitoring, right-click a system, hardware, or controller node to open its context menu.

   For filtered monitoring, open the context menu of a filter node.

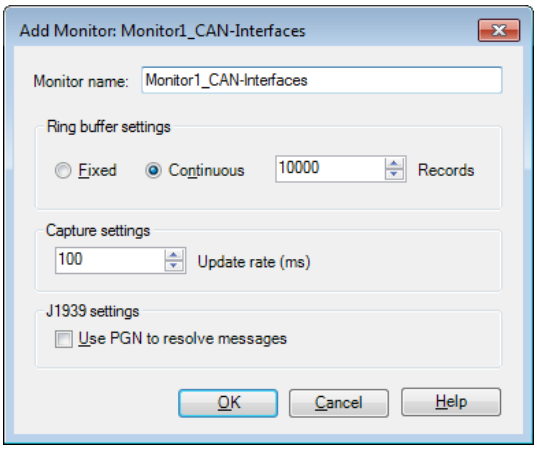
2. On the context menu, click **Add Monitor**.

   ControlDesk adds a monitor node below the selected node and opens a monitoring list in the working area.

   Note

   If you close the monitoring list, ControlDesk removes the monitor node from the Bus Navigator tree.

In addition, ControlDesk opens the **Add Monitor** dialog:



3 Specify the settings in the **Add Monitor** dialog.

> **Tip**
>
> You can change the settings later by selecting **Edit** from the context menu of the monitor node.

4 Click **OK**.

ControlDesk closes the dialog.

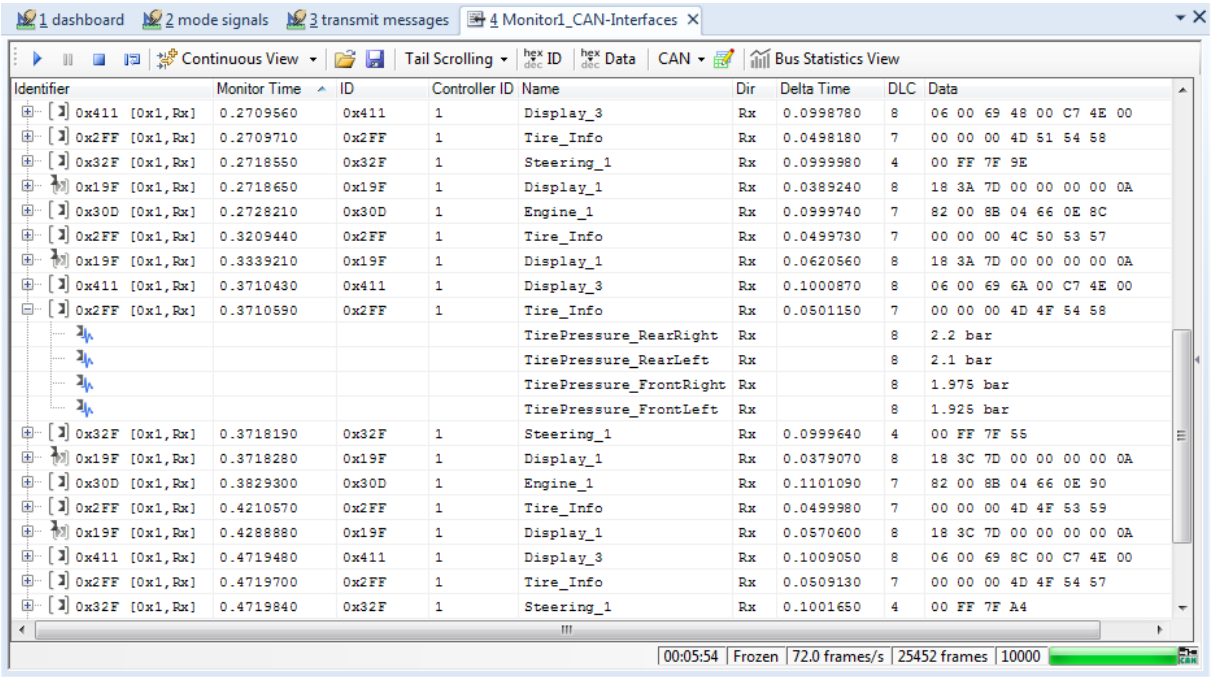
5 Right-click the monitor node, and select **Activate Monitor**.

6 On the **Home** ribbon, click **Status Control – Go Online** to start online calibration.

> **Tip**
>
> For monitoring bus communication via a bus monitoring device, you do not have to start online calibration, the **Connected** state is sufficient.

7 Right-click the monitor node and select **Start** or click in the monitoring list.

Bus monitoring starts and the monitoring list displays the monitored bus communication (for an animated graphic, refer to dSPACE Help).



Identifier	Monitor Time	ID	Controller ID	Name	Dir	Delta Time	DLC	Data
0x411 [0x1,Rx]	0.2709560	0x411	1	Display_3	Rx	0.0998780	8	06 00 69 48 00 C7 4E 00
0x2FF [0x1,Rx]	0.2709710	0x2FF	1	Tire_Info	Rx	0.0498180	7	00 00 00 4D 51 54 58
0x32F [0x1,Rx]	0.2718550	0x32F	1	Steering_1	Rx	0.0999980	4	00 FF 7F 9E
0x19F [0x1,Rx]	0.2718650	0x19F	1	Display_1	Rx	0.0389240	8	18 3A 7D 00 00 00 00 0A
0x30D [0x1,Rx]	0.2728210	0x30D	1	Engine_1	Rx	0.0999740	7	82 00 8B 04 66 0E 8C
0x2FF [0x1,Rx]	0.3209440	0x2FF	1	Tire_Info	Rx	0.0499730	7	00 00 00 4C 50 53 57
0x19F [0x1,Rx]	0.3339210	0x19F	1	Display_1	Rx	0.0620560	8	18 3A 7D 00 00 00 00 0A
0x411 [0x1,Rx]	0.3710430	0x411	1	Display_3	Rx	0.1000870	8	06 00 69 6A 00 C7 4E 00
0x2FF [0x1,Rx]	0.3710590	0x2FF	1	Tire_Info	Rx	0.0501150	7	00 00 00 4D 4F 54 58
				TirePressure_RearRight	Rx		8	2.2 bar
				TirePressure_RearLeft	Rx		8	2.1 bar
				TirePressure_FrontRight	Rx		8	1.975 bar
				TirePressure_FrontLeft	Rx		8	1.925 bar
0x32F [0x1,Rx]	0.3718190	0x32F	1	Steering_1	Rx	0.0999640	4	00 FF 7F 55
0x19F [0x1,Rx]	0.3718280	0x19F	1	Display_1	Rx	0.0379070	8	18 3C 7D 00 00 00 00 0A
0x30D [0x1,Rx]	0.3829300	0x30D	1	Engine_1	Rx	0.1101090	7	82 00 8B 04 66 0E 90
0x2FF [0x1,Rx]	0.4210570	0x2FF	1	Tire_Info	Rx	0.0499980	7	00 00 00 4D 4F 53 59
0x19F [0x1,Rx]	0.4288880	0x19F	1	Display_1	Rx	0.0570600	8	18 3C 7D 00 00 00 00 0A
0x411 [0x1,Rx]	0.4719480	0x411	1	Display_3	Rx	0.1009050	8	06 00 69 8C 00 C7 4E 00
0x2FF [0x1,Rx]	0.4719700	0x2FF	1	Tire_Info	Rx	0.0509130	7	00 00 00 4D 4F 54 57
0x32F [0x1,Rx]	0.4719840	0x32F	1	Steering_1	Rx	0.1001650	4	00 FF 7F A4

Tip

- You can change the view of the monitoring list via the toolbar or the column header context menu.
- Only a limited number of layouts can be open simultaneously. If you encounter problems while opening the monitoring list, reduce the number of open layouts.
- If the monitoring list does not display message instances, the bus communication might be inactive. For CAN, you can open the Bus Instrument (TX status type) of a communication cluster to enable bus communication. Refer to How to Specify Global Transmission Settings for a CAN or FlexRay Controller on page 69 if you modeled the bus communication with a blockset, or to How to Specify Global Transmission Settings for a CAN or LIN Controller on page 61 if you modeled the bus communication with the Bus Manager.
- Blue lines are displayed during the monitoring of Ethernet bus communication if frames are lost. The **Summary** column informs you about the occurrence of buffer overflows and lost frames.

Result

You added a monitor node, specified its monitoring settings and started monitoring.

Related topics

References

Add (CAN/LIN/Ethernet) Monitor..... 193
Add/Edit Monitor Dialog..... 201
Bus Navigator Page..... 207
Monitoring List..... 240
Set as Favorite Column Set (Monitoring List)..... 249
Start (Monitor)..... 251

How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN)

Objective

To specify a filter for monitoring and/or logging *selected* bus communication data.

Note

- The filters described in this topic refer to CAN and LIN bus communication. If you want to filter Ethernet bus communication, refer to Basics on Filtering Ethernet Bus Communication on page 91.
- You can also add a filter to a CAN replay node to reduce the data that is replayed. Refer to How to Replay Logged CAN or Ethernet Bus Communication on page 101.

Basics on filters

There are three types of filters:

- Range: Passes or blocks messages/frames with a message/frame ID within a specified range. The range is specified in the **Identifier** field of the filter dialog in the format <lower_limit>-<upper_limit>. The values of <lower_limit> and <upper_limit> can be specified in decimal or hexadecimal format. They must be separated by a hyphen, for example, 100-150 or 0x64-0x96. If you specify the values in decimal format they are automatically converted and displayed in hexadecimal format.
- ECU/Network node: Passes or blocks all messages/frames of the specified ECU or Network node.
- Message/Frame: Passes or blocks the specified message/frame.

Each filter can consist of one or more filter items for passing or blocking messages/frames. Filters of the same filter type (pass-pass or block-block) are combined by an implicit OR operator, filters of different filter type (pass-block) are combined by an implicit AND operator.

Preconditions

You must have an experiment with a bus configuration.

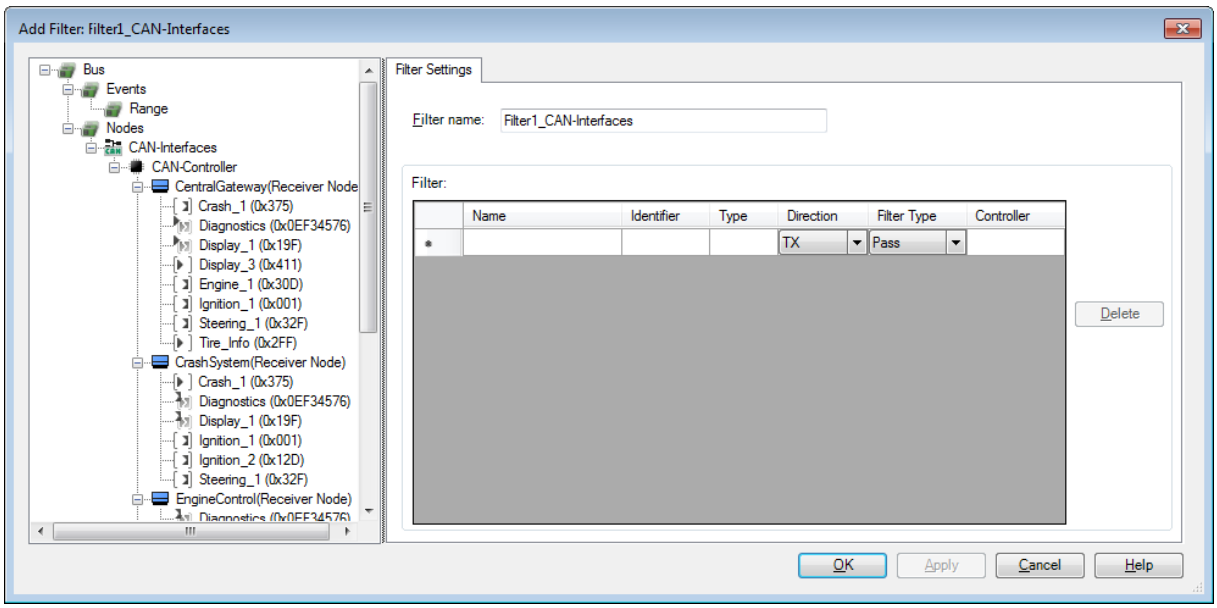
Method

To specify a filter for bus monitoring and logging

1 In the Bus Navigator tree, right-click a hardware node to open its context menu.

2 In the context menu, click Add Filter.

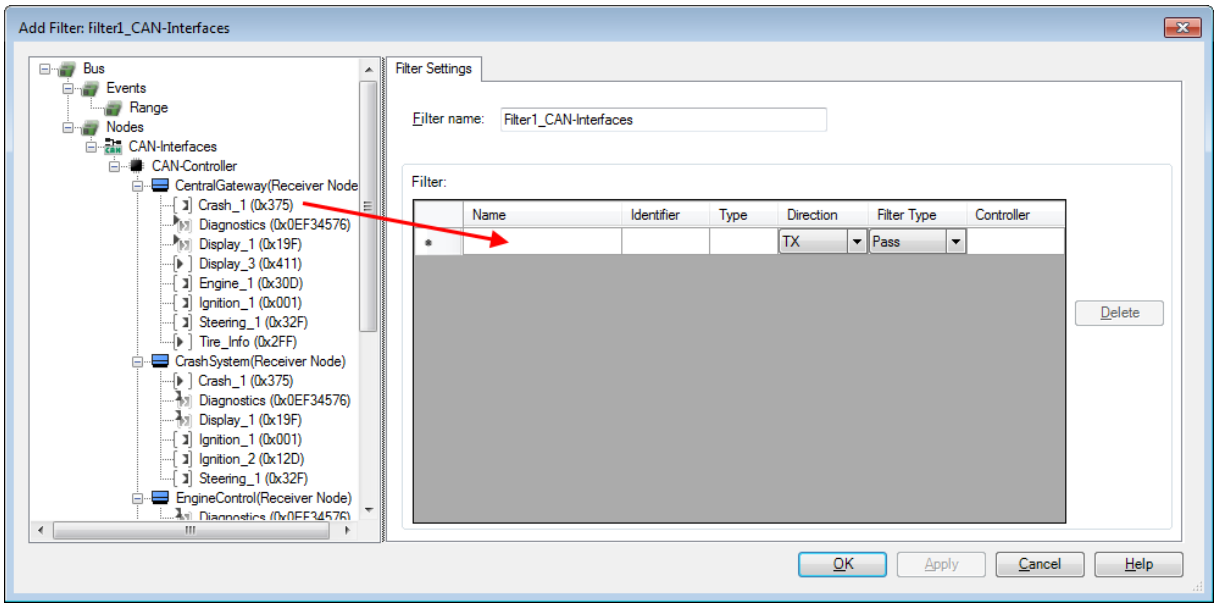
ControlDesk opens the Add Filter dialog.



The dialog shows the tree of a CAN bus on the left. It contains all the CAN messages grouped by ECUs and elements which you can use in your filter items. The Filter Settings page lets you specify the filter.

3 On the Filter Settings page, specify the filter name.

4 Specify a filter item by dragging, for example, a CAN message or ECU from the Bus Navigator tree to the **Filter** list and set the filter criteria.



5 Repeat the previous step for additional filter items.

For examples, refer to Example of Specifying Filters for Bus Monitoring or Logging (CAN, LIN) on page 88.

6 Click **OK** to save your settings and to close the dialog.

ControlDesk adds a filter node below the hardware node.

Result

You created a filter to monitor and/or log selected bus communication data.

Next steps

You can add a monitor node or logger node via the filter context menu to start filtered monitoring or logging.

Related topics

HowTos

How to Log Raw Bus Communication Data..... 95
How to Monitor Bus Communication..... 82

Examples

Example of Specifying Filters for Bus Monitoring or Logging (CAN, LIN)..... 88

References

Add (CAN/LIN/Ethernet) Filter..... 190
Add (CAN/LIN/Ethernet) Logger..... 191
Add (CAN/LIN/Ethernet) Monitor..... 193

Example of Specifying Filters for Bus Monitoring or Logging (CAN, LIN)

Introduction

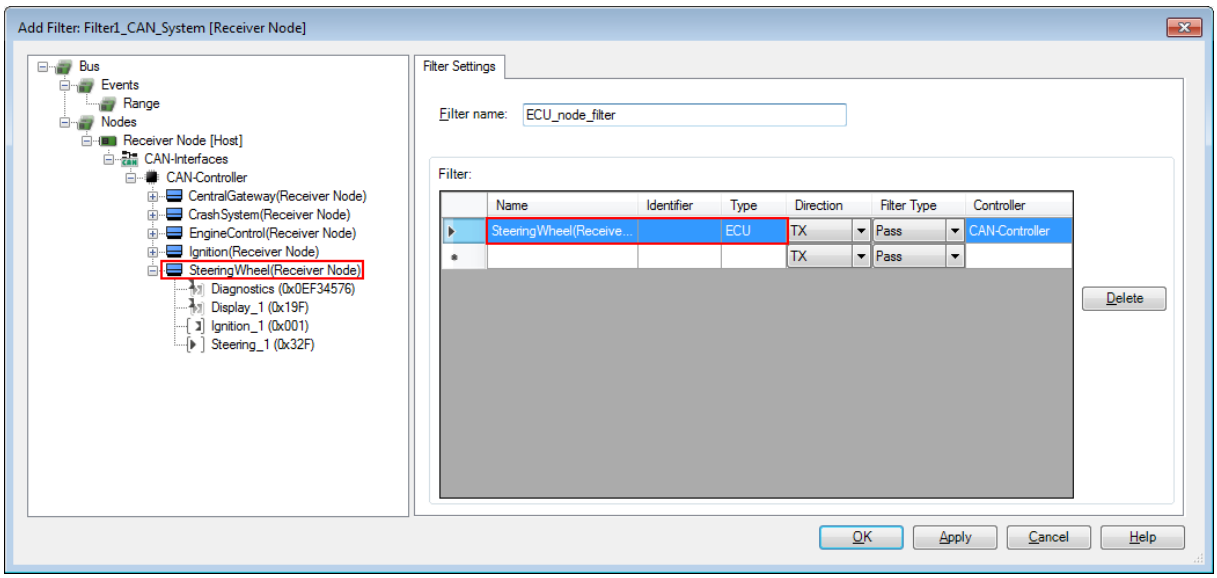
This topic shows some examples of specifying filters for bus monitoring or logging.

Tip

The examples in this topic show how to filter or log CAN messages. The handling is the same for communication data on a LIN bus.
An exception is Ethernet communication. For information on filtering Ethernet communication, refer to Basics on Filtering Ethernet Bus Communication on page 91.

Filtering all the CAN messages of an ECU node

To filter all the CAN messages of an ECU node, drag the ECU node from the tree and drop it to the filter list.



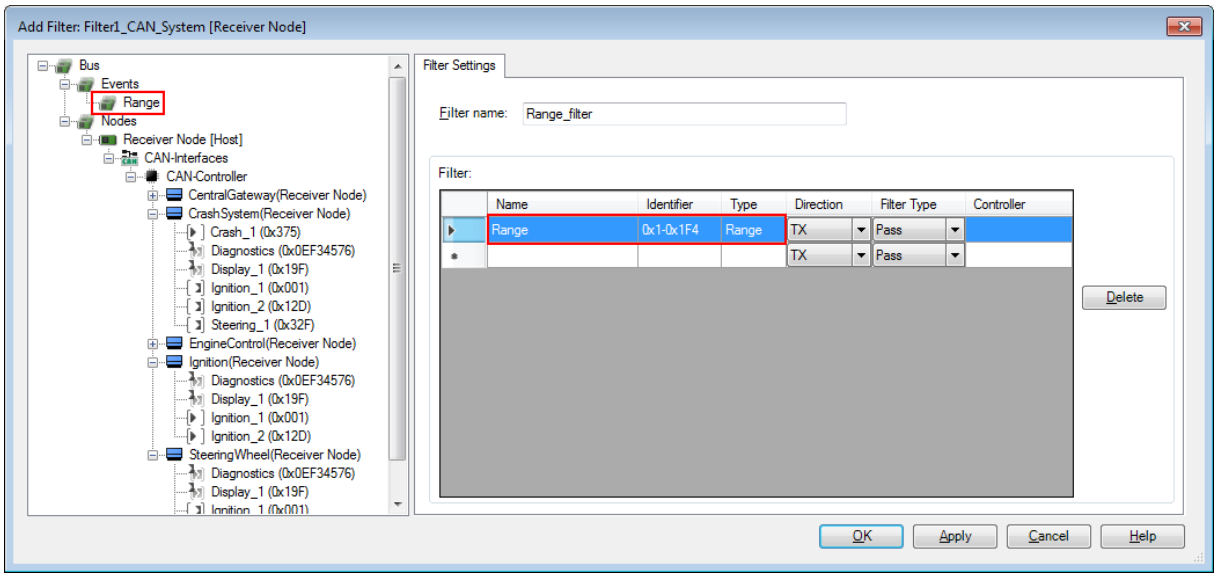
The following table describes all the settings for the filter item:

Filter Criteria	Description
Name	Name of the ECU node. It is set automatically when you drop the ECU node.
Identifier	It is empty because there is no special identifier.
Type	Type is ECU. It is set automatically when you drop the ECU node.
Direction	Direction of the messages. In this example, only TX messages are filtered. The relevant direction is the direction specified in the RTICANMM MainBlock and not the direction specified in the database file.

Filter Criteria	Description
Filter Type	In this example, the filter type is **Pass**. This means all the TX messages of the ECU node pass the filter.
Controller	Name of the controller for which the ECU node is configured. Only messages configured for the ECU node of this controller are filtered.

Filtering a range of message IDs

To filter CAN messages with an identifier within a specified range, drag the Range element from the tree and drop it to the filter list.

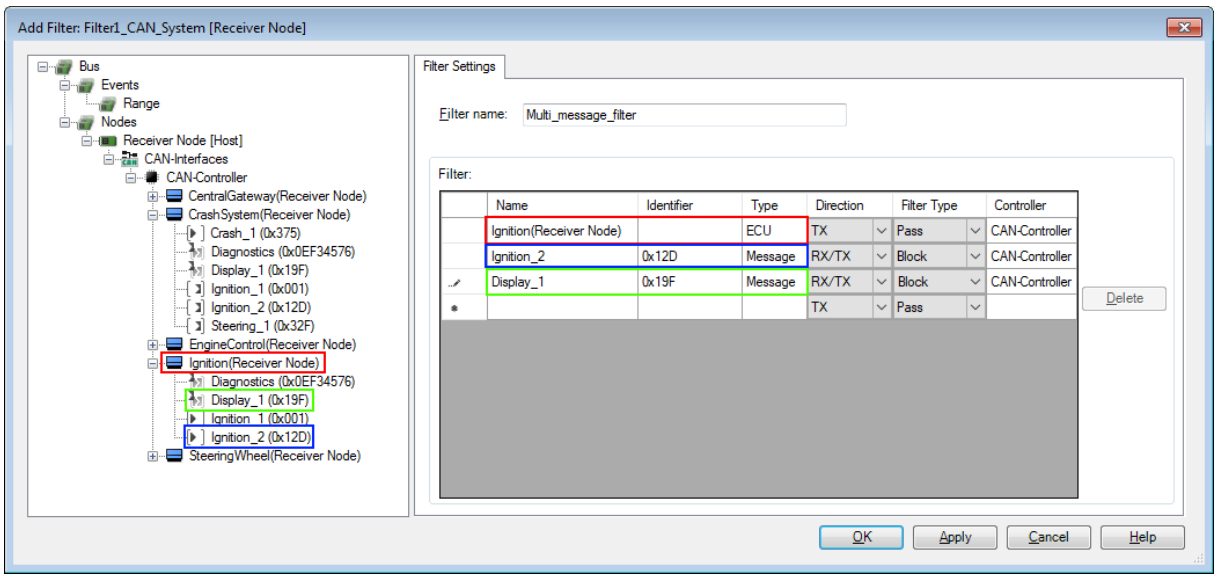


The following table describes all the settings for the filter item:

Filter Criteria	Description
Name	Name is **Range**. It is set automatically when you drop the **Range** node.
Identifier	In this field the lower and upper limit of the range must be specified. In the example, it is "1-500" to filter CAN messages with an identifier between 1 and 500.
Type	Type is **Range**. It is set automatically after dropping the **Range** element.
Direction	Direction of the messages. In this example, only TX messages are filtered. The relevant direction is the direction specified in the RTICANMM MainBlock and not the direction specified in the database file.
Filter Type	In this example, the filter type is **Pass**. This means, all TX messages of the node pass the filter.
Controller	Empty because the range is not specified for a specific controller.

Filtering several messages

This example shows how several CAN messages can be filtered. The first filter item lets all the TX messages of the Ignition ECU node pass. The second filter item blocks the Ignition_2 CAN message. The third filter item blocks the Display_1 CAN message. Only the rest of the CAN messages of the ECU node can pass. These are IGNITION_1 and DIAGNOSTICS.



The following tables describe all the settings for the filter items:

1st Filter Criteria	Description
Name	Name of the ECU node. It is set automatically when you drop the ECU node.
Identifier	It is empty because there is no special identifier.
Type	Type is ECU. It is set automatically when you drop the ECU node.
Direction	Direction of the messages. In this example, only TX messages are filtered. The relevant direction is the direction specified in the RTICANMM MainBlock and not the direction specified in the database file.
Filter Type	In this example, the filter type is Pass. This means, all TX messages of the ECU node pass the filter.
Controller	Name of the controller the ECU node is configured for. Only messages configured for the ECU node of this controller pass the filter.

2nd Filter Criteria	Description
Name	Name of the CAN message. It is set automatically when you drop the message node.
Identifier	Displays the value of the message identifier.

2nd Filter Criteria	Description
Type	Type is **Message**. It is set automatically when you drop the message node.
Direction	Direction of the messages. In this example, only the TX message is blocked (RX messages are already blocked by the 1st filter item). The relevant direction is the direction specified in the RTICANMM MainBlock and not the direction specified in the database file.
Filter Type	In this example, the filter type is **Block**. This means that the message does not pass the filter.
Controller	Name of the controller the message is configured for. Only the message configured for this controller is blocked.

3rd Filter Criteria	Description
Name	Name of the CAN message. It is set automatically after dropping the message node.
Identifier	Displays the value of the message identifier.
Type	Type is **Message**. It is set automatically after dropping the message node.
Direction	Direction of the messages. In this example, only the TX message is blocked (RX messages are already blocked by the 1st filter item). The relevant direction is the direction specified in the RTICANMM MainBlock and not the direction specified in the database file.
Filter Type	In this example, the filter type is **Block**. This means that the message does not pass the filter.
Controller	Name of the controller the message is configured for. Only the message configured for this controller is blocked.

Related topics

HowTos

How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN)..... 85

Basics on Filtering Ethernet Bus Communication

Introduction

You can reduce the amount of data captured by the Ethernet Bus Monitoring device by specifying a filter for Ethernet bus communication.

Filter node

To filter the Ethernet bus communication you must add an Ethernet filter to one of the following nodes in the Bus Navigator tree:

- System node
- Hardware node
- Controller node

The filter reduces the amount of data captured by the Ethernet Bus Monitoring device. You can add various Ethernet filter nodes, but only one can be active at a time.

The Ethernet filter is configured using the Berkeley Packet Filter (BPF) syntax.

Berkeley Packet Filter (BPF)

The Berkeley Packet Filter (BPF) provides a protocol-independent interface to data link layers. A data link layer is a protocol layer that transfers data between network nodes. BPF supports the filtering of the transferred data.

For more information on BPF, refer to https://en.wikipedia.org/wiki/Berkeley_Packet_Filter.

In the Bus Navigator, the Ethernet filter expressions that specify the Ethernet packets to be captured are defined according to BPF syntax.

Predefined filters and keywords

You can use predefined filters or keywords to configure the settings of an Ethernet filter node.

Predefined filters

Broadcast or Multicast
Ethernet type 0x0806 (ARP)
HTTP TCP port (80)
ICMP only
IPv4 only
IPv6 only
No ARP
No ARP and no DNS
No Broadcast and no Multicast
TCP only
TCP or UDP port 80 (HTTP)
UDP only

Keywords

aarp	connectmsg	hdpc	inbound	llc	oam	ra	stp
action	csnp	hfisu	ip	lsp	oamf4	radio	subtype
address1	decnet	hlssu	ip6	lssu	oamf4ec	rarp	ta
address2	direction	hmsu	ipx	lsu	oamf4sc	reason	tcp
address3	dpc	hopc	isis	mask	on	rnr	type
address4	dst	host	is-is	metac	opc	rset	udp
ah	esis	hsio	iso	metaconnect	outbound	sc	vci
arp	es-is	hsls	l1	mopdl	pim	sca	vlan
atalk	esp	icmp	l2	moprc	port	sctp	vpi
bcc	ether	icmp6	lane	mpls	portrange	sio	vrrp
broadcast	fddi	igmp	lat	msu	pppoed	sls	
byte	fisu	igrp	len	multicast	pppoes	snp	
carp	gateway	iih	less	net	proto	src	
clnp	greater	ilmic	link	netbeui	psnp	srnr	

User-defined filters

You can also create user-defined filters and add them to the list of filters.

User defined filters

My_Filter_1
My_Filter_2

Predefined filters

Keywords

You can export and import user-defined filters. Refer to Add/Edit Filter Dialog (Ethernet) on page 197.

Ethernet filter examples

For Ethernet filter examples, refer to Examples of Filtering Ethernet Bus Communication on page 94.

Configuring Ethernet filter settings

You can configure the settings of an Ethernet filter when you add or edit an Ethernet filter node in the Bus Navigator tree. Refer to Add/Edit Filter Dialog (Ethernet) on page 197.

Related topics

Examples

Examples of Filtering Ethernet Bus Communication..... 94

References

Add/Edit Filter Dialog (Ethernet)..... 197

Examples of Filtering Ethernet Bus Communication

Introduction

Some examples show you how to define Ethernet filter expressions according to the Berkeley Packet Filter syntax.

Ethernet filter examples

The table below shows filter examples for filtering Ethernet bus communication.

Example	Description
dst host 10.61.77.251	Filters for packets that are sent to the specified IP address.
src host 10.61.77.251	Filters for packets that are sent from the specified IP address.
ether dst host 00:00:5E:00:53:00	Filters for Ethernet packets that are sent to the specified MAC address.
ether src host 00:00:5E:00:53:00	Filters for Ethernet packets that are sent from the specified MAC address.
dst FF02::1:3	Filters for packets that are sent to the specified IPv6 address.
src FF02::1:3	Filters for packets that are sent from the specified IPv6 address.
ether[0:2] = 0xFFFF	Filters for packets that contain specific Ethernet frame byte values using an [offset:size]-notation. Description of the example: Go to the first byte of the Ethernet frame and compare it and the following byte with the specified value. Size must be either one (default), two or four.
ip	Filters for IPv4 packets.
ip6	Filters for IPv6 packets.
vlan 100	Filters for packets that belong to the specified VLAN.
tcp[tcpflags] & (tcp-ack) !=0	Filters by TCP flags. Syntax: `[ip|ip6] tcp[tcpflags] & (tcp-[ack|fin|syn|rst|push|urg)`
len = 60	Filters for packets of the specified length.
len > 60	Filters for packets that exceed the specified length.
proto UDP	Filters for packets that belong to the specified protocol, in this case UDP.

Example	Description
proto not UDP	Filters for packets that match all protocols except for UDP.
dst port 111	Filters packets that are sent to the specified port.
src port 111	Filters for packets that are sent from the specified port.
port 111	Filters for packets that are either sent to or from the specified port.
tcp dst portrange 80-90	Filters for TCP packets addressed to a port in the specified port range.
ip and proto UDP	You can combine filters. The example filters for IPv4 packets that also match the UDP protocol.

Related topics

Basics

Basics on Filtering Ethernet Bus Communication.....91

References

Add/Edit Filter Dialog (Ethernet).....197

How to Log Raw Bus Communication Data

Objective

To log raw bus communication data.

The instructions below show you how to perform *filtered* logging.

Preconditions

- The experiment must contain a bus configuration.
- You created at least one filter. Refer to How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN) on page 85.

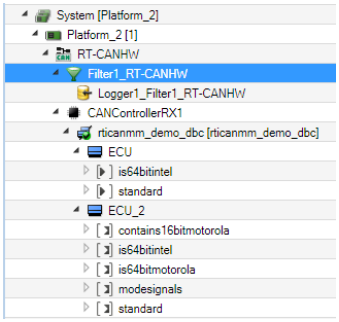
Method

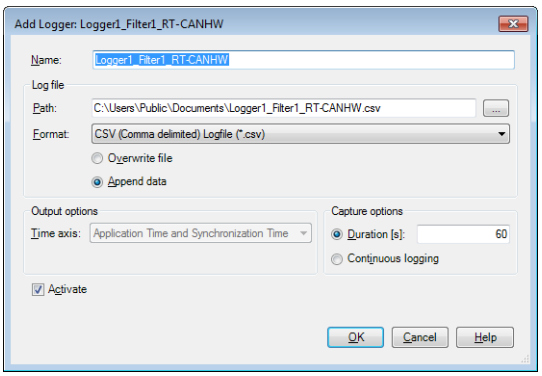
To log raw bus communication data

1 In the Bus Navigator tree, right-click a filter node to open its context menu.

2 In the context menu, click **Add Logger**.

ControlDesk opens the **Add Logger** dialog and a logger node is added below the selected filter node.





3 In the dialog, specify the name of the logger and specify a new log file or select an existing CSV, ASC, or PCAPNG log file to write the data to.

4 For ASC files, select the time axis:

- **Application time**: The time stamps relate to the start of the application.
- **Synchronization time**: The time stamps relate to the start of the PC.

5 Specify the duration of data logging:

- Specify a **Duration** value if you want to log data for a fixed period of time.

  Or:

- Select **Continuous logging** if you want to log data continuously and not for a fixed duration.

6 Confirm your settings with **OK**.

ControlDesk closes the dialog.

7 On the **Home** ribbon, click **Status Control – Go Online** to start online calibration.

If you are using a bus monitoring device, the connected state is sufficient, you do not have to start online calibration.

The logger node is activated.

8 Right-click the logger node and select **Start Logger** from its context menu.

ControlDesk starts logging the raw data of the filtered bus communication. If you specified a fixed time period for data logging, logging stops automatically after this time period.

9 Open the log file to which you saved the logged data to check the logging results.

The log file provides information on the logged messages in raw data format, such as the controller ID, message ID, delta time, message direction, etc.

Result

You specified settings for logging the raw data of filtered messages/frames/PDUs and performed the logging.

Related topics

References

Add (CAN/LIN/Ethernet) Filter..... 190
Add (CAN/LIN/Ethernet) Logger..... 191

How to Display Bus Statistics

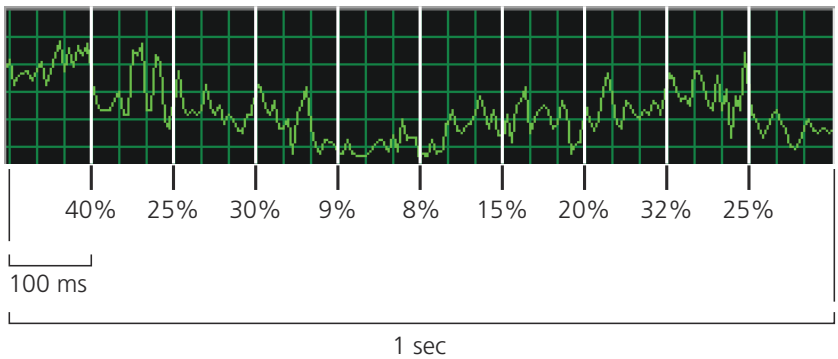
Introduction

To enable the display of bus statistics during bus monitoring.

Basics on bus statistics

Beside frame count values and the time for which the application has been running, the **Bus Statistics** display information on the bus load.

ControlDesk measures the bus load every 100 ms. Based on these measurements, the **Bus Statistics** display the highest measured bus load (peak) and the mean bus load value (average) once per second. Refer to the following illustration, which shows an exemplary time plot of the bus load.



Tip

You can use a logger to save the bus statistics together with the logged data. Refer to Add (CAN/LIN/Ethernet) Logger on page 191.

Preconditions

- You must use a bus monitoring device in connection with a CAN or LIN interface of a SCALEXIO system, MicroAutoBox III, or VEOS.
- You must have added a monitor to the controller node of the bus monitoring device.

Method

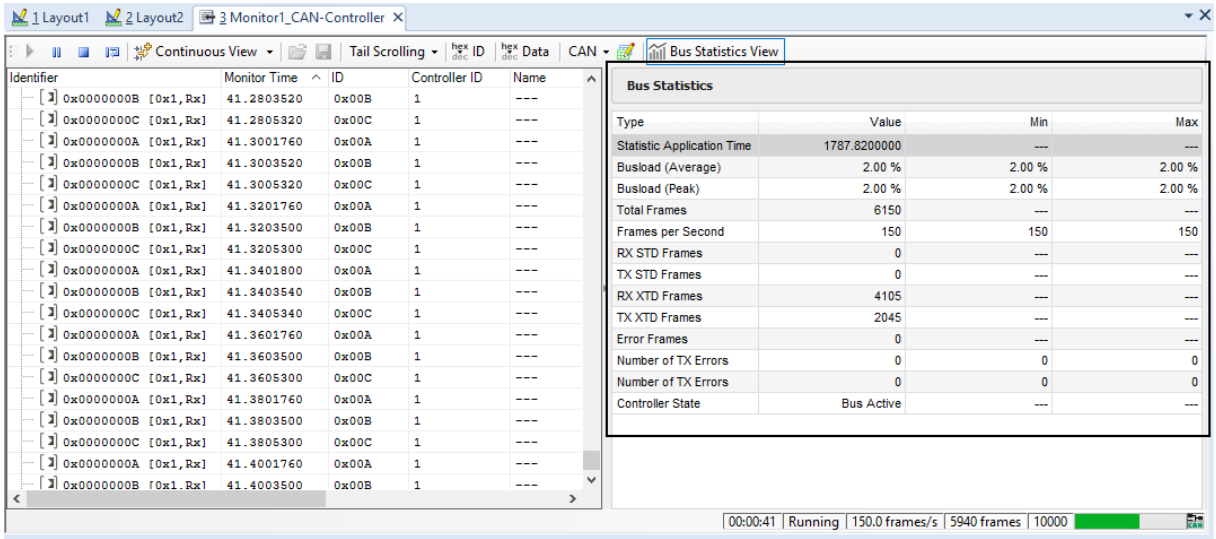
To display bus statistics

1 Open the monitoring list of the controller of a CAN or LIN Bus Monitoring device and start the monitoring.

2 Click Bus Statistics View to display the Bus Statistics values.

Result

The Bus Statistics are displayed in the monitoring list.



Related topics

References

Add (CAN/LIN/Ethernet) Monitor..... 193
Monitoring List..... 240

Replaying CAN and Ethernet Bus Communication

Where to go from here

Information in this section

Basics on Replaying CAN and Ethernet Bus Communication..... 99
The Bus Navigator lets you replay the raw data of logged CAN and Ethernet bus communication. You can also monitor the data that is being replayed.

How to Replay Logged CAN or Ethernet Bus Communication..... 101
To replay the raw data of logged CAN messages and Ethernet packets.

Basics on Replaying CAN and Ethernet Bus Communication

Introduction

The Bus Navigator lets you replay the raw data of logged CAN and Ethernet bus communication. You can also monitor the data that is being replayed.

Note

You are responsible for ensuring that ControlDesk's replay function is used only in accordance with the intended purpose and the applicable legal regulations. Refer to Important Information on Replaying Logged Ethernet Raw Data (ControlDesk Introduction and Overview 🕮).

Supporting interfaces

The following table specifies the interfaces that support CAN and Ethernet replay.

Interface Type	Interfaces Supporting CAN Replay	Interfaces Supporting Ethernet Replay
Bus channels of a dSPACE real-time platform[1)]	▪ DS100x-based systems ▪ MicroLabBox ▪ MicroAutoBox II ▪ SCALEXIO	–
Monitoring device in connection with a PC-based interface	▪ dSPACE DCI-CAN2 or DCI-CAN/LIN1[2)] ▪ Various interfaces of Vector Informatik, such as VN16xx, VN5610, and VN8900[2)] For more information, refer to Supported CAN Interfaces (ControlDesk Platform Management 🕮).	▪ Vector Informatik interfaces VN5610 and VN5610A ▪ Loopback adapter of the host PC ▪ Network interface cards of the host PC For more information, refer to Supported Ethernet Interfaces

Interface Type	Interfaces Supporting CAN Replay	Interfaces Supporting Ethernet Replay
		(ControlDesk Platform Management 🕮).
Monitoring device in connection with bus channels of a dSPACE platform	▪ SCALEXIO[3)] ▪ MicroAutoBox III[3)] ▪ VEOS[4)]	▪ SCALEXIO[3)] ▪ MicroAutoBox III[3)] ▪ VEOS[4)]

1) Only for RTICANMM-based bus communication. Requires Real-Time Testing (RTT) of version 1.7.1 or later on the host PC. Refer to Compatibility with Real-Time Testing (ControlDesk Introduction and Overview 🕮).
2) These interfaces also support CAN FD.
3) Transmission timing of messages is not exact, because it is controlled by a clock source on the host PC.
4) If you use real-time acceleration factors other than **1**, the timing of the replayed messages is not correct.

Supported log file formats

You can replay log files of the following formats:

Bus type	File Format	Description
CAN	CSV	Comma-separated files generated with ControlDesk.
	ASC	File format used for log files created with CANalyzer (Vector Informatik GmbH).[1)]
Ethernet	PCAPNG	File format widely used for storing captured packet traces. A PCAPNG file contains several blocks that contain different types of information. The blocks can be used to rebuild captured packets.

1) The ASC format that was introduced with CANalyzer 7.5 is supported only if the *absolute time format* (message format) is enabled and *logging values of variables symbolic* is disabled for logging.

Monitoring and filtering

The Add/Edit Replay dialog lets you configure the Bus Navigator to start monitoring automatically when the replay starts. This lets you monitor the replayed communication data.

For CAN, you can also specify a filter to replay only selected bus communication data.

Related topics

Basics

Compatibility with Real-Time Testing (ControlDesk Introduction and Overview 🕮)
Limitations for the Bus Navigator 267

HowTos

How to Log Raw Bus Communication Data 95
How to Monitor Bus Communication 82
How to Replay Logged CAN or Ethernet Bus Communication 101
How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN) 85

How to Replay Logged CAN or Ethernet Bus Communication

Objective

To replay the raw data of logged CAN messages and Ethernet packets.

Preconditions

For CAN replay with a dSPACE platform that requires RTT, the following preconditions must be fulfilled:

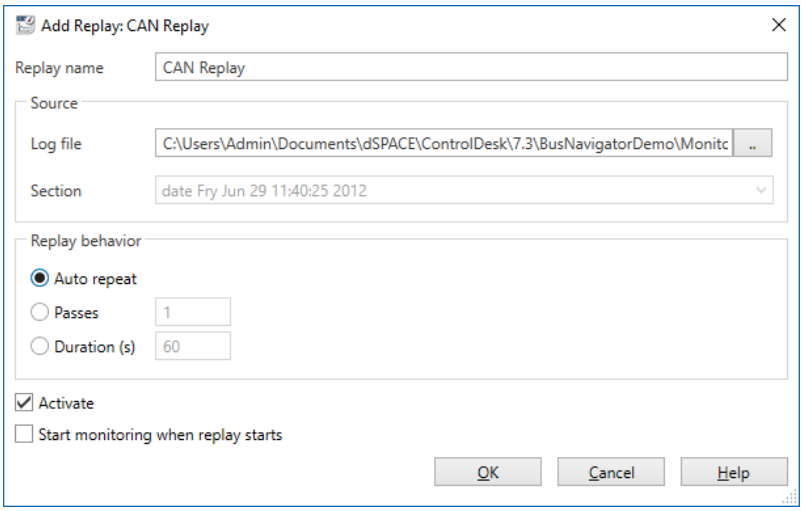
- Real-Time Testing must be installed and activated on the host PC in the version required by the simulation platform. Refer to Basics on Replaying CAN and Ethernet Bus Communication on page 99.
- Real-Time Testing must be enabled in the real-time application for which you want to replay logged communication data. Refer to Enabling Real-Time Testing for dSPACE Platforms (Real-Time Testing Guide 🕮).

Method

To replay logged bus communication

1. In the Bus Navigator tree, right-click a CAN or Ethernet controller to open its context menu.
2. In the context menu, click Add Replay.

   ControlDesk adds a replay node below the selected controller node and opens the Add Replay dialog.



3. Select the log file you want to replay, and configure the other dialog settings according to your needs.

   Select Activate to directly enable the replay. This automatically deactivates any other replay nodes of the selected controller. However, you can also activate the replay at a later time via its context menu in the Bus Navigator tree. The replay node must be active before you can start the replay.

   For RTICANMM-based bus communication, the activation can take some time since a MAT file must be generated for each replay.

4 If you want to monitor the replayed communication data, select **Start monitoring when replay starts**.

5 Click **OK** to close the dialog.

6 From the context menu of the added replay node, select **Start**.

Result

ControlDesk starts replaying the log file you specified.

If you selected **Start monitoring when replay starts**, monitoring starts automatically just before the replay starts. You have to stop the monitoring manually after you stopped the replay.

Next steps

For CAN bus communication, you can add a filter to a replay node via its context menu. When you do this, the **Add Filter** dialog opens for you to specify the filter settings and a new filter node is displayed in the tree above the replay node.

Refer to the following illustration.

Related topics

Basics

Basics on Replaying CAN and Ethernet Bus Communication..... 99
Compatibility with Real-Time Testing (ControlDesk Introduction and Overview 🕮)

HowTos

How to Log Raw Bus Communication Data..... 95
How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN)..... 85

References

Add (CAN/Ethernet) Replay..... 189
Add (CAN/LIN/Ethernet) Filter..... 190
Start (Replay)..... 252

Reference Information

Where to go from here

Information in this section

Bus Instruments..... 104
The Bus Navigator provides various Bus Instrument types which are accessible via context menus of the Bus Navigator tree.

Bus Navigator Commands and Dialogs..... 182
The ControlDesk Bus Navigator provides various commands and dialogs.

Bus Instruments

Introduction

The Bus Navigator provides various Bus Instrument types which are accessible via context menus of the Bus Navigator tree.

The available bus instruments and the information and configurable settings they display depend on the setup you use to access the bus communication:

- For a bus configuration related to a simulation application on a platform, advanced inspection and manipulation features can be displayed, depending on the model basis, i.e., a blockset or the Bus Manager.
- For a bus configuration related to a bus monitoring device, only basic information on the signals contained in a PDU can be displayed.

Where to go from here

Information in this section

Bus Instruments (Bus Manager)..... 104

Bus Instruments (Blockset)..... 139

Bus Instruments (Related to Bus Monitoring Devices)..... 174

Bus Instruments (Bus Manager)

Where to go from here

Information in this section

Global Instruments..... 105

CAN Instruments..... 106

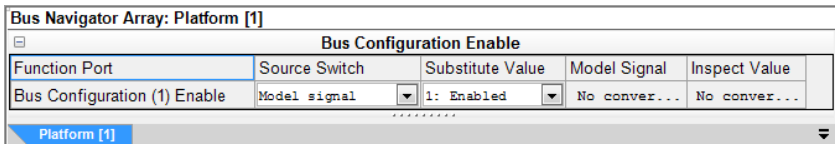
LIN Instruments..... 125

Global Instruments

Bus Instrument (Global Layout Type)

Illustration

The following illustration shows an example of the instrument:



Purpose

To enable and disable the bus communication of a bus configuration fully or partly in one step.

Description

The Bus Instrument (Global layout type) lets you enable or disable the bus communication of a bus configuration fully or partly.

The Bus Instrument (Global layout type) displays only the bus configuration settings which are enabled to be displayed and/or manipulated by ControlDesk during run time.

For information on the related configuration elements in the Bus Manager, refer to:

- Bus Manager in ConfigurationDesk: Enabling and Disabling Bus Configurations (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Enabling and Disabling Bus Configurations (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Instrument elements

The following regions can be available in the instrument:

Region	Description
Bus Configuration Enable	Displays the available bus configurations and lets you specify the source switch settings as well as the substitute values. To enable or disable a bus configuration via the substitute value, set Source Switch to Substitute value. There are three possible settings for the substitute value: ▪ 0: Disabled: The bus configuration is fully disabled. ▪ 1: Enabled: The bus configuration is fully enabled. ▪ 2: RX enabled only: The bus configuration is partly enabled, i.e., data can be received but not transmitted via the bus.

The instrument regions can contain the following elements:

Function Port Displays function ports configured via the Bus Manager.

Source Switch Lets you switch between the behavior model (**Model signal**) and the substitute value as the source of the signal value.

When test automation support is enabled for a signal, you can specify the source of the signal value. You can use the static substitute value and adjust it in your experiment or exchange dynamic values with a behavior model via the model signal. Refer to Configuring Test Automation Support (ConfigurationDesk I/O Function Implementation Guide 🕮).

Substitute Value Lets you specify the substitute value. To transmit the substitute value instead of the value of the model signal, set **Source Switch** to **Substitute value**.

Model Signal Displays the model signal value. It is calculated using the control algorithms of the behavior model. To transmit the dynamic values of the behavior model on the bus, set **Source Switch** to **Model signal**.

Inspect Value Displays the current value that is transmitted on the bus.

Related topics

References

Generate Layout..... 231

CAN Instruments

Where to go from here

Information in this section

Bus Instrument (Inspection Type for CAN)..... 107
To display information on a PDU that has been configured for inspection purposes.

Bus Instrument (Manipulation Type for CAN)..... 110
To change the settings of a PDU that has been configured for manipulation purposes.

Bus Instrument (RX Type for CAN)..... 114
To display information on an RX message.

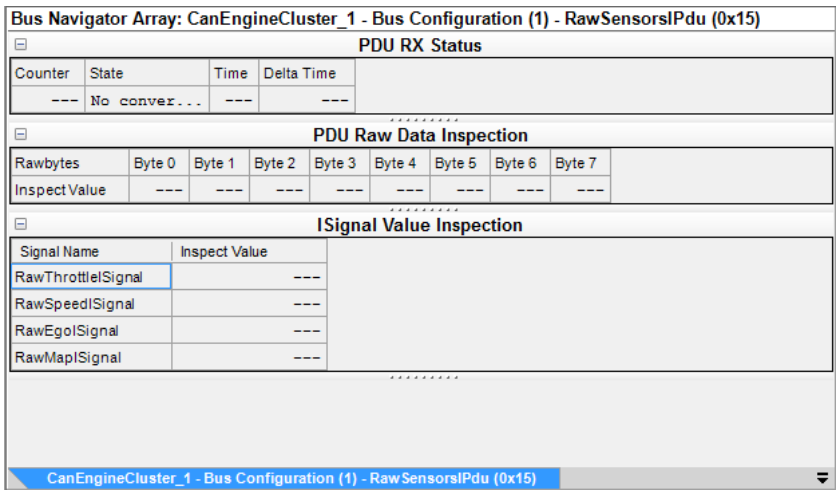
Bus Instrument (TX Status Type for CAN)..... 119
To specify whether TX messages/PDUs of a communication cluster are transmitted on the bus and to specify basic transmission options.

Bus Instrument (TX Type for CAN)..... 121
To display, configure, and transmit a TX message.

Bus Instrument (Inspection Type for CAN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To display information on a PDU that has been configured for inspection purposes.

Description

You can create one or more Bus Instruments (Inspection type for CAN) for each RX and TX/RX PDU that has been configured for inspection purposes. The instrument opens in a separate layout when you create it via the **Generate Inspection Layout** command.

Instrument elements

The Bus Instrument (Inspection type for CAN) displays information on received PDUs (RX PDUs) that have been configured for inspection purposes.

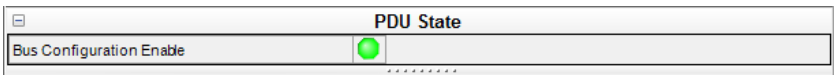
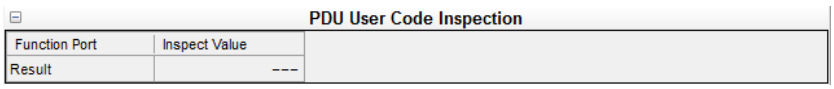
The instrument content is tailor-made, i.e., the displayed settings depend on the settings specified in the Bus Manager.

For information on the related configuration elements in the Bus Manager, refer to:

- Bus Manager in ConfigurationDesk: Working with Bus Configuration Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Working with Bus Configuration Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

The following regions can be available in the instrument:

Region	Description
ISignal Value Inspection	**ISignal Value Inspection** Signal Name / Inspect Value RawThrottleISignal --- RawSpeedISignal --- RawEgoISignal --- RawMapISignal --- Displays all the ISignal values of a PDU that are transmitted on the bus.
ISignalGroup EndToEnd Protection Status Inspection	**ISignalGroup EndToEnd Protection Status Inspection** Signal Group / / / SignalGroup_E2E_BE_P01 / / Inspect Value / Calculated CRC / --- / E2E State / No conver... Displays information on the status of a received end-to-end protected signal group. The following values are displayed: ▪ Calculated CRC: The checksum value that is calculated for the received signal group. ▪ E2E State: The state of the end-to-end protection, i.e., whether the received end-to-end protection information indicates a failure. For more information on possible function port values, refer to: ▪ Bus Manager in ConfigurationDesk: Observing the Status of Received End-to-End-Protected ISignal Groups (ConfigurationDesk Bus Manager Implementation Guide 🕮) ▪ Bus Manager (stand-alone): Observing the Status of Received End-to-End-Protected ISignal Groups (Bus Manager (Stand-Alone) Implementation Guide 🕮)
PDU Raw Data Inspection	**PDU Raw Data Inspection** Rawbytes / Byte 0 / Byte 1 / Byte 2 / Byte 3 / Byte 4 / Byte 5 / Byte 6 / Byte 7 Inspect Value / --- / --- / --- / --- / --- / --- / --- / --- Displays the raw data values for each byte of a PDU.
PDU RX Status	**PDU RX Status** Counter / State / Time / Delta Time --- / No conver... / --- / --- Displays status information of the PDU.
PDU SecOC Inspection	**PDU SecOC Inspection** Function Port / Source Switch / Substitute Value / Model Signal / Inspect Value Authenticator Value [0] / / / / --- Authenticator Value [1] / / / / --- Authenticator Value [2] / / / / --- Freshness Value / / / / --- Calculated Freshness Value / / / / --- State / / / / --- Enable Verification / Model signal / 0 / --- / --- Displays authentication information of a secured PDU and lets you enable and disable verification of the authentication information. The following data is displayed: ▪ Authenticator Value[0..n]: The value of each element of the authenticator value vector that is received on the bus. ▪ Freshness Value: The freshness value that is received on the bus. ▪ Calculated Freshness Value: The freshness value that is calculated via user code. ▪ State: The state of the verification, for example, whether the authentication information is valid.

Region	Description
	▪ Enable Verification: You can enable or disable verification of the authentication information by setting **Source Switch** to **Model Signal** and setting **Substitute Value** to `1` or `0`, respectively. For more information on possible function port values, refer to: ▪ Bus Manager in ConfigurationDesk: Verifying the Authentication Information of Received Secured IPDUs (ConfigurationDesk Bus Manager Implementation Guide 🕮) ▪ Bus Manager (stand-alone): Verifying the Authentication Information of Received Secured IPDUs (Bus Manager (Stand-Alone) Implementation Guide 🕮)
PDU State	 Displays the status of the entire bus configuration. Green means that communication is enabled, red means that it is disabled.
PDU User Code Inspection	 Displays the value that the user code execution returns.

The instrument regions can contain the following elements:

Counter Displays the value of the counter. The counter value is incremented by 1 each time the PDU is received via the bus. If an overflow occurs, the counter value is reset to 0.

Delta Time Displays the delta time (in seconds) between the current and the previous reception of the PDU. Since there is no actual delta time when the PDU is received for the first time, the function port provides the time of the simulation application in this case. When the PDU is received for the second time, the function port provides the actual delta time.

Function Port Displays function ports configured via the Bus Manager.

Inspect Value Displays the current value that is transmitted on the bus.

Rawbytes – Byte <number> Displays the raw data value for a specific byte of a PDU.

Signal Name Displays the name of the signal.

State Displays the state of the PDU, i.e., if the PDU is received in the current sampling step:

- 0: The PDU is not received in the current sampling step.
- 1: The PDU is received in the current sampling step.

Time Displays the time (in seconds) of the simulation application when the PDU was received.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

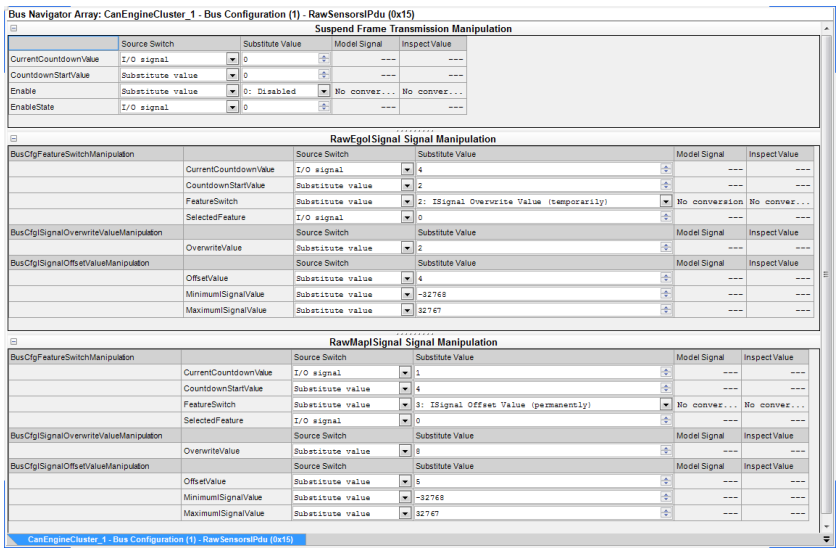
References

Generate Inspection Layout..... 230

Bus Instrument (Manipulation Type for CAN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To change the settings of a PDU that has been configured for manipulation purposes.

Description

You can create one or more Bus Instruments (Manipulation type for CAN) for each PDU that has been configured for manipulation purposes. The instrument opens in a separate layout when you create it via the **Generate Manipulation Layout** command.

Instrument elements

The Bus Instrument (Manipulation type for CAN) displays information on PDUs that have been configured for manipulation purposes.

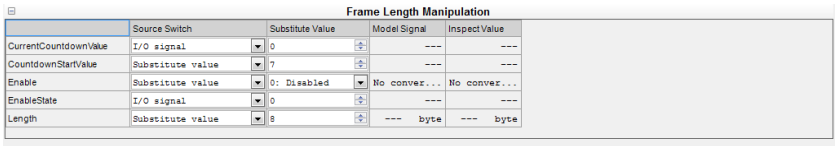
The instrument displays instrument elements and settings depending on the settings specified in the Bus Manager.

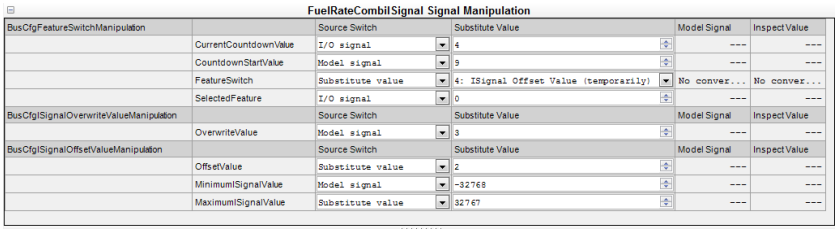
For information on the related configuration elements in the Bus Manager, refer to:

- Bus Manager in ConfigurationDesk: Working with Bus Configuration Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Working with Bus Configuration Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

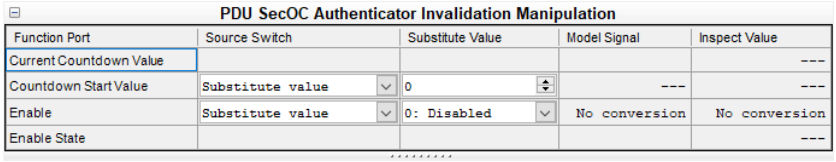
The following regions can be available in the instrument:

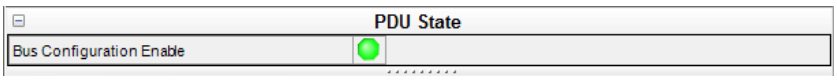
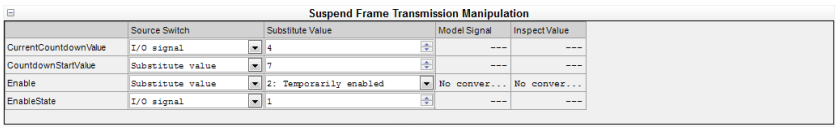
Region	Description

Frame Length Manipulation	
	Frame Length Manipulation
	Lets you manipulate the payload length of a CAN frame.
	The following functions are available:
	▪ **CurrentCountdownValue**: Lets you specify the current countdown value. This value is evaluated only if frame length manipulation is temporarily enabled.
	▪ **CountdownStartValue**: Lets you specify the countdown start value. If frame length manipulation is temporarily enabled, this value determines how often the frame length is manipulated.
	▪ **Enable**: Lets you enable frame length manipulation. The following settings are available:
	  ▪ 0: Disabled
	  ▪ 1: Permanently enabled
	  ▪ 2: Temporarily enabled
	▪ **EnableState**: Lets you specify the number that is provided as the number of the active setting of the **Enable** feature. This allows you to transmit manipulated information to the behavior model, for example.
	▪ **Length**: Lets you specify the payload length of the CAN frame. This value is evaluated only if frame length manipulation is enabled.
	  In the following cases, the specified length is automatically adjusted at run time:
	  ▪ For classic CAN frames, the maximum length is limited to 8 bytes.
	  ▪ A payload length that is invalid according to CAN FD is extended by padding bytes to the next higher valid length. The value of each padding byte is 255.

<ISignal Name> Signal Manipulation	 **FuelRateCombiSignal Signal Manipulation** Displays the available signal manipulation features and lets you select the one you want to use for the manipulation of the bus communication.

Region	Description
	The following functions are available to select and configure the manipulation: ▪ CurrentCountdownValue: Lets you specify the current countdown value. This value is evaluated only if the bus manipulation feature is temporarily enabled. ▪ CountdownStartValue: Lets you specify the countdown start value. If the bus manipulation feature is temporarily enabled, this value determines how often the ISignal is manipulated. ▪ FeatureSwitch: Displays the available manipulation features and lets you select one. The selection also specifies whether the manipulation applies permanently or temporary. If you select a temporarily manipulation, the CountdownStartValue and the CurrentCountdownValue are used to control the manipulation. ▪ SelectedFeature: Lets you specify the number that is provided as the number of the active manipulation feature. This allows you to transmit manipulated information to the behavior model, for example. The following functions are available for the OverwriteValueManipulation feature: ▪ Overwrite value: Lets you specify the value of the ISignal. The following functions are available for the SignalOffsetValueManipulation feature: ▪ Offset value: Lets you apply an offset value to the current value of the ISignal. ▪ MinimumISignalValue: Lets you specify a minimum value for the ISignal. ▪ MaximumISignalValue: Lets you specify a maximum value for the ISignal.

PDU SecOC Authenticator Invalidation Manipulation	
	PDU SecOC Authenticator Invalidation Manipulation
	Function Port / Source Switch / Substitute Value / Model Signal / Inspect Value
	Current Countdown Value / --- / ---
	Countdown Start Value / Substitute value / 0 / --- / ---
	Enable / Substitute value / 0: Disabled / No conversion / No conversion
	Enable State / --- /
	Displays authentication information on a secured PDU and lets you invalidate the authentication information.
	The following functions are available:
	▪ CurrentCountdownValue: Lets you specify the current countdown value.
	▪ CountdownStartValue: Lets you specify the countdown start value.
	▪ Enable: Lets you enable authenticator invalidation. The following settings are available:
	  ▪ 0: Disabled
	  ▪ 1: Permanently enabled
	  ▪ 2: Temporarily enabled
	▪ EnableState: Lets you specify the number that is provided as the number of the active setting of the Enable feature. This allows you to transmit manipulated information to the behavior model, for example.

PDU State	 PDU State Bus Configuration Enable Displays the status of the entire bus configuration. Green means that communication is enabled, red means that it is disabled.
Suspend Frame Transmission Manipulation	 Suspend Frame Transmission Manipulation Source Switch / Substitute Value / Model Signal / Inspect Value CurrentCountdownValue / I/O signal / 4 / --- / --- CountdownStartValue / Substitute value / 7 / --- / --- Enable / Substitute value / 2: Temporarily enabled / No conver... / No conver... EnableState / I/O signal / 1 / --- / --- Lets you suspend the transmission of the frame the PDU is included in. The following functions are available: ▪ CurrentCountdownValue: Lets you specify the current countdown value. This value is evaluated only if the frame transmission suspension is temporarily enabled. ▪ CountdownStartValue: Lets you specify the countdown start value. If the frame transmission suspension is temporarily enabled, this value determines how often the frame transmission is suspended.

Region	Description
	▪ Enable: Lets you enable the frame transmission. The following settings are available: ▪ 0: Disabled ▪ 1: Permanently enabled ▪ 2: Temporarily enabled ▪ EnableState: Lets you specify the number that is provided as the number of the active setting of the Enable feature. This allows you to transmit manipulated information to the behavior model, for example.

The instrument regions can contain the following elements:

Inspect Value Displays the current value that is transmitted on the bus.

Model Signal Displays the value of the model signal or the I/O signal (if available). The model signal is calculated using the control algorithms of the behavior model. The I/O signal is read from the device under test.

- To transmit the values of the behavior model on the bus, set Source Switch to Model signal.
- To transmit the values of the device under test on the bus, set Source Switch to I/O signal.

Source Switch Lets you select the source of the signal value. Depending on the bus configuration, the following cases are possible:

- You can switch between the model signal (from the behavior model) and the substitute value.
- You can switch between the I/O signal (from the device under test) and the substitute value.

Refer to Configuring Test Automation Support (ConfigurationDesk I/O Function Implementation Guide 🕮).

Substitute Value Lets you specify the substitute value. To transmit the substitute value instead of the model signal value or I/O signal value, set Source Switch to Substitute value.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

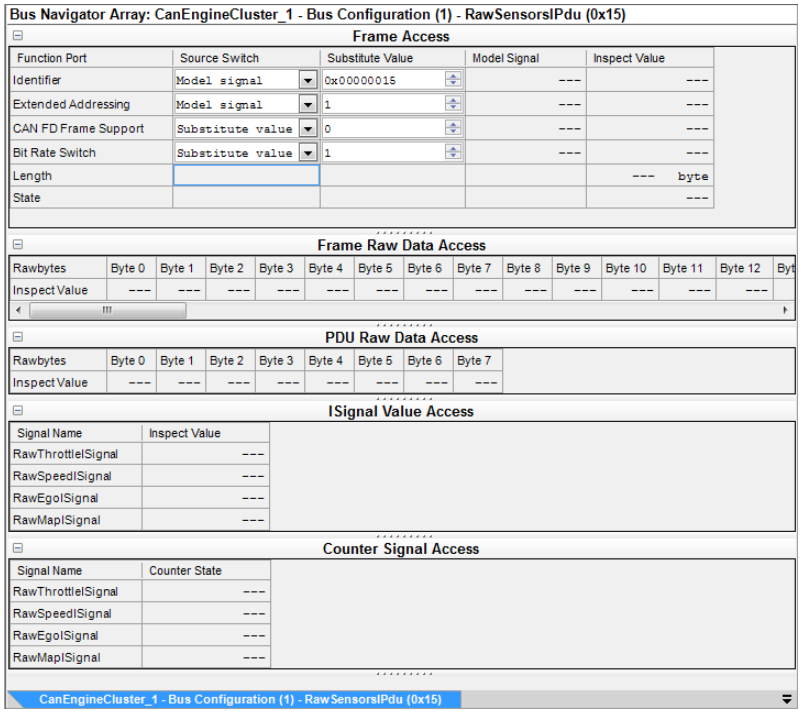
References

Generate Manipulation Layout..... 231

Bus Instrument (RX Type for CAN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To display information on an RX message.

Description

You can create one or more Bus Instruments (RX type for CAN) for each RX and TX/RX message/PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate RX Layout** command.

Instrument elements

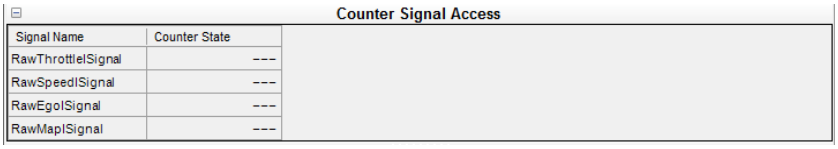
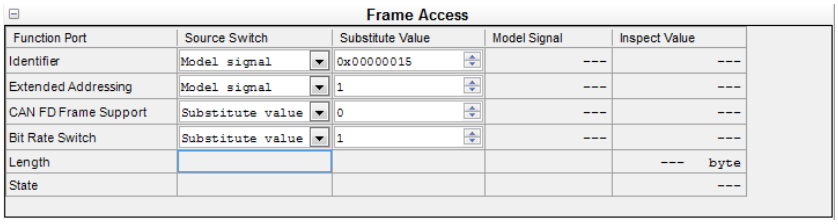
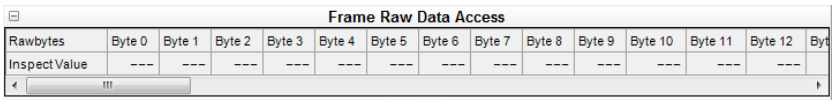
The Bus Instrument (RX type for CAN) displays information on received PDUs (RX PDUs) that have been configured in the Bus Manager for simulation purposes.

The instrument content is tailor-made, i.e., the displayed settings depend on the settings specified in the Bus Manager.

For information on the related configuration elements in the Bus Manager, refer to:

- Bus Manager in ConfigurationDesk: Working with Bus Configuration Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Working with Bus Configuration Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

The following regions can be available in the instrument:

Region	Description
Counter Signal Access	 Displays the counter state value: ▪ 4: Initial state, counter signal not received yet ▪ 0: Counter signal with expected counter value received ▪ 64: Counter signal with unexpected counter value received
Frame Access	 Displays the settings of the CAN frame the PDU is included in. Because the settings belong to the frame access feature, you can change them in RX instruments via substitute values. To transmit a substitute value on the bus, set **Source Switch** to **Substitute value**. The following frame functions are available: ▪ Identifier: Lets you specify the identifier of the frame. ▪ Standard identifier format: 0 … 2^{11} -1 ▪ Extended identifier format: 0 … 2^{29} -1 ▪ Extended Addressing: Lets you specify the identifier format of the frame. ▪ 0: Standard identifier format (11-bit) ▪ 1: Extended identifier format (29-bit) ▪ CAN FD Frame Support: Lets you specify the state of the frame's CAN FD support. ▪ 0: CAN FD support disabled, i.e., the frame is a classic CAN frame ▪ 1: CAN FD support enabled ▪ Bit Rate Switch: Lets you specify the state of the bit rate switch. ▪ 0: Bit rate switch disabled, i.e., the data phase of the frame is transmitted with the bit rate that is specified for the **Baud rate** property of the related CAN communication cluster. ▪ 1: Bit rate switch enabled, i.e., the data phase of the frame is transmitted with the bit rate that is specified for the **Data phase baud rate** property of the related CAN communication cluster. ▪ Length: Displays the actual payload length of the frame at run time. ▪ State: Displays the state of the frame. ▪ 0: The frame is not received in the current sampling step. ▪ 1: The frame is received in the current sampling step.
Frame Raw Data Access	 Displays the raw data for each byte of the frame.

Region	Description

Global Time Domain – BusCfgGTSTimeBaseDataAccess	**Global Time Domain CLST_A_TD_B**
	BusCfgGTSTimeBaseDataAccess | | Inspect Value
	Time | ---
	SynchronizedToGateway | No conversion
	UserDataLength | ---
	UserBytes [0] | ---
	UserBytes [1] | ---
	UserBytes [2] | ---
	
	Displays the data that is read from the synchronized time base manager.
	The following data can be displayed:
	▪ Time: The synchronized time that is received over the CAN bus.
	▪ SynchronizedToGateway: The value of the SYNCHRONIZED_TO_GATEWAY bit.
	  ▪ 0: The time base instance is synchronized to the global time master.
	  ▪ 1: The forwarding of the global time was interrupted, therefore the time base instance is not synchronized to the global time master.
	▪ UserDataLength: Specifies how many user bytes (0-3) are received.
	▪ UserBytes [0..2]: The user bytes that are received.

Global Time Domain – BusCfgGTSValidationAccess	**Global Time Domain CLST_A_TD_D** BusCfgGTSValidationAccess | | Source Switch | Substitute Value | Model Signal | Inspect Value PartialValidation | Model signal | False | No conver... | No conver... Result | | | | | --- Displays the result of the transport protocol validation and lets you enable partial validation. When **PartialValidation** is enabled, the following validation results are ignored: ▪ SYNC SQC OK ▪ SYNC CRC OK ▪ SYNC UNEXPECTED LEN ▪ FUP SQC OK ▪ FUP CRC OK ▪ FUP UNEXPECTED LEN ▪ FUP TIMEOUT For more information, refer to: ▪ Bus Manager in ConfigurationDesk: Accessing Validity Checks for Time Synchronization Messages (ConfigurationDesk Bus Manager Implementation Guide 🕮) ▪ Bus Manager (stand-alone): Accessing Validity Checks for Time Synchronization Messages (Bus Manager (Stand-Alone) Implementation Guide 🕮) If **Result** is 0, the validation tests were passed. If you enabled **PartialValidation**, this does not include the tests listed above.
ISignalGroup EndToEnd Protection Status Access	**ISignalGroup EndToEnd Protection Status Access** Signal Group | | | SignalGroup_E2E_P01A | | Inspect Value Calculated CRC | --- E2E State | No conver... Displays information on the status of a received end-to-end protected signal group. The following values are displayed: ▪ Calculated CRC: The checksum value that is calculated for the received signal group. ▪ E2E State: The state of the end-to-end protection, i.e., whether the received end-to-end protection information indicates a failure. For more information on the possible function port values, refer to: ▪ Bus Manager in ConfigurationDesk: Observing the Status of Received End-to-End-Protected ISignal Groups (ConfigurationDesk Bus Manager Implementation Guide 🕮) ▪ Bus Manager (stand-alone): Observing the Status of Received End-to-End-Protected ISignal Groups (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Region	Description
ISignal Value Access	**ISignal Value Access** Signal Name | Inspect Value RawThrottleISignal --- RawSpeedISignal --- RawEgoISignal --- RawMapISignal --- Displays the value of each ISignal in the PDU.
PDU Raw Data Access	**PDU Raw Data Access** Rawbytes | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 Inspect Value | --- | --- | --- | --- | --- | --- | --- | --- Displays the raw data for each byte of the PDU.
PDU RX Interrupt Access	**PDU RX Interrupt Access** Function Port | Source Switch | Substitute Value | Model Signal | Inspect Value Pdu Rx Interrupt Enable | Substitute value | True | No conversion | No conversion Lets you enable or disable the PDU RX Interrupt feature. For more information on RX interrupts, refer to: ▪ Bus Manager in ConfigurationDesk: Triggering the Execution of Functions in a Behavior Model via RX Interrupts (ConfigurationDesk Bus Manager Implementation Guide 🕮) ▪ Bus Manager (stand-alone): Triggering the Execution of Functions in a Behavior Model via RX Interrupts (Bus Manager (Stand-Alone) Implementation Guide 🕮)
PDU RX Status Access	**PDU RX Status Access** Counter | State | Time | Delta Time --- | No conver... | --- | --- Displays the status of the PDU.
PDU SecOC Access	**PDU SecOC Access** Function Port | Source Switch | Substitute Value | Model Signal | Inspect Value State | --- | Enable Verification | Model signal | 0 | --- | --- Displays the state of the verification, for example, whether the verified authentication information is valid, and lets you enable and disable verification of the authentication information.
PDU State	**PDU State** Bus Configuration Enable Communication Controller Enable Access PDU Enable Access Depending on the settings specified in the Bus Manager, the status of the entire bus configuration, the communication controller, and the PDU can be displayed. Green means that communication is enabled, red means that it is disabled.
PDU User Code Access	**PDU User Code Access** Function Port | Inspect Value Result | --- Displays the value that the user code execution returns.

The instrument regions can contain the following elements:

Counter Displays the value of the counter that is implemented for the PDU.

Counter State Displays the counter state value.

Delta Time Displays the time difference in seconds between the current and the previous reception of the PDU. By default, the initial value is 0 until the PDU

is received for the first time. Because there is no actual time difference when the PDU is received for the first time, the function port provides the time of the executable application. With the second reception of the PDU, the function port provides the actual time difference.

Function Port Displays function ports configured via the Bus Manager.

Inspect Value Displays the current value that is transmitted on the bus.

Model Signal Displays the value of the model signal or the I/O signal (if available). The model signal is calculated using the control algorithms of the behavior model. The I/O signal is read from the device under test.

- To transmit the values of the behavior model on the bus, set Source Switch to Model signal.
- To transmit the values of the device under test on the bus, set Source Switch to I/O signal.

Rawbytes – Byte <number> Displays the raw data for each byte of a frame or a PDU.

Signal Name Displays the name of the signal.

Source Switch Lets you switch between the behavior model (Model signal) and the substitute value as the source of the signal value.

When test automation support is enabled for a signal, you can specify the source of the signal value. You can use the static substitute value and adjust it in your experiment or exchange dynamic values with a behavior model via the model signal. Refer to Configuring Test Automation Support (ConfigurationDesk I/O Function Implementation Guide 🕮).

State Displays the state of the PDU, i.e., whether the PDU is received in the current sampling step (1) or not (0).

Time Displays the time in seconds of the executable application when the PDU was received. By default, the initial value is 0 until the PDU is received for the first time.

Substitute Value Lets you specify the substitute value. To transmit the substitute value instead of the value of the model signal, set Source Switch to Substitute value.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

Basics

Basics on Global Time Synchronization..... 45

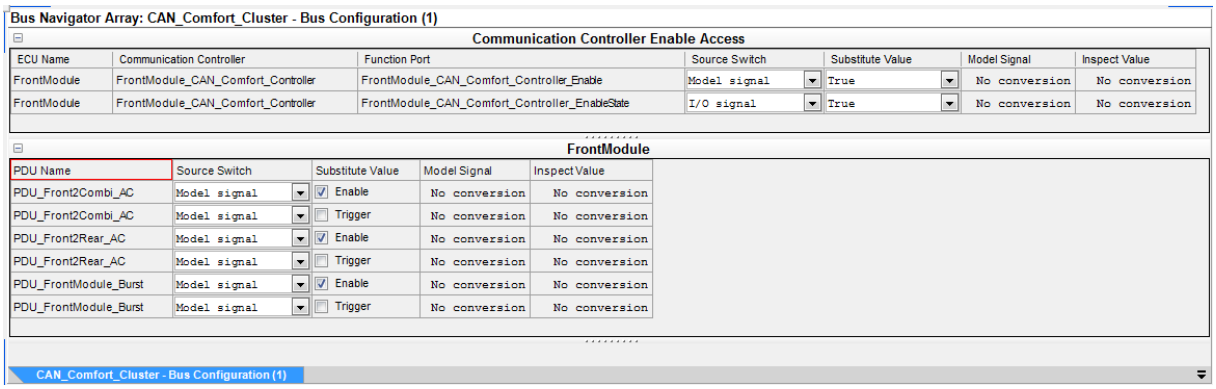
References

Generate RX Layout..... 232

Bus Instrument (TX Status Type for CAN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To specify whether TX messages/PDUs of a communication cluster are transmitted on the bus and to specify basic transmission options.

Description

You can create one or more Bus Instruments (TX status type for CAN) for each CAN communication cluster node in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate TX Status Layout** command. Refer to Generate TX Status Layout on page 235.

Instrument elements

The Bus Instrument (TX status type for CAN) lets you enable or disable communication controllers and specify trigger access options for PDUs.

The instrument displays all the communication controllers of the selected database for which test automation support is enabled in the bus configuration.

For information on the related configuration elements in the Bus Manager, refer to:

- Bus Manager in ConfigurationDesk: Working with Bus Configuration Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Working with Bus Configuration Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

The following elements can be available in the instrument:

Communication Controller Displays the name of the communication controller.

ECU Name Displays the name of the ECU.

Function Port Displays the name of the function port.

Inspect Value Displays the current value that is transmitted on the bus.

Model Signal Displays the model signal value. It is calculated using the control algorithms of the behavior model. To transmit the dynamic values of the behavior model on the bus, set **Source Switch** to **Model signal**.

PDU Name Displays the name of the PDU.

Source Switch Lets you switch between the behavior model (model signal) or the I/O signal from the device under test and the substitute value as the source of the signal value.

When test automation support is enabled for a signal, you can specify the source of the signal value. You can use the static substitute value and adjust it in your experiment or exchange dynamic values with a behavior model via the model signal. Refer to Configuring Test Automation Support (ConfigurationDesk I/O Function Implementation Guide 🕮).

Substitute Value Lets you specify the static substitute value. To transmit the substitute value on the bus, set **Source Switch** to **Substitute value**.

To disable a communication controller or the enable state signal of a communication controller, set **Source Switch** to **Substitute Value** and **Substitute Value** to **False**.

If enabling and disabling the transmission is specified for a PDU, you can enable or disable the transmission via the **Enable** checkbox in the related PDU region.

If triggered transmission is specified for a PDU, you can enable or disable triggering via the **Trigger** checkbox in the related PDU region.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

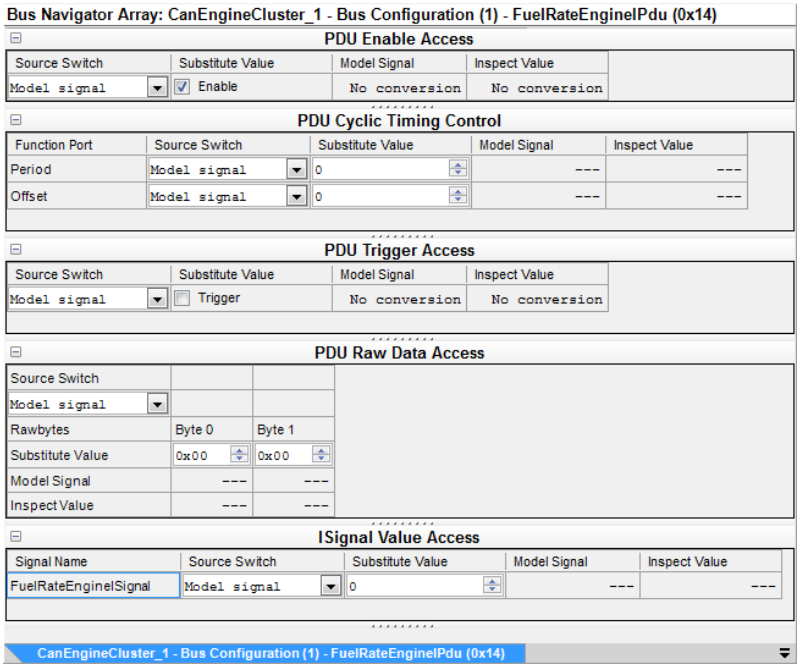
References

Generate TX Status Layout..... 235

Bus Instrument (TX Type for CAN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To display, configure, and transmit a TX message.

Description

You can create one or more Bus Instruments (TX type for CAN) for each TX and TX/RX message/PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the Generate TX Layout command.

Instrument elements

The Bus Instrument (TX type for CAN) lets you transmit TX PDUs that have been configured in the Bus Manager for simulation purposes on a CAN bus. The default transmission conditions are set via various options in the Bus Manager. If enabled, you can manipulate several frame-, PDU- and signal-specific settings before the PDU is transmitted on the bus.

The Bus Instrument (TX type for CAN) is tailor-made, i.e., it displays only the frame-, PDU-, and signal-specific settings which are enabled to be displayed and/or manipulated by ControlDesk during run-time.

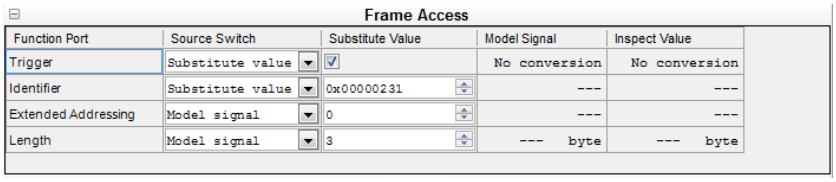
For information on the related configuration elements in the Bus Manager, refer to:

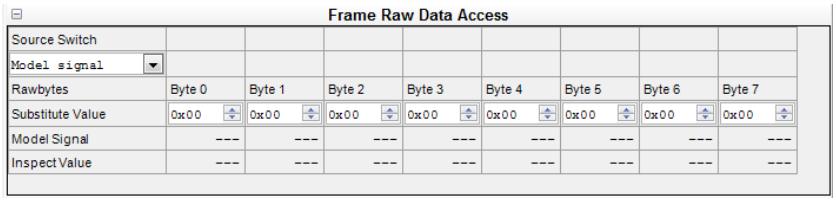
- Bus Manager in ConfigurationDesk: Working with Bus Configuration Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Working with Bus Configuration Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

The following regions can be available in the instrument:

Region	Description

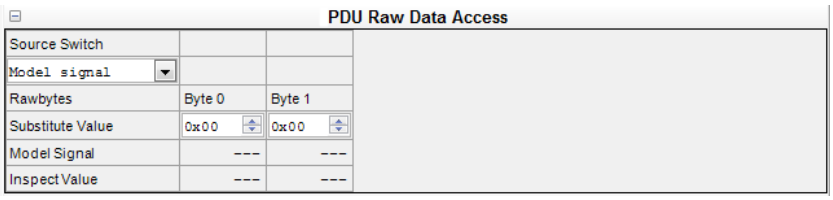
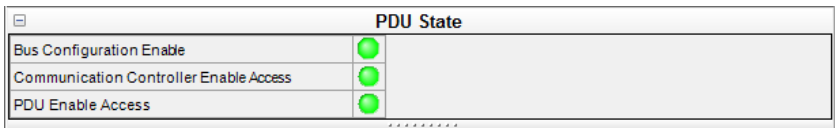
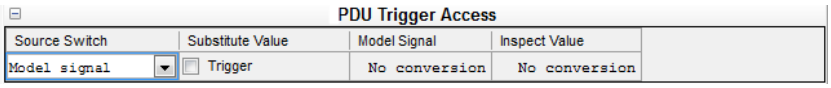
Frame Access	
	Displays the settings of the CAN frame the PDU is included in and lets you change the settings via substitute values. To transmit a substitute value on the bus, set **Source Switch** to **Substitute value**.
	The following frame functions are available:
	▪ Trigger: Lets you trigger the transmission of the frame.
	▪ Identifier: Lets you specify the identifier of the frame.
	  ▪ Standard identifier format: 0 … 2^{11} -1
	  ▪ Extended identifier format: 0 … 2^{29} -1
	▪ Extended Addressing: Lets you specify the identifier format of the frame.
	  ▪ 0: Standard identifier format (11-bit)
	  ▪ 1: Extended identifier format (29-bit)
	▪ Bit Rate Switch: Lets you specify the state of the bit rate switch. If switching the bit rate is enabled, the data phase of the frame can be transmitted with a higher bit rate. The bit rate switch value applies only to the frame if CAN FD frame support is enabled.
	  ▪ 0: Bit rate switch disabled, i.e., the data phase of the frame is transmitted with the bit rate that is specified for the **Baud rate** property of the related CAN communication cluster.
	  ▪ 1: Bit rate switch enabled, i.e., the data phase of the frame is transmitted with the bit rate that is specified for the **Data phase baud rate** property of the related CAN communication cluster.
	▪ Length: Lets you specify the actual payload length of the frame at run time. The default payload length is derived from the related setting in the communication matrix.
	  In the following cases, the specified length is automatically adjusted at run time:
	  ▪ For classic CAN frames, the maximum length is limited to 8 bytes.
	  ▪ A payload length that is invalid according to CAN FD is extended by padding bytes to the next higher valid length. The value of each padding byte is 255.

Frame Raw Data Access	 Displays the raw data for each byte of the frame's payload (including the PDU). Lets you specify the static substitute values in the range 0 … 255 (displayed in hexadecimal format). To transmit the substitute values on the bus, set **Source Switch** to **Substitute value**.

Region	Description

Global Time Domain – BusCfgGTSTimeBaseDataAccess	**Global Time Domain CLST_A_TD_B**
	BusCfgGTSTimeBaseDataAccess | | Inspect Value
	Time ---
	Status ---
	UserDataLength ---
	UserBytes [0] ---
	UserBytes [1] ---
	UserBytes [2] ---
	Displays the data that is read from the synchronized time base manager and sent over the CAN bus.
	The following data is displayed:
	▪ Time: The synchronized time that is sent over the CAN bus.
	▪ Status: The synchronization status of the time base instance. Only the SYNCHRONIZED_TO_GATEWAY bit is transmitted over CAN with the following information:
	  ▪ 0: The time base instance is synchronized to the global time master.
	  ▪ 1: The forwarding of the global time was interrupted, therefore the time base instance is not synchronized to the global time master.
	▪ UserDataLength: Specifies how many user bytes (0-3) are transferred.
	▪ UserBytes [0..2]: The user bytes that are transferred.

Global Time Domain – BusCfgGTSTransmissionControlAccess	**Global Time Domain CLST_A_TD_D** BusCfgGTSTransmissionControlAccess | | Source Switch | Substitute Value | Model Signal | Inspect Value Period Model signal 1 --- --- Trigger Model signal False No conversion No conversion Displays and lets you change transmission properties of the time synchronization PDU. To change the transmission period or enable/disable triggered transmission, set Source Switch to Substitute value. If you set Period to 0, cyclic time synchronization is disabled.
ISignal Value Access	**ISignal Value Access** Signal Name | Source Switch | Substitute Value | Model Signal | Inspect Value Front2Combi_AC Model signal 0 --- --- Displays the value of each ISignal in the PDU. Lets you specify the static substitute values. To transmit a substitute value on the bus, set Source Switch to Substitute value.
PDU Cyclic Timing Control	**PDU Cyclic Timing Control** Function Port | Source Switch | Substitute Val... | Model Signal | Inspect Value Period Model signal 0 --- --- Offset Model signal 0 --- --- Displays the cyclic timing control settings of the PDU and lets you specify the substitute values. To transmit a substitute value on the bus, set Source Switch to Substitute value. The following cyclic timing control functions are available: ▪ Period: Specifies a period (i.e., a cycle time) in seconds for the PDU. ▪ Offset: Specifies an offset (i.e., a delay time) in seconds for the PDU.
PDU Enable Access	**PDU Enable Access** Source Switch | Substitute Value | Model Signal | Inspect Value Model signal ☑ Enable No conversion No conversion Displays the transmission state of the PDU and lets you specify the substitute value. To enable or disable the transmission of the PDU via the substitute value's Enable checkbox, set Source Switch to Substitute value.

Region	Description
PDU Raw Data Access	 Displays the raw data for each byte of the PDU. Lets you specify the static substitute values (displayed in hexadecimal format). To transmit a substitute value on the bus, set Source Switch to Substitute value. If you use raw data for a PDU, you cannot use the values of the ISignals that are included in the PDU. In this case, the ISignal values are ignored and the raw data is used instead.
PDU State	 Depending on the settings specified in the Bus Manager, the status of the entire bus configuration, the communication controller, and the PDU can be displayed. Green means that communication is enabled, red means that it is disabled.
PDU Trigger Access	 Displays the trigger status of the PDU and lets you specify the substitute value. To enable or disable triggering via the substitute value's Trigger checkbox, set Source Switch to Substitute value. The behavior of the trigger depends on the trigger mode specified in the Bus Manager.

The instrument regions can contain the following elements:

Function Port Displays function ports configured via the Bus Manager.

Inspect Value Displays the current value that is transmitted on the bus.

Model Signal Displays the model signal value. It is calculated using the control algorithms of the behavior model. To transmit the dynamic values of the behavior model on the bus, set Source Switch to Model signal.

Rawbytes – Byte <number> Displays the raw data for each byte of a frame or a PDU. Lets you specify the static substitute value in hexadecimal format. To transmit the substitute value on the bus, set Source Switch to Substitute value.

If you use raw data for a PDU, you cannot use the values of the ISignals that are included in the PDU. In this case, the ISignal values are ignored and the raw data is used instead.

Signal Name Displays the name of the signal.

Source Switch Lets you switch between the behavior model (Model signal) and the substitute value as the source of the signal value.

When test automation support is enabled for a signal, you can specify the source of the signal value. You can use the static substitute value and adjust it in your experiment or exchange dynamic values with a behavior model via the model

signal. Refer to Configuring Test Automation Support (ConfigurationDesk I/O Function Implementation Guide 🕮).

Substitute Value Lets you specify the substitute value. To transmit the substitute value instead of the value of the model signal, set **Source Switch** to **Substitute value**.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

Basics

Basics on Global Time Synchronization.....45

References

Generate TX Layout.....234

LIN Instruments

Where to go from here

Information in this section

Bus Instrument (Inspection Type for LIN).....126
To display information on a PDU that has been configured for inspection purposes.

Bus Instrument (Manipulation Type for LIN).....128
To display the value of a PDU that has been configured for manipulation purposes.

Bus Instrument (RX Type for LIN).....131
To display information on an RX message.

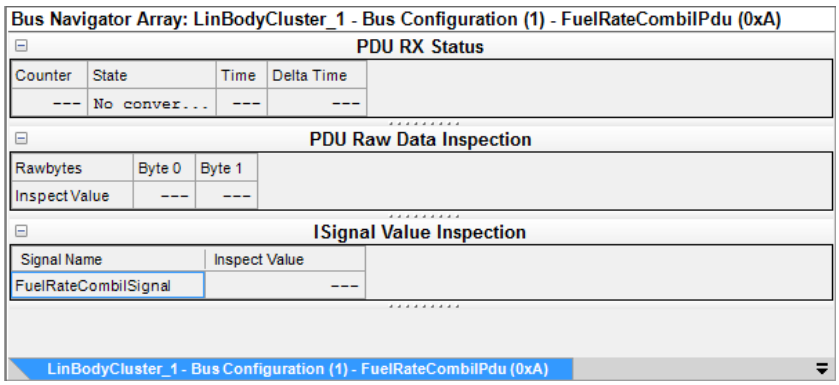
Bus Instrument (TX Status Type for LIN).....134
To specify whether TX messages/PDUs of a communication cluster are transmitted on the bus and to specify basic transmission options.

Bus Instrument (TX Type for LIN).....136
To display, configure, and transmit a TX message.

Bus Instrument (Inspection Type for LIN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To display information on a PDU that has been configured for inspection purposes.

Description

You can create one or more Bus Instruments (Inspection type for LIN) for each RX PDU that has been configured for inspection purposes. The instrument opens in a separate layout when you create it via the Generate Inspection Layout command.

Instrument elements

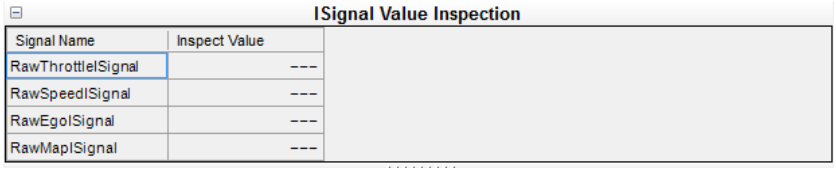
The Bus Instrument (Inspection type for LIN) displays information on received PDUs (RX PDUs) that have been configured for inspection purposes. The displayed information is view-only.

The instrument content is tailor-made, i.e., the displayed settings depend on the settings specified in the Bus Manager.

For information on the related configuration elements in the Bus Manager, refer to:

- Bus Manager in ConfigurationDesk: Working with Bus Configuration Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Working with Bus Configuration Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

The following regions can be available in the instrument:

Region	Description
ISignal Value Inspection	

Region	Description
	Displays all available ISignal values of a PDU.
ISignalGroup EndToEnd Protection Status Inspection	**ISignalGroup EndToEnd Protection Status Inspection** Signal Group SignalGroup_E2E_BE_P01 — Inspect Value Calculated CRC — --- E2E State — No conver... Displays information on the status of a received end-to-end protected signal group. The following values are displayed: ▪ Calculated CRC: The checksum value that is calculated for the received signal group. ▪ E2E State: The state of the end-to-end protection, i.e., whether the received end-to-end protection information indicates a failure. For more information on possible function port values, refer to: ▪ Bus Manager in ConfigurationDesk: Observing the Status of Received End-to-End-Protected ISignal Groups (ConfigurationDesk Bus Manager Implementation Guide 🕮) ▪ Bus Manager (stand-alone): Observing the Status of Received End-to-End-Protected ISignal Groups (Bus Manager (Stand-Alone) Implementation Guide 🕮)
PDU Raw Data Inspection	**PDU Raw Data Inspection** Rawbytes — Byte 0 — Byte 1 — Byte 2 — Byte 3 — Byte 4 — Byte 5 — Byte 6 — Byte 7 Inspect Value — --- — --- — --- — --- — --- — --- — --- — --- Displays the raw data values for each byte of a PDU.
PDU RX Status	**PDU RX Status** Counter — State — Time — Delta Time --- — No conver... — --- — --- Displays status information on the PDU.
PDU State	**PDU State** Bus Configuration Enable Displays the status of the entire bus configuration. Green means that communication is enabled, red means that it is disabled.
PDU User Code Inspection	**PDU User Code Inspection** Function Port — Inspect Value Result — --- Displays the value that the user code execution returns.

The instrument regions can contain the following elements:

Counter Displays the value of the counter. The counter value is incremented by 1 each time the PDU is received via the bus. If an overflow occurs, the counter value is reset to 0.

Delta Time Displays the delta time (in seconds) between the current and the previous reception of the PDU. Since there is no actual delta time when the PDU is received for the first time, the function port provides the time of the simulation

application in this case. When the PDU is received for the second time, the function port provides the actual delta time.

Function Port Displays function ports configured via the Bus Manager.

Inspect Value Displays the current value that is transmitted on the bus.

Rawbytes – Byte <number> Displays the raw data value for a specific byte of a PDU.

Signal Name Displays the name of the signal.

State Displays the state of the PDU, i.e., if the PDU is received in the current sampling step:

- 0: The PDU is not received in the current sampling step.
- 1: The PDU is received in the current sampling step.

Time Displays the time (in seconds) of the simulation application when the PDU was received.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

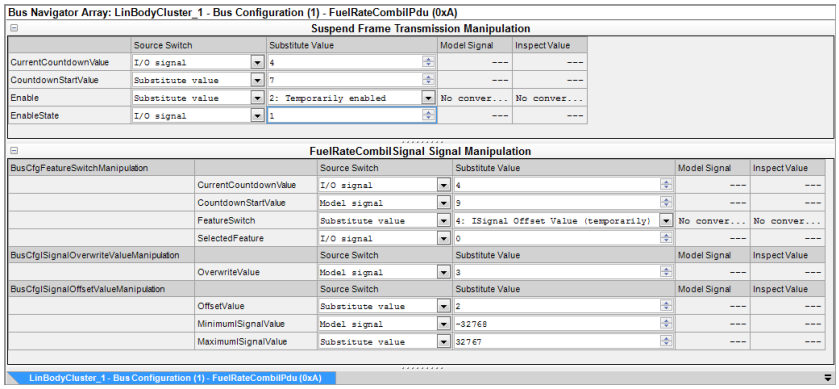
References

Generate Inspection Layout..... 230

Bus Instrument (Manipulation Type for LIN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To change the settings of a PDU that has been configured for manipulation purposes.

Description

You can create one or more Bus Instruments (Manipulation type for LIN) for each TX PDU that has been configured for manipulation purposes. The instrument opens in a separate layout when you create it via the **Generate Manipulation Layout** command.

Instrument elements

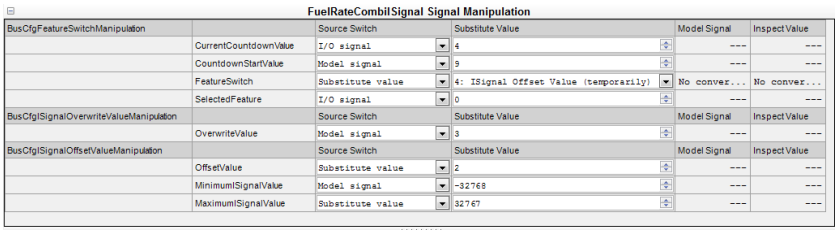
The Bus Instrument (Manipulation type for LIN) displays information on PDUs that have been configured for manipulation purposes.

The instrument displays settings depending on the settings specified in the Bus Manager.

For information on the related configuration elements in the Bus Manager, refer to:

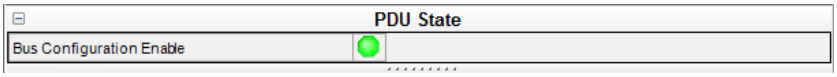
- Bus Manager in ConfigurationDesk: Working with Bus Configuration Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Working with Bus Configuration Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

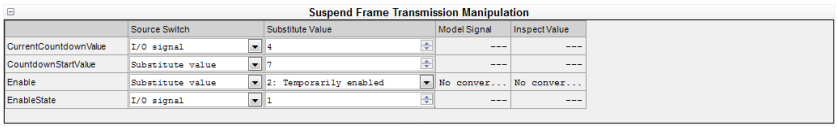
The following regions can be available in the instrument:

Region	Description
<ISignal Name> Signal Manipulation	 Displays the available signal manipulation features and lets you select the one you want to use for the manipulation of the bus communication. The following functions are available to select and configure the manipulation: ▪ **CurrentCountdownValue**: Lets you specify the current countdown value. This value is evaluated only if the bus manipulation feature is temporarily enabled. ▪ **CountdownStartValue**: Lets you specify the countdown start value. If the bus manipulation feature is temporarily enabled, this value determines how often the ISignal is manipulated. ▪ **FeatureSwitch**: Displays the available manipulation features and lets you select one. The selection also specifies whether the manipulation applies permanently or temporary. If you

FuelRateCombiISignal Signal Manipulation

BusCfgFeatureSwitchManipulation		Source Switch	Substitute Value	Model Signal	Inspect Value
	CurrentCountdownValue	I/O signal	4	---	---
	CountdownStartValue	Model signal	9	---	---
	FeatureSwitch	Substitute value	4: ISignal Offset Value (temporarily)	No conver...	No conver...
	SelectedFeature	I/O signal	0	---	---
BusCfgISignalOverwriteValueManipulation		Source Switch	Substitute Value	Model Signal	Inspect Value
	OverwriteValue	Model signal	3	---	---
BusCfgISignalOffsetValueManipulation		Source Switch	Substitute Value	Model Signal	Inspect Value
	OffsetValue	Substitute value	2	---	---
	MinimumISignalValue	Model signal	-32768	---	---
	MaximumISignalValue	Substitute value	32767	---	---

Region	Description
	select a temporarily manipulation, the CountdownStartValue and the CurrentCountdownValue are used to control the manipulation. ▪ SelectedFeature: Lets you specify the number that is provided as the number of the active manipulation feature. This allows you to transmit manipulated information to the behavior model, for example. The following functions are available for the OverwriteValueManipulation feature: ▪ Overwrite value: Lets you specify the value of the ISignal. The following functions are available for the SignalOffsetValueManipulation feature: ▪ Offset value: Lets you apply an offset value to the current value of the ISignal. ▪ MinimumISignalValue: Lets you specify a minimum value for the ISignal. ▪ MaximumISignalValue: Lets you specify a maximum value for the ISignal.
PDU State	 Displays the status of the entire bus configuration. Green means that communication is enabled, red means that it is disabled.

Suspend Frame Transmission Manipulation	
	Lets you suspend the transmission of the frame the PDU is included in.
	The following functions are available:
	▪ CurrentCountdownValue: Lets you specify the current countdown value. This value is evaluated only if the frame transmission suspension is temporarily enabled.
	▪ CountdownStartValue: Lets you specify the countdown start value. If the frame transmission suspension is temporarily enabled, this value determines how often the frame transmission is suspended.
	▪ Enable: Lets you enable the frame transmission. The following settings are available:
	  ▪ 0: Disabled
	  ▪ 1: Permanently enabled
	  ▪ 2: Temporarily enabled
	▪ EnableState: Lets you specify the number that is provided as the number of the active setting of the Enable feature. This allows you to transmit manipulated information to the behavior model, for example.

The instrument regions can contain the following elements:

Inspect Value Displays the current value that is transmitted on the bus.

Model Signal Displays the value of the model signal or the I/O signal (if available). The model signal is calculated using the control algorithms of the behavior model. The I/O signal is read from the device under test.

- To transmit the values of the behavior model on the bus, set Source Switch to Model signal.
- To transmit the values of the device under test on the bus, set Source Switch to I/O signal.

Source Switch Lets you select the source of the signal value. Depending on the bus configuration, the following cases are possible:

- You can switch between the model signal (from the behavior model) and the substitute value.

- You can switch between the I/O signal (from the device under test) and the substitute value.

Refer to Configuring Test Automation Support (ConfigurationDesk I/O Function Implementation Guide 🕮).

Substitute Value Lets you specify the substitute value. To transmit the substitute value instead of the model signal value or I/O signal value, set Source Switch to Substitute value.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

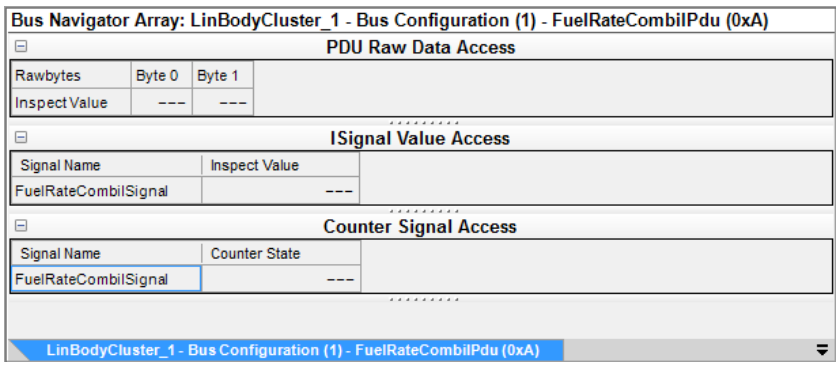
References

Generate Manipulation Layout..... 231

Bus Instrument (RX Type for LIN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To display information on an RX message.

Description

You can create one or more Bus Instruments (RX type for LIN) for each RX and TX/RX frame/PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the Generate RX Layout command.

Instrument elements

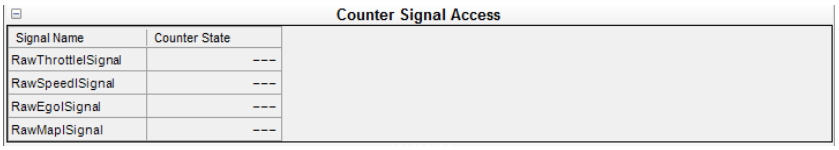
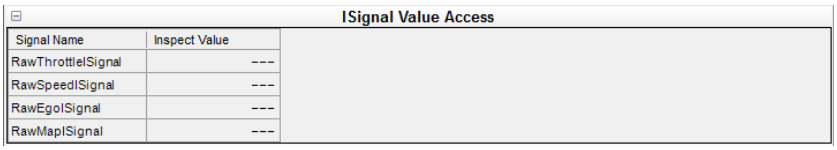
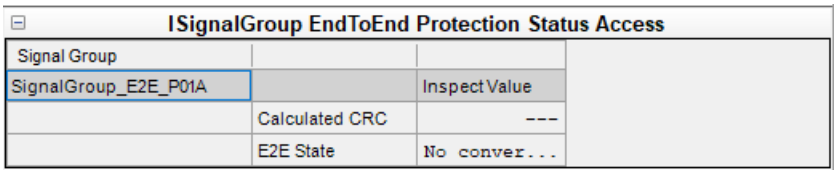
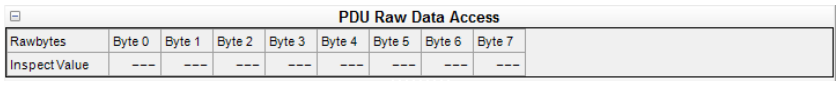
The Bus Instrument (RX type for LIN) displays information on received PDUs (RX PDUs) that have been configured in the Bus Manager for simulation purposes. The displayed information is view-only.

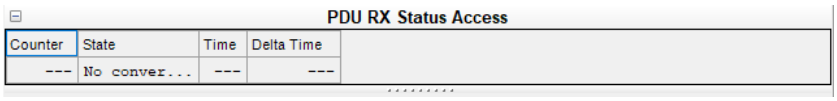
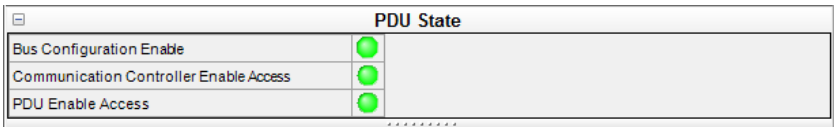
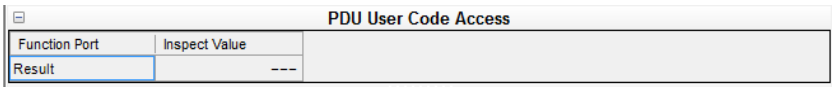
The instrument content is tailor-made, i.e., the displayed settings depend on the settings specified in the Bus Manager.

For information on the related configuration elements in the Bus Manager, refer to:

- Bus Manager in ConfigurationDesk: Working with Bus Configuration Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Working with Bus Configuration Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

The following regions can be available in the instrument:

Region	Description
Counter Signal Access	 Displays the counter state value: ▪ 4: Initial state, counter signal not received yet ▪ 0: Counter signal with expected counter value received ▪ 64: Counter signal with unexpected counter value received
ISignal Value Access	 Displays the value of each ISignal in the PDU.
ISignalGroup EndToEnd Protection Status Access	 Displays information on the status of a received end-to-end protected signal group. The following values are displayed: ▪ Calculated CRC: The checksum value that is calculated for the received signal group. ▪ E2E State: The state of the end-to-end protection, i.e., whether the received end-to-end protection information indicates a failure. For more information on the possible function port values, refer to: ▪ Bus Manager in ConfigurationDesk: Observing the Status of Received End-to-End-Protected ISignal Groups (ConfigurationDesk Bus Manager Implementation Guide 🕮) ▪ Bus Manager (stand-alone): Observing the Status of Received End-to-End-Protected ISignal Groups (Bus Manager (Stand-Alone) Implementation Guide 🕮)
PDU Raw Data Access	 Displays the raw data for each byte of the PDU.

Region	Description
PDU RX Status Access	 Displays the status of the PDU.
PDU State	 Depending on the settings specified in the Bus Manager, the status of the entire bus configuration, the communication controller, and the PDU can be displayed. Green means that communication is enabled, red means that it is disabled.
PDU User Code Access	 Displays the value that the user code execution returns.

The instrument regions can contain the following elements:

Counter Displays the value of the counter that is implemented for the PDU.

Counter State Displays the counter state value.

Delta Time Displays the time difference in seconds between the current and the previous reception of the PDU. By default, the initial value is 0 until the PDU is received for the first time. Because there is no actual time difference when the PDU is received for the first time, the function port provides the time of the executable application. With the second reception of the PDU, the function port provides the actual time difference.

Function Port Displays function ports configured via the Bus Manager.

Inspect Value Displays the current value that is transmitted on the bus.

Rawbytes – Byte <number> Displays the raw data for each byte of a PDU.

Signal Name Displays the name of the signal.

State Displays the state of the PDU, i.e., whether the PDU is received in the current sampling step (1) or not (0).

Time Displays the time in seconds of the executable application when the PDU was received. By default, the initial value is 0 until the PDU is received for the first time.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

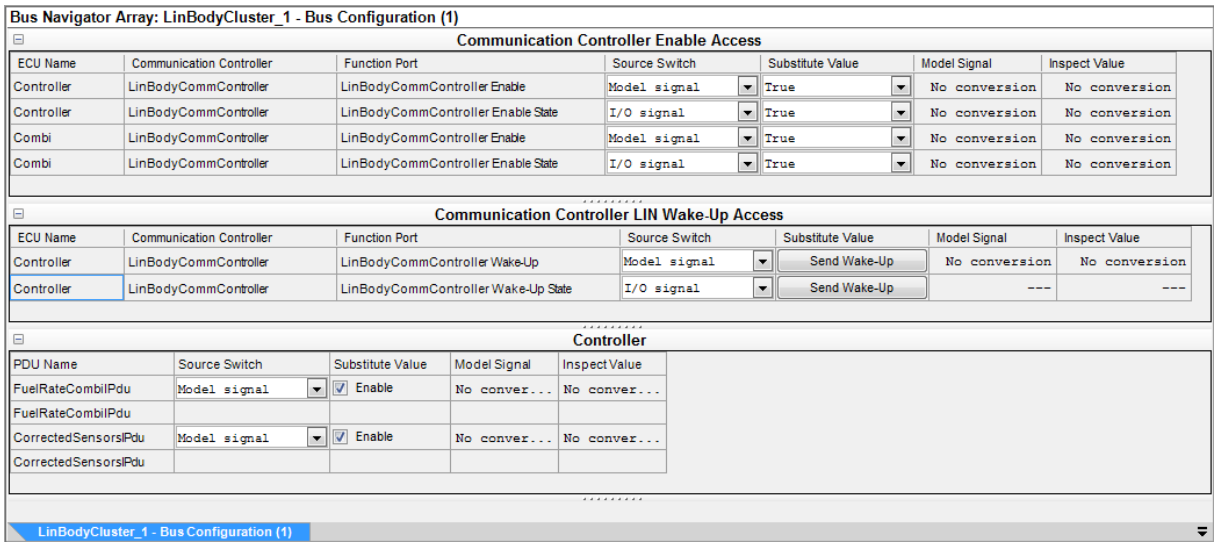
References

Generate RX Layout..... 232

Bus Instrument (TX Status Type for LIN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To specify whether TX messages/PDUs of a communication cluster are transmitted on the bus and to specify basic transmission options.

Description

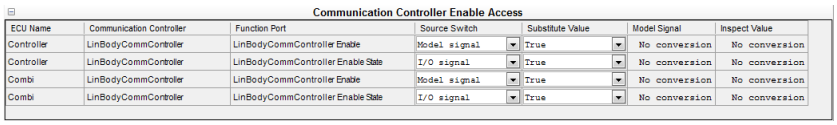
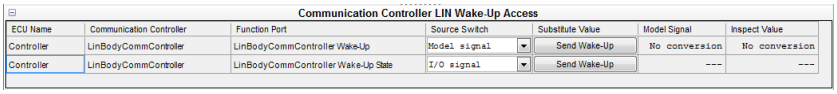
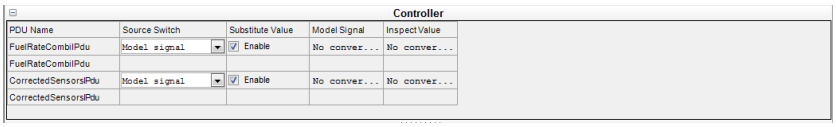
You can create one or more Bus Instruments (TX status type for LIN) for each LIN communication cluster node in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate TX Status Layout** command.

The Bus Instrument (TX status type for LIN) lets you enable or disable communication controllers and specify trigger access options for PDUs. The instrument displays all the communication controllers of the selected communication cluster for which test automation support is enabled in the bus configuration.

For information on the related configuration elements in the Bus Manager, refer to:

- Bus Manager in ConfigurationDesk: Working with Bus Configuration Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Working with Bus Configuration Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Instrument elements The following regions can be available in the instrument:

Region	Description
Communication Controller Enable Access	 Displays the enable settings and the enable state of the communication controllers of the network nodes (ECUs) that are members of the communication cluster and lets you specify substitute values. To enable or disable a communication controller or change the enable state signal via the substitute value, set **Source Switch** to **Substitute Value**. The following functions are available: ▪ Enable: Lets you enable or disable the communication controller. ▪ Enable State: Lets you change the value of the state signal of the communication controller.
Communication Controller LIN Wake-Up Access	 Displays the LIN wake-up state of the communication controllers of a communication cluster (separated by ECUs) and lets you specify substitute values. To transmit a substitute value on the bus, set **Source Switch** to **Substitute value**. The following LIN wake-up functions are available: ▪ Wake-Up: Lets you send a wake-up signal. ▪ Wake-Up State: Lets you change the value of the wake-up state signal.
<Network node (ECU)>	 Displays the transmission states of the PDUs that are available for a network node (ECU) and lets you specify substitute values. Depending on the configuration of a PDU in the Bus Manager, you can enable or disable transmission or triggering for a PDU.

The instrument regions can contain the following elements:

Communication Controller Displays the name of the communication controller.

ECU Name Displays the name of the ECU.

Function Port Displays the name of the function port.

Inspect Value Displays the current value that is transmitted on the bus.

Model Signal Displays the model signal value. It is calculated using the control algorithms of the behavior model. To transmit the dynamic values of the behavior model on the bus, set **Source Switch** to **Model signal**.

PDU Name Displays the name of the PDU.

Source Switch Lets you switch between the behavior model (model signal) and the substitute value as the source of the signal value.

When test automation support is enabled for a signal, you can specify the source of the signal value. You can use the static substitute value and adjust it in your

experiment or exchange dynamic values with a behavior model via the model signal. Refer to Configuring Test Automation Support (ConfigurationDesk I/O Function Implementation Guide 🕮).

Substitute Value Lets you specify the substitute value. To transmit the substitute value instead of the model signal's value, set Source Switch to Substitute value.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

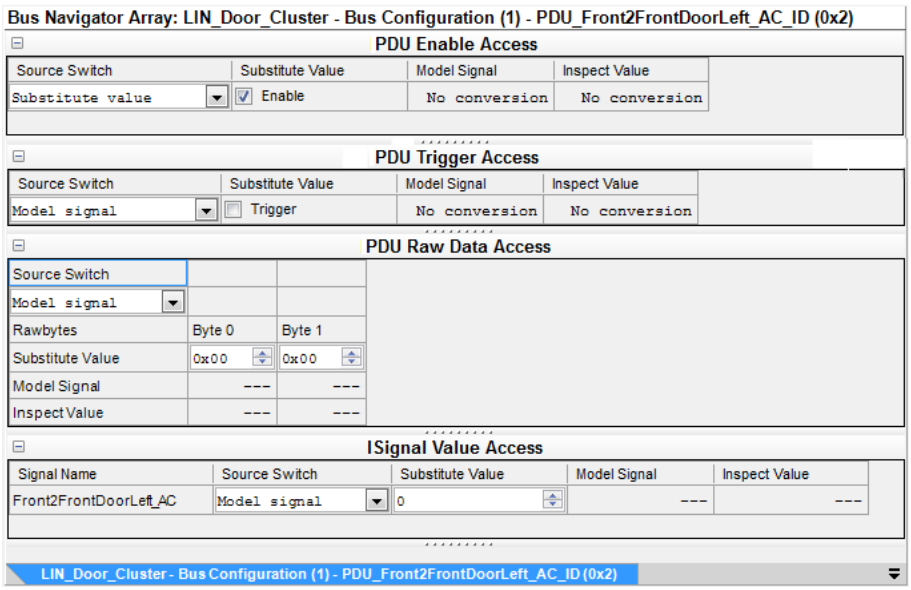
References

Generate TX Status Layout..... 235

Bus Instrument (TX Type for LIN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To display, configure, and transmit a TX message.

Description

You can create one or more Bus Instruments (TX type for LIN) for each TX and TX/RX frame/PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate TX Layout** command.

Instrument elements

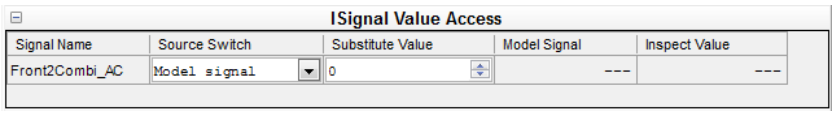
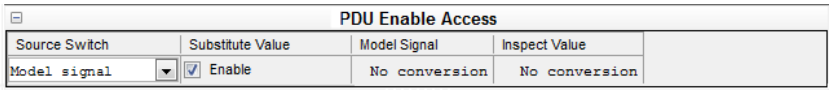
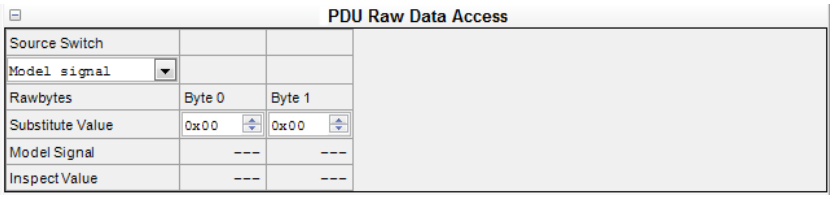
The Bus Instrument (TX type for LIN) lets you transmit TX PDUs that have been configured in the Bus Manager for simulation purposes on a LIN bus. The default transmission conditions are set via various options in the Bus Manager. If enabled, you can manipulate several frame- and signal-specific settings before the frame is transmitted on the bus.

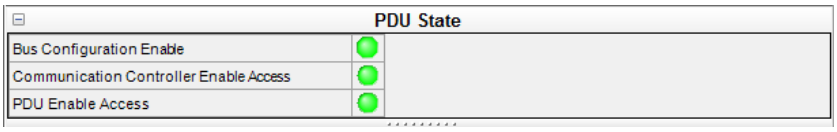
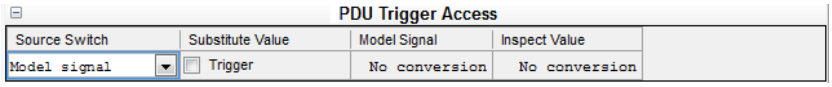
The Bus Instrument (TX type for LIN) is tailor-made, i.e., it displays only the PDU- and signal-specific settings which are enabled to be displayed and/or manipulated by ControlDesk during run time.

For information on the related configuration elements in the Bus Manager, refer to:

- Bus Manager in ConfigurationDesk: Working with Bus Configuration Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Working with Bus Configuration Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

The following regions can be available in the instrument:

Region	Description
ISignal Value Access	 Displays the value of each ISignal in the PDU. Lets you specify the static substitute values. To transmit a substitute value on the bus, set **Source Switch** to **Substitute value**.
PDU Enable Access	 Displays the transmission state of the PDU and lets you specify the substitute value. To enable or disable the transmission of the PDU via the substitute value's **Enable** checkbox, set **Source Switch** to **Substitute value**.
PDU Raw Data Access	 Displays the raw data for each byte of the PDU. Lets you specify the static substitute values (displayed in hexadecimal format). To transmit a substitute value on the bus, set **Source Switch** to **Substitute value**. If you use raw data for a PDU, you cannot use the values of the ISignals that are included in the PDU. In this case, the ISignal values are ignored and the raw data is used instead.

Region	Description
PDU State	 Depending on the settings specified in the Bus Manager, the status of the entire bus configuration, the communication controller, and the PDU can be displayed. Green means that communication is enabled, red means that it is disabled.
PDU Trigger Access	 Displays the trigger status of the PDU and lets you specify the substitute value. To enable or disable triggering via the substitute value's **Trigger** checkbox, set **Source Switch** to **Substitute value**. The behavior of the trigger depends on the trigger mode specified in the Bus Manager.

The instrument regions can contain the following elements:

Inspect Value Displays the current value that is transmitted on the bus.

Model Signal Displays the model signal value. It is calculated using the control algorithms of the behavior model. To transmit the dynamic values of the behavior model on the bus, set **Source Switch** to **Model signal**.

Rawbytes – Byte <number> Displays the raw data for each byte of a PDU. Lets you specify the static substitute value in hexadecimal format. To transmit the substitute value on the bus, set **Source Switch** to **Substitute value**.

If you use raw data for a PDU, you cannot use the values of the ISignals that are included in the PDU. In this case, the ISignal values are ignored and the raw data is used instead.

Signal Name Displays the name of the signal.

Source Switch Lets you switch between the behavior model (**Model signal**) and the substitute value as the source of the signal value.

When test automation support is enabled for a signal, you can specify the source of the signal value. You can use the static substitute value and adjust it in your experiment or exchange dynamic values with a behavior model via the model signal. Refer to Configuring Test Automation Support (ConfigurationDesk I/O Function Implementation Guide 🕮).

Substitute Value Lets you specify the substitute value. To transmit the substitute value instead of the value of the model signal, set **Source Switch** to **Substitute value**.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

References

Generate TX Layout..... 234

Bus Instruments (Blockset)

Where to go from here

Information in this section

CAN Instruments..... 139

FlexRay Instruments..... 154

LIN Instruments..... 169

CAN Instruments

Where to go from here

Information in this section

Bus Instrument (Gateway Type for CAN)..... 140
To specify whether or not messages are gatewayed between two CAN buses and to change the specified gateway filters during run time.

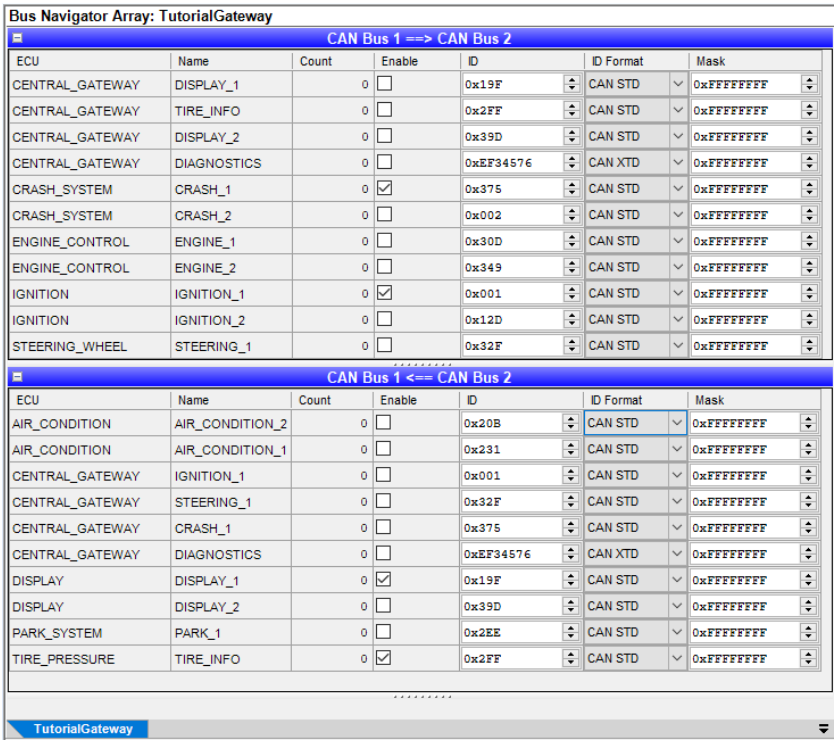
Bus Instrument (RX Type for CAN)..... 142
To display information on an RX message.

Bus Instrument (TX Status Type for CAN)..... 145
To specify whether TX messages/PDUs of a communication cluster are transmitted on the bus and to specify basic transmission options.

Bus Instrument (TX Type for CAN)..... 147
To display, configure, and transmit a TX message.

Bus Instrument (Gateway Type for CAN)

Illustration



Bus Navigator Array: TutorialGateway

CAN Bus 1 ==> CAN Bus 2

ECU	Name	Count	Enable	ID	ID Format	Mask
CENTRAL_GATEWAY	DISPLAY_1	0	☐	0x19F	CAN STD	0xFFFFFFFF
CENTRAL_GATEWAY	TIRE_INFO	0	☐	0x2FF	CAN STD	0xFFFFFFFF
CENTRAL_GATEWAY	DISPLAY_2	0	☐	0x39D	CAN STD	0xFFFFFFFF
CENTRAL_GATEWAY	DIAGNOSTICS	0	☐	0xEF34576	CAN XTD	0xFFFFFFFF
CRASH_SYSTEM	CRASH_1	0	☑	0x375	CAN STD	0xFFFFFFFF
CRASH_SYSTEM	CRASH_2	0	☐	0x002	CAN STD	0xFFFFFFFF
ENGINE_CONTROL	ENGINE_1	0	☐	0x30D	CAN STD	0xFFFFFFFF
ENGINE_CONTROL	ENGINE_2	0	☐	0x349	CAN STD	0xFFFFFFFF
IGNITION	IGNITION_1	0	☑	0x001	CAN STD	0xFFFFFFFF
IGNITION	IGNITION_2	0	☐	0x12D	CAN STD	0xFFFFFFFF
STEERING_WHEEL	STEERING_1	0	☐	0x32F	CAN STD	0xFFFFFFFF

CAN Bus 1 <== CAN Bus 2

ECU	Name	Count	Enable	ID	ID Format	Mask
AIR_CONDITION	AIR_CONDITION_2	0	☐	0x20B	CAN STD	0xFFFFFFFF
AIR_CONDITION	AIR_CONDITION_1	0	☐	0x231	CAN STD	0xFFFFFFFF
CENTRAL_GATEWAY	IGNITION_1	0	☐	0x001	CAN STD	0xFFFFFFFF
CENTRAL_GATEWAY	STEERING_1	0	☐	0x32F	CAN STD	0xFFFFFFFF
CENTRAL_GATEWAY	CRASH_1	0	☐	0x375	CAN STD	0xFFFFFFFF
CENTRAL_GATEWAY	DIAGNOSTICS	0	☐	0xEF34576	CAN XTD	0xFFFFFFFF
DISPLAY	DISPLAY_1	0	☑	0x19F	CAN STD	0xFFFFFFFF
DISPLAY	DISPLAY_2	0	☐	0x39D	CAN STD	0xFFFFFFFF
PARK_SYSTEM	PARK_1	0	☐	0x2EE	CAN STD	0xFFFFFFFF
TIRE_PRESSURE	TIRE_INFO	0	☑	0x2FF	CAN STD	0xFFFFFFFF

TutorialGateway

Purpose

To specify whether or not messages are gatewayed between two CAN buses and to change the specified gateway filters during run time.

Description

You can create one or more Bus Instruments (Gateway type for CAN) for each gateway node in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate Gateway Layout** command.

Gatewaying means exchanging messages between two CAN buses. In the real-time model, you can specify gateways using the RTICANMM Gateway block. In the RTICANMM Gateway block, you can exclude messages from being gatewayed and enable the changing of message exclude lists during run time. Refer to RTICANMM Gateway (RTI CAN MultiMessage Blockset Reference 🕮).

The Bus Instrument (Gateway type for CAN) is available if changing the message exclude lists during run time is enabled in the RTICANMM Gateway block. The instrument provides message filters to specify whether or not messages are being gatewayed. The filters are available for each TX message of an ECU. Depending on the settings specified in the RTICANMM Gateway block, the filters are available for only one or for both gateway directions. The default settings of the filters are the settings specified in the message exclude lists of the RTICANMM Gateway block.

Instrument elements

Information at the top of the instrument Gateway direction: Displays the gateway direction for which the specified filters are valid.

Count Displays the number of message transmissions for which the message is excluded from gatewaying. The running hit counter indicates that the message filter is active and working accurately.

Destination (Available only for J1939 messages) Displays the ID of the receiving ECU (destination ECU) of J1939 peer-to-peer messages. You can change the destination ECU to exclude other messages from being gatewayed.

ECU Displays all the ECUs that TX messages are specified for.

Enable Lets you enable and disable the filter. If selected, the filter is enabled and the message is excluded from being gatewayed.

ID Displays the filter ID and lets you change the specified filter ID. The filter ID determines which messages are filtered. Messages whose IDs are identical to the filter ID are filtered. However, you can specify a filter mask to filter messages whose IDs differ slightly from the filter ID.

ID Format Displays the format of a message. For a free raw message, you can select a message format for the filter. The selected format affects the number of raw bytes to be sent for the selected message, and specifies the message's bit rate switch (BRS) value (CAN FD only).

Mask Lets you specify a mask for the filter. The mask lets you filter messages whose IDs are not identical to the filter ID. The mask determines which bits of the message ID and filter ID are compared. If these bits are identical, the message is filtered even if other bits of its ID differ from the filter ID.

Example: The specified mask is 000011. This means, the last two bits of a message ID are compared with the last two bits of the filter ID. Suppose the filter ID is 123456.

- The message ID is 789056. The last two bits of the message and filter IDs are identical and the message is filtered.
- The message ID is 123400. The last two bits of the message and filter IDs differ and the message is not filtered.

Name Displays the name of the TX message the filter is referenced to.

Priority (Available only for J1939 messages) Displays the message priority that is filtered by the specified filter.

Source (Available only for J1939 messages) Displays the ID of the transmitting ECU (source ECU) of J1939 messages. You can change the source ECU to exclude other messages from being gatewayed.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

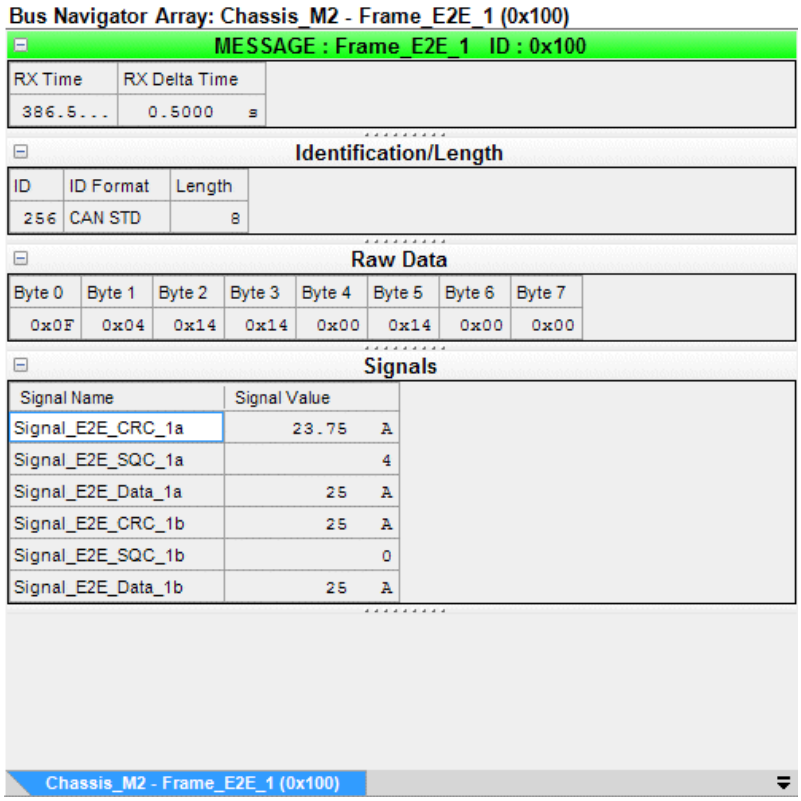
Related topics

References

Generate Gateway Layout..... 229

Bus Instrument (RX Type for CAN)

Illustration



Purpose

To display information on an RX message.

Description

You can create one or more Bus Instruments (RX type for CAN) for each RX and TX/RX message/PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate RX Layout** command.

Instrument elements

The Bus Instrument (RX type for CAN) displays information on received messages (RX messages). The instrument content is tailor-made, i.e., the displayed settings

depend on the settings specified in the RTICANMM MainBlock of the real-time model, and on the message type (standard CAN message or J1939 message).

The header color of the instrument indicates whether the message is active or inactive:

Header Color	Description
Red	The message is inactive.
Green	The message is active. However, a green header does not indicate that the message is being transmitted or received. The indicator for the transmission or reception of a message is a running RX time.

Byte <number> Displays the raw data for each byte of an RX message. This display must be enabled on the **RX Raw Data Display** page of the RTICANMM MainBlock. Refer to RX Raw Data Display Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

CRC Displays the checksum calculation status. Available if a signal is defined as a checksum signal on the **Checksum Messages** page of the RTICANMM MainBlock. Refer to Checksum Messages Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Error Displays the transmission of a signal's error value. If the LED is red, the error value is transmitted. Available if an error value is defined for a signal and signal errors are enabled on the **Signal Errors** page of the RTICANMM MainBlock. Refer to Signal Errors Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

ID (Not available for J1939 messages) Displays the ID of an RX message.

ID Format Displays the format of a message. The selected format affects the number of raw bytes to be sent for the selected message, and specifies the message's bit rate switch (BRS) value (CAN FD only). Refer to Free Raw Messages Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

J1939 DA (Available only for J1939 messages) Displays the address of the receiving ECU (destination ECU) of a J1939 RX message. For broadcast messages, 255 is displayed because they are received by all the ECUs of a CAN bus.

J1939 DN (Available only for J1939 peer-to-peer messages) Displays the ID of the receiving ECU (destination ECU) of a J1939 peer-to-peer RX message.

J1939 PGN (Available only for J1939 messages) Displays the parameter group number of a J1939 RX message.

J1939 SA (Available only for J1939 messages) Displays the address of the source ECU of a J1939 RX message.

J1939 SN (Available only for J1939 messages) Displays the ID of the source ECU of a J1939 RX message.

Length Displays the number of bytes defined for the RX message.

Message Prio (Available only for J1939 messages) Displays the priority of a J1939 RX message.

RX Delta Time Displays the delta time between the current and previous reception of the RX message.

RX Time Displays the time the RX message was received.

Signal Name Displays the name of the signals defined for the RX message.

Signal Value Displays the current signal value.

Visualization of different signal types

The RTI CAN MultiMessage Blockset lets you define different signal types, such as checksum signals or mode and mode-dependent signals. The following signal types can be identified by the color of the **Signal Name** entry in the instrument:

Color	Description
White	Indicates a checksum signal.
Violet	Indicates a mode signal.
Red	Indicates an inactive mode-dependent signal. The signal is currently not being transmitted.
Green	Indicates an active mode-dependent signal. The signal is being transmitted.

Tip

Mode and mode-dependent signals can also be identified by their node icons in the Bus Navigator tree. Refer to Bus Configuration Structure in the Bus Navigator Tree on page 19.

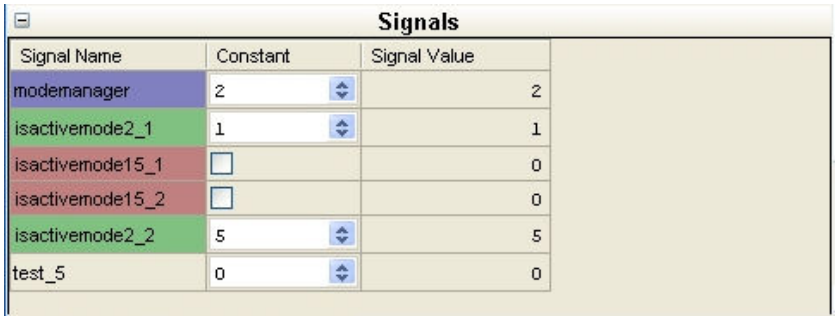
Mode and mode-dependent signals The value of the mode signal triggers the transmission of the mode-dependent signals of a message. Multiple signals can be transmitted via the same bytes of a message in this way.

Example: The *modemanager* mode signal triggers the transmission of the following mode-dependent signals:

- *isactivemode2_1*
- *isactivemode2_2*
- *isactivemode15_1*
- *isactivemode15_2*

The first two are specified for a *modemanager* value of 2. The latter two are specified for a *modemanager* value of 15.

If the *modemanager* mode signal has a value of 2, it activates the *isactivemode2_1* and *isactivemode2_2* signals. Both signals are transmitted on the bus. The *isactivemode15_1* and *isactivemode15_2* signals remain inactive.



Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

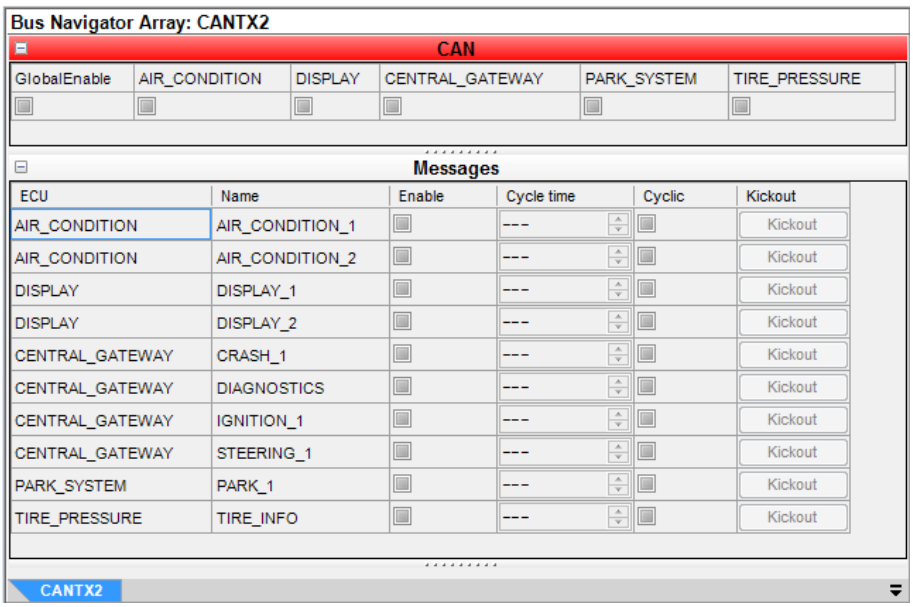
Related topics

References

Generate RX Layout..... 232

Bus Instrument (TX Status Type for CAN)

Illustration



Purpose

To specify whether TX messages/PDUs of a communication cluster are transmitted on the bus and to specify basic transmission options.

Description

You can create one or more Bus Instruments (TX status type for CAN) for each CAN communication cluster node in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate TX Status Layout** command. Refer to Generate TX Status Layout on page 235.

Instrument elements

The Bus Instrument (TX status type for CAN) displays all the ECUs of the selected communication cluster for which TX and TX/RX messages are specified. The instrument lets you enable ECUs and specify basic transmission options for the messages, such as cyclic transmission or transmission via kickout.

The header color of the instrument indicates whether the bus communication is active or inactive:

Header Color	Description
Red	Bus communication is inactive.
Green	Bus communication is active.

Cycle Time Lets you specify a cycle time. The message is transmitted cyclically on the bus according to the specified cycle time.

Cyclic Lets you transmit the message cyclically. Available if **TRC** or **BOTH** is selected on the **Message Cyclic** page of the RTICANMM MainBlock. Refer to Message Cyclic Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

ECU Displays the name of the ECUs defined for the communication cluster and lets you enable or disable the ECUs. If a checkbox is cleared, the relevant ECU is disabled and none of its messages are transmitted.

Enable Lets you enable a message.

GlobalEnable Lets you enable all the TX and RX messages of a CAN bus globally.

Kickout Lets you transmit the message once immediately. Available if **TRC** or **BOTH** is selected on the **Message Kickout** page of the RTICANMM MainBock. Refer to Message Kickout Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Name Displays the name of the TX message and lets you specify whether or not the message is transmitted. Available if **TRC** or **BOTH** is selected on the **Message Enable** page of the RTICANMM MainBlock. Refer to Message Enable Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

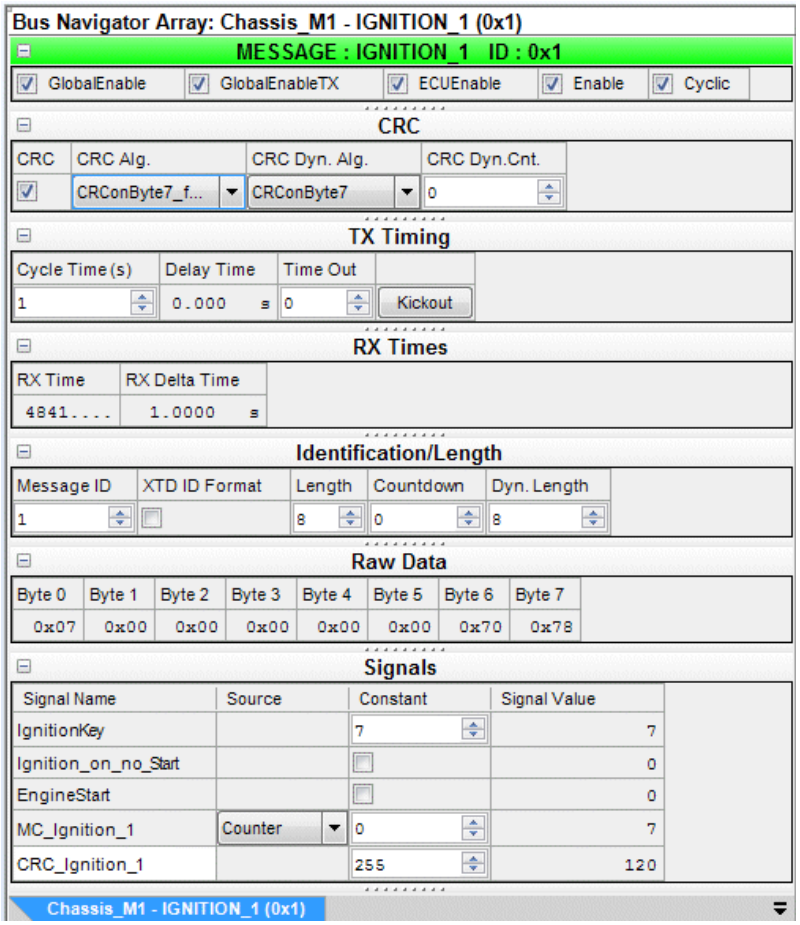
Related topics

References

Generate TX Status Layout..... 235

Bus Instrument (TX Type for CAN)

Illustration



Purpose

To display, configure, and transmit a TX message.

Description

You can create one or more Bus Instruments (TX type for CAN) for each TX and TX/RX message/PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate TX Layout** command.

Instrument elements

The Bus Instrument (TX type for CAN) lets you transmit TX messages on a CAN bus. The default transmission conditions are set via various options of the RTICANMM MainBlock. If enabled, you can manipulate several message- and signal-specific settings before the message is transmitted on the bus.

The Bus Instrument (TX type for CAN) is tailor-made, i.e., it displays only the message- and signal-specific settings which are enabled to be displayed and/or manipulated by ControlDesk during run-time.

The header color of the instrument indicates whether the message is active or inactive:

Header Color	Description
Red	The message is inactive.
Green	The message is active. However, a green header does not indicate that the message is being transmitted or received. The indicator for the transmission or reception of a message is a running RX time.

Base Time Lets you specify the base time (in milliseconds) for the selected message. Available if **Base/Update Time** is selected on the **Message** page of the RTICANMM MainBlock. Refer to Message Page (RTICANMM MainBlock Message View) (RTI CAN MultiMessage Blockset Reference 🕮).

Byte <number> Displays the raw data for each byte of a TX message. This display must be enabled on the **RX Raw Data Display** page of the RTICANMM MainBlock. Refer to RX Raw Data Display Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

If enabled, you can change the raw data of the bytes via ControlDesk during run time, refer to RAW bytes on page 152.

Constant Lets you specify a constant value for a signal or enable the transmission of a predefined signal constant value.

Countdown (dyn. length value) Lets you define the number of message transmissions the **Dyn. Length** value is to be transmitted for. Available if **Activate dynamic MessageLength** is selected on the **Manipulation Options** page of the RTICANMM MainBlock. Refer to Manipulation Options Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Countdown (signal) Lets you define the number of signal transmissions the **Dyn. Value** value is to be transmitted for. Available if a signal is selected on the **Dynamic Signal Values** page of the RTICANMM MainBlock. Refer to Dynamic Signal Values Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

CRC Lets you enable checksum calculation. Available if a signal is defined as a checksum signal on the **Checksum Messages** page of the RTICANMM

MainBlock. Refer to Checksum Messages Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

CRC Alg. Lets you select a checksum algorithm. The checksum is calculated and transmitted by the checksum signal according to the selected algorithm. Available if a signal is defined as a checksum signal on the **Checksum Messages** page of the RTICANMM MainBlock. Refer to Checksum Messages Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

CRC Dyn. Alg. / CRC Dyn. Cnt. Lets you select a checksum algorithm and specify the number of message transmissions the selected checksum algorithm is to be transmitted for. Available if a checksum signal is defined for the message and **Activate dynamic CRC** is selected on the **Manipulation Options** page of the RTICANMM MainBlock. Refer to Manipulation Options Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Cycle Time (s) Lets you specify a cycle time. The message is transmitted cyclically on the bus according to the specified cycle time.

Cyclic Lets you transmit the message cyclically. Available if **TRC** or **BOTH** is selected on the **Message Cyclic** page of the RTICANMM MainBlock. Refer to Message Cyclic Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Delay Time Displays the delay time of the message. The message is transmitted on the bus with this delay. By default, the delay time is view-only. You can edit the delay time if you select **Edit delay time online** on the **Cycle Time Defaults** page of the RTICANMM MainBlock. Refer to Cycle Time Defaults Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Dyn. Length Lets you manipulate the message length in the range 0 ... 8 bytes (CAN) or 0 ... 64 bytes (CAN FD) for a specified number of message transmissions. The number of message transmissions is defined via **Countdown** (message). Available if **Activate dynamic MessageLength** is selected on the **Manipulation Options** page of the RTICANMM MainBlock. Refer to Manipulation Options Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Dyn. Value Lets you specify a signal value that is to be transmitted for a specified number of signal transmissions. The number of signal transmissions is defined via **Countdown** (signal). Available if a signal is selected on the **Dynamic Signal Values** page of the RTICANMM MainBlock. Refer to Dynamic Signal Values Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

ECUEnable Lets you enable the transmitting ECU of the message. Available if **TRC** or **BOTH** is selected on the **Network Node Enable** page of the

RTICANMM MainBlock. Refer to Network Node Enable Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Enable Lets you enable the transmission of the message. Available if **TRC** or **BOTH** is selected on the **Message Enable** page of the RTICANMM MainBlock. Refer to Message Enable Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Error Displays the transmission of a signal's error value. If the LED is red, the error value is transmitted. Available if an error value is defined for a signal and signal errors are enabled on the **Signal Errors** page of the RTICANMM MainBlock. Refer to Signal Errors Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Gateway Lets you enable the exchange of messages between two CAN buses (gateway) via two RTICANMM MainBlocks. Available if gatewaying messages via two RTICANMM MainBlocks is enabled on the **Gateway Signals** page. Refer to Gateway Signals Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Gateway Source Lets you manipulate a received gateway signal before it is transmitted on the bus. You can select to transmit an error value, parity value, counter value, etc. The available value types depend on the value types specified on one of the following pages of the RTICANMM MainBlock:

- Counter Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Dynamic Signal Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Gateway Signals Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Parity Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Signal Default Manipulation Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Signal Errors Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Toggle Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Model Signals (TX) Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)

Available if **Enable Manipulation** is selected on the **Signal Default Manipulation** page of the RTICANMM MainBlock. Refer to Signal Default Manipulation Page (Gateway) (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

GlobalEnable Lets you enable all the TX and RX messages of a CAN controller globally. Available if **GlobalEnable_TRC** is selected on the **Triggering Options** page of the RTICANMM MainBlock. Refer to Triggering Options Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

GlobalEnableTX Lets you enable all the TX messages of a CAN controller globally. Available if **GlobalEnableTX_TRC** is selected on the **Triggering**

Options page of the RTICANMM MainBlock. Refer to Triggering Options Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

ID Format Lets you select the format for a free raw message. The selected format affects the number of raw bytes to be sent for the selected message. You can select the standard ID format (11 bit) or the extended ID format (29 bit) for a CAN or a CAN FD message and specify whether the bit rate switch (BRS) is set (CAN FD only). Refer to Free Raw Messages Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Kickout Lets you transmit the message once immediately. Available if TRC or BOTH is selected on the Message Kickout page of the RTICANMM MainBock. Refer to Message Kickout Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Length Lets you manipulate the message length in the range 0 ... 8 bytes (CAN) or 0 ... 64 bytes (CAN FD). Available if TRC is selected on the TX Message Length page of the RTICANMM MainBlock. Refer to TX Message Length Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Message ID Lets you specify the message ID in decimal format. You can modify the message ID if this is enabled on the TX ID page of the RTICANMM MainBlock. Refer to TX ID Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

J1939 DA (Available only for J1939 messages) Displays the address of the receiving ECU (destination ECU) of a J1939 peer-to-peer TX message. For broadcast messages, the global address (255) is displayed. You can modify the address of the destination ECU for peer-to-peer messages if Node Select via is set to ADDRESS and Source/Destination is set to TRC or BOTH (DEFAULT TRC) on the J1939 TX ID page of the RTICANMM MainBlock. Refer to J1939 TX ID Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

J1939 DN (Available only for J1939 peer-to-peer messages with node selection via NODE INDEX) Displays the ID of the receiving ECU (destination ECU) of a J1939 peer-to-peer TX message. You can modify the ID of the destination ECU if Source/Destination is set to TRC or BOTH (DEFAULT TRC) on the J1939 TX ID page of the RTICANMM MainBlock. Refer to J1939 TX ID Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

J1939 Prio (Available only for J1939 messages) Displays the priority of a J1939 TX message. You can modify the priority value if Priority is set to TRC or DEFAULT TRC on the J1939 TX ID page of the RTICANMM MainBlock. Refer to

J1939 TX ID Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

J1939 PGN (Available only for J1939 messages) Displays the parameter group number of a J1939 TX message.

Tip

The dSPACE J1939 PGN parameter is a 16-bit number in the 29-bit message identifier of a CAN message defined in a J1939-compliant DBC file. Each PGN references a parameter group that groups parameters and assigns them to the 8-byte data field of the message.

J1939 SA (Available only for J1939 messages) Displays the address of the transmitting ECU (source ECU) of a J1939 TX message. You can modify the address of the source ECU if **Node Select via** is set to **ADDRESS** and **Source/Destination** is set to **TRC** or **BOTH (DEFAULT TRC)** on the **J1939 TX ID** page of the RTICANMM MainBlock. Refer to J1939 TX ID Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

J1939 SN (Available only for J1939 messages) Displays the ID of the transmitting ECU (source ECU) of a J1939 TX message. You can modify the ID of the source ECU if **Source/Destination** is set to **TRC** or **BOTH (DEFAULT TRC)** on the **J1939 TX ID** page of the RTICANMM MainBlock. Refer to J1939 TX ID Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

RAW bytes Lets you enable the manipulation of the bytes of a message with raw data. If selected, you can use the **Byte <number>** edit fields to manipulate each byte of a message. Available if **TRC** and **Active** is selected on the **TX Raw Data** page of the RTICANMM MainBlock. Refer to TX Raw Data Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

RX Delta Time Displays the delta time between the current message transmission and the previous message transmission.

RX Time Displays the time the message is received on the CAN bus. The running RX time indicates that the message is received on the bus.

Signal Name Displays the names of the signals defined for the message.

Signal Value Displays the current signal value that is transmitted on the bus.

Source Lets you select a signal value type for transmitting a signal. You can select to transmit values such as error value, parity value, counter value, etc. The available value types depend on the value types specified on one of the following pages of the RTICANMM MainBlock:

- Counter Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Dynamic Signal Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Gateway Signals Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)

- Parity Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Signal Errors Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Toggle Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)
- Model Signals (TX) Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮)

Available if at least one value type is selected for a signal.

Timeout Lets you disable message transmission for a specified number of possible message transmissions. The specified value determines how often a TX message will not be transmitted even if transmission is triggered by the other trigger conditions. Available if **Enable_Timeout** is selected on the **Triggering Options** page of the RTICANMM MainBlock. Refer to Triggering Options Page (RTICANMM MainBlock) (RTI CAN MultiMessage Blockset Reference 🕮).

Update Number Lets you specify the number of transmissions. Available if **Base/Update Time** is selected on the **Message** page of the RTICANMM MainBlock. Refer to Message Page (RTICANMM MainBlock Message View) (RTI CAN MultiMessage Blockset Reference 🕮).

Update Time Lets you specify the update time (in milliseconds) for the selected message. Available if **Base/Update Time** is selected on the **Message** page of the RTICANMM MainBlock. Refer to Message Page (RTICANMM MainBlock Message View) (RTI CAN MultiMessage Blockset Reference 🕮).

Visualization of different signal types

The RTI CAN MultiMessage Blockset lets you define different signal types, such as checksum signals or mode and mode-dependent signals. The following signal types can be identified by the color of the **Signal Name** entry in the instrument:

Color	Description
White	Indicates a checksum signal.
Violet	Indicates a mode signal.
Red	Indicates an inactive mode-dependent signal. The signal is currently not being transmitted.
Green	Indicates an active mode-dependent signal. The signal is being transmitted.

Tip

Mode and mode-dependent signals can also be identified by their node icons in the Bus Navigator tree. Refer to Bus Configuration Structure in the Bus Navigator Tree on page 19.

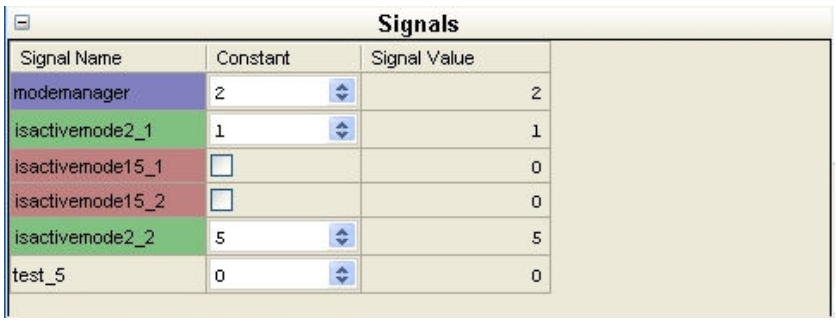
Mode and mode-dependent signals The value of the mode signal triggers the transmission of the mode-dependent signals of a message. Multiple signals can be transmitted via the same bytes of a message in this way.

Example: The *modemanager* mode signal triggers the transmission of the following mode-dependent signals:

- *isactivemode2_1*
- *isactivemode2_2*
- *isactivemode15_1*
- *isactivemode15_2*

The first two are specified for a *modemanager* value of 2. The latter two are specified for a *modemanager* value of 15.

If the *modemanager* mode signal has a value of 2, it activates the *isactivemode2_1* and *isactivemode2_2* signals. Both signals are transmitted on the bus. The *isactivemode15_1* and *isactivemode15_2* signals remain inactive.



Signals

Signal Name	Constant	Signal Value
modemanager	2	2
isactivemode2_1	1	1
isactivemode15_1	☐	0
isactivemode15_2	☐	0
isactivemode2_2	5	5
test_5	0	0

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

References

Generate TX Layout..... 234

FlexRay Instruments

Where to go from here

Information in this section

Bus Instrument (Error Status Type for FlexRay)..... 155
To display the error value and the slot status for all the PDUs of a FlexRay communication cluster.

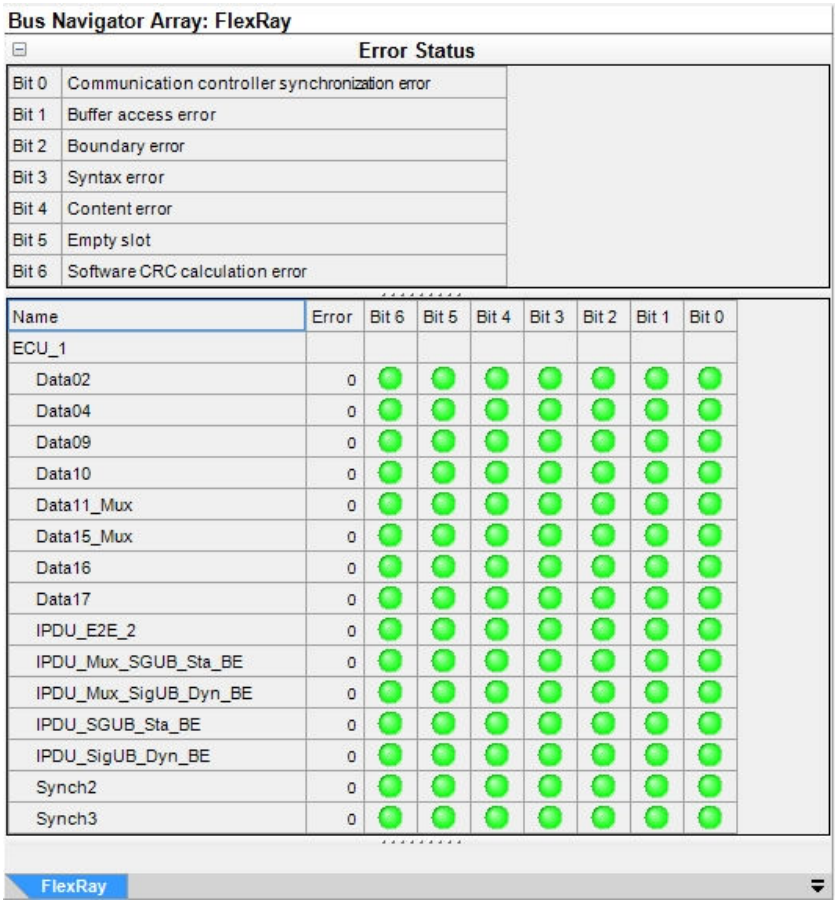
Bus Instrument (RX Type for FlexRay)..... 157
To display information on a single RX or TX/RX PDU.

Bus Instrument (TX Type for FlexRay)..... 160
To display, configure, and transmit a single TX or TX/RX PDU on a FlexRay bus.

Bus Instrument (TX Status Type for FlexRay)..... 166
To display and configure the transmission options of all the TX and TX/RX PDUs of a FlexRay communication cluster.

Bus Instrument (Error Status Type for FlexRay)

Illustration



Purpose

To display the error value and the slot status for all the PDUs of a FlexRay communication cluster.

Description

You can create one or more Bus Instruments (Error status type for FlexRay) for each FlexRay communication cluster node in the Bus Navigator tree via the **Generate Error Status Layout** command. The instrument opens in a separate layout.

The Bus Instrument (Error status type for FlexRay) displays all the RX and TX/RX PDUs that are specified for each ECU of the selected communication cluster.

For information on error values, refer to RTIFLEXRAYCONFIG PDU RX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 🕮) or, for SCALEXIO systems and MicroAutoBox III, to FLEXRAYCONFIG PDU RX Mapping Subsystems (FlexRay Configuration Blockset Reference 🕮).

Instrument elements

Bit 0…6 Displays a status LED for each error bit of the PDU.

Error Displays the error value of the PDU.

Name Displays the name of the PDU.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

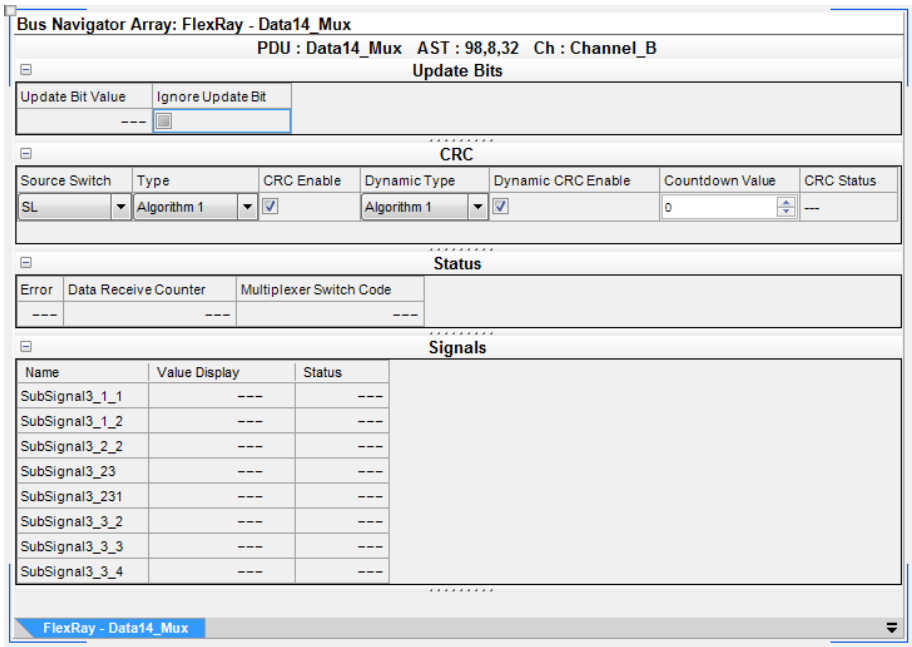
Related topics

References

Generate Error Status Layout..... 228

Bus Instrument (RX Type for FlexRay)

Illustration



Purpose

To display information on a single RX or TX/RX PDU.

Description

You can create one or more Bus Instruments (RX type for FlexRay) for each RX and TX/RX PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate RX Layout** command.

The Bus Instrument (RX type for FlexRay) displays information on RX and TX/RX PDUs. The displayed settings depend on the settings specified in the FlexRay bus configuration.

The Bus Instrument (RX type for FlexRay) supports signals that use text tables or a combination of a linear scaling and text tables. It displays converted values, i.e., alphanumerical values are used for linear scalings and text strings for text table scalings. For dSPACE platforms, text tables or a combination of a linear scaling and text tables must be enabled in the FlexRay Configuration tool. Refer to Generators Page (FlexRay Configuration Tool Reference 🕮).

Instrument elements

Information at the top of the instrument

- PDU: PDU name
- AST: Absolutely scheduled timing (slot ID, base cycle, cycle repetition)
- Ch: Channel name

All the PDU- and signal-specific settings that can be displayed in the instrument are listed in alphabetical order below.

Byte <number> Displays the raw data for each byte of the PDU.

CRC Enable Lets you enable or disable the CRC calculation of the PDU.

- 0: CRC calculation is disabled.
- 1: CRC calculation is enabled.

CRC Status Displays the status of the CRC calculation.

- 0: CRC calculation OK.
- 1: CRC calculation not OK.
- 2: CRC not calculated.

Data Receive Counter Displays the number of received PDUs. Null frames are ignored.

Dynamic CRC Enable Lets you enable manipulation of the checksum calculation of received PDUs for the number of receptions specified in `Countdown Value`. When `Countdown Value` reaches 0, checksum calculation returns to the previous mode. For more information, refer to Dynamic Checksum Calculation Manipulation (FlexRay Configuration Features 🕮).

Dynamic Type Lets you select the algorithm that is used for CRC calculation when the `Dynamic CRC Enable` property is set. The CRC algorithms are specified in the FlexRay Configuration Tool. For more information, refer to Using User-Defined Checksum Algorithms (FlexRay Configuration Tool Guide 🕮).

Error Displays error statuses. For information on possible values, refer to RTIFLEXRAYCONFIG PDU RX Mapping Subsystems (RTI FlexRay Configuration Blockset Reference 🕮).

Ignore Update Bit Displays whether the `Ignore Update Bit` property is active.

Multiplexer Switch Code Displays the switch code that was used for a received PDU (only for dynamic PDUs containing sub-PDUs)

Nullframe Displays whether the received frame was a null frame (only for static PDUs).

- 0: The last received frame was not a null frame.
- 1: The last received frame was a null frame.

Payload Length Value Displays the payload length of the RX raw data (only available if the `Payload length` property of the PDU is enabled. Refer to Configurable Properties for ECUs, Signals, and Frames (FlexRay Configuration Tool Reference 🕮)).

Physical Value Displays the physical value of the received signal (if conversion was possible). Available only if both of the following conditions are met:

- Signal Value Access is selected in the Element Selection dialog of the FlexRay Configuration Tool. Refer to Element Selection (FlexRay Configuration Tool Reference 🕮).
- Use physical data type is enabled for the Trace File Generator in the Code Generation dialog of the FlexRay Configuration Tool. Refer to Generators Page (FlexRay Configuration Tool Reference 🕮).

Source Switch Lets you switch the data source:

- SL: The values of the Simulink model are sent.
- TRC: Data specified via the trace variable are sent. If CRC is enabled, values of the `Type` and `CRC Enable` variables are used.
- TRC Dynamic: The values of the `Dynamic Type` and `Dynamic CRC Enable` trace variables are sent for the number of times specified in `Countdown Value`. When `Countdown Value` reaches 0, the `Switch` variable is automatically switched back to the switch state that was set before.

Status (signal) Displays status information on the received signal.

Available only if Signal RX Status Access is selected in the Element Selection dialog of the FlexRay Configuration Tool. Refer to Element Selection (FlexRay Configuration Tool Reference 🕮).

Type Lets you select the algorithm that is used for CRC calculation. The CRC algorithms are specified in the FlexRay Configuration Tool. For more information, refer to Using User-Defined Checksum Algorithms (FlexRay Configuration Tool Guide 🕮).

Update Bit Value Displays the value of the received update bit.

Value Display (signal) Displays the value of the received or monitored signal.

Available only if Signal Value Access is selected in the Element Selection dialog of the FlexRay Configuration Tool. Refer to Element Selection (FlexRay Configuration Tool Reference 🕮).

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

HowTos

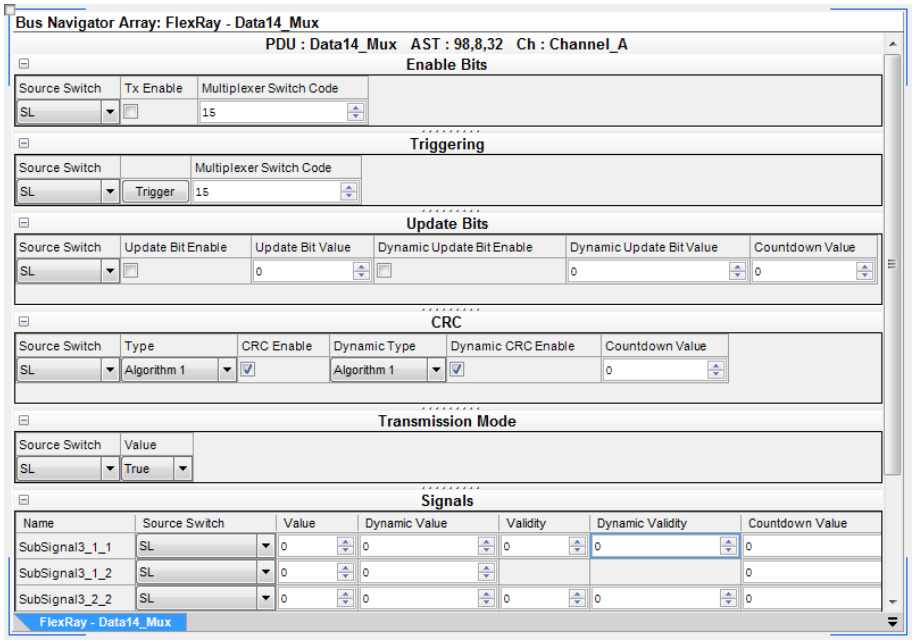
How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs..... 50

References

Generate RX Layout..... 232

Bus Instrument (TX Type for FlexRay)

Illustration



Purpose

To display, configure, and transmit a single TX or TX/RX PDU on a FlexRay bus.

Description

You can create one or more Bus Instruments (TX type for FlexRay) for each TX and TX/RX PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate TX Layout** command. Refer to Generate TX Layout on page 234.

The Bus Instrument (TX type for FlexRay) lets you transmit a single PDU on a FlexRay bus. The default transmission conditions are set via the FlexRay configuration. Depending on the settings of the FlexRay configuration, the Bus

Instrument (TX type for FlexRay) lets you display and manipulate several PDU-specific and signal-specific settings before the PDU is transmitted on the bus.

For dSPACE platforms, the Bus Instrument (TX type for FlexRay) supports signals that use text tables. The Bus Instrument lets you enter the text table values numerically or as text strings by selecting them from a list. Refer to Generators Page (FlexRay Configuration Tool Reference 🕮).

Instrument elements

Information at top of the instrument

- PDU: PDU name
- AST: Absolutely scheduled timing (slot ID, base cycle, cycle repetition)
- CH: Channel name

AC Enabled Lets you enable or disable the alive counter.

AC Offset Lets you specify an offset value that is added to the alive counter.

AC Runtime Behavior Lets you specify the behavior of the alive counter at run time:

- 0: The alive counter runs independently of the selected source for the signal.
- 1: The alive counter runs only if its value is really sent. If the source for the signal is switched to another source than the alive counter, for example, to SL (a value from the Simulink model), the alive counter stops.
- 2: The alive counter stops at the current value.

For more information, refer to Using an Alive Counter in the Simulation (FlexRay Configuration Features 🕮).

AC Value Displays the current value of the alive counter.

Byte <number> Displays the raw data for each byte of a TX PDU.

Countdown Value Lets you specify the time span for the dynamic signal manipulation, i.e., how often the dynamic values are sent. For more information on dynamic signal manipulation, refer to Dynamic Signal Manipulation (FlexRay Configuration Features 🕮).

CRC Enable Lets you enable or disable the CRC calculation of the PDU.

Dynamic CRC Enable Lets you enable manipulation of the checksum calculation of TX PDUs for the number of transmissions specified in `Countdown Value`. When `Countdown Value` reaches 0, checksum calculation returns to the previous mode. For more information, refer to Dynamic Checksum Calculation Manipulation (FlexRay Configuration Features 🕮).

Dynamic Physical Value Lets you specify the value to be sent during dynamic signal manipulation as a physical data type. Only available if both of the following conditions are met:

- Signal Value Access has been selected in the Element Selection dialog of the FlexRay Configuration Tool. Refer to Element Selection (FlexRay Configuration Tool Reference 🕮).

- **Use physical data type** has been enabled for the Trace File Generator in the **Code Generation** dialog of the FlexRay Configuration Tool. Refer to Generators Page (FlexRay Configuration Tool Reference 🕮).

For more information on dynamic signal manipulation, refer to Dynamic Signal Manipulation (FlexRay Configuration Features 🕮).

Dynamic Type Lets you select the algorithm that is used for CRC calculation during dynamic CRC manipulation. The CRC algorithms are specified in the FlexRay Configuration Tool. For more information, refer to Using User-Defined Checksum Algorithms (FlexRay Configuration Tool Guide 🕮).

Dynamic Update Bit Enable Lets you enable manipulation of the update bit of a TX PDU for a specified number of transmissions. After the manipulation, the update bit has the value of the data source that was selected before. For more information, refer to Dynamic Update Bit Manipulation (FlexRay Configuration Features 🕮).

Dynamic Update Bit Value Lets you specify the update bit value to be sent during dynamic update bit manipulation.

Dynamic Validity Lets you specify validity information on the signal to be sent during dynamic signal manipulation.

The following validity values are possible:

- 0: NOT VALID
- 1: VALID
- 2: ERROR
- 4: NOT AVAILABLE
- 8: NOT DEFINED
- 16: OTHER

Note

You cannot select the bitfield values of the **Combined Status** or **Status** variable mentioned in the FlexRay Configuration Tool Guide. For more information, refer to Basics of Code Generation (FlexRay Configuration Tool Guide 🕮).

The variable is generated if an invalid value (coded and/or physical) exists for the respective signal.

Available if **Signal Validity Control** has been selected in the **Element Selection** dialog of the FlexRay Configuration Tool. Refer to Element Selection (FlexRay Configuration Tool Reference 🕮).

For more information on dynamic signal manipulation, refer to Dynamic Signal Manipulation (FlexRay Configuration Features 🕮).

Dynamic Value Lets you specify the value to be sent during dynamic signal manipulation as a physical data type.

Available only if **Signal RX Status Access** has been selected in the **Element Selection** dialog of the FlexRay Configuration Tool. Refer to Element Selection (FlexRay Configuration Tool Reference 🕮).

For more information on dynamic signal manipulation, refer to Dynamic Signal Manipulation (FlexRay Configuration Features 🕮).

MDT Enable Lets you enable or disable the minimum delay time support of the PDU.

- 0: Minimum delay time support is disabled.
- 1: Minimum delay time support is enabled.

The minimum delay time of a PDU, which is defined in the AUTOSAR system description file, specifies the minimum delay time between successive transmissions of the new PDU data. **MDT Enable** is available only for PDUs for which minimum delay time support is enabled in the FlexRay Configuration Tool. Refer to How to Configure PDUs for Minimum Delay Time Support (FlexRay Configuration Tool Guide 🕮).

Multiplexer Switch Code (Available only for dynamic PDUs that contain sub-PDUs.) Lets you select the switch code of a sub-PDU. You can only enter valid values that are documented in the variable description. The switch codes in the description are different for event and cyclic sub-PDUs.

Physical Value Lets you specify the value to be sent as a physical data type. Available only if both of the following conditions are met:

- **Signal Value Access** has been selected in the **Element Selection** dialog of the FlexRay Configuration Tool. Refer to Element Selection (FlexRay Configuration Tool Reference 🕮).
- **Use physical data type** has been enabled for the Trace File Generator in the **Code Generation** dialog of the FlexRay Configuration Tool. Refer to Generators Page (FlexRay Configuration Tool Reference 🕮).

Raw Data Enable Lets you specify the raw data which is sent:

- Activated: Raw data specified in the Simulink model is sent. If it is not available, signal data specified using the TRC file (if the `Source Switch` variable is TRC) or signal data specified in the Simulink model (if the `Source Switch` variable is SL) is sent. If signal data is not available, the data last sent or the initial values are sent.
- Not activated: Raw data specified by the `Tx Bytes` variable is sent.

  The setting is effective only if the `Source Switch` variable is not SL.

Source Switch Lets you switch the data source:

- SL: The values of the Simulink model are sent.
- TRC: Data specified via the trace variable is sent.
  - If update bits are enabled, the values of the `Update Bit Enable` variables are used.
  - If CRC is enabled, the values of the `Type` and `CRC Enable` variables are used.
- TRC Dynamic:
  - If dynamic update bits are enabled, the values of the `Dynamic Value` and `Dynamic Update Bit Enable` variables are sent for the number of times specified in `Countdown Value`. When `Countdown Value` reaches 0, the

`Source Switch` variable is automatically switched back to the switch state that was set before.

- If dynamic CRC is enabled, the values of the `Dynamic Type` and `Dynamic CRC Enable` variables are sent for the number of times specified in `Countdown Value`. When `Countdown Value` reaches 0, the `Source Switch` variable is automatically switched back to the switch state that was set before.

Status (Available only if minimum delay time support is enabled for the PDU.) Displays information on the last captured trigger and on whether and how the trigger was considered for triggering.

Value	Meaning	Description
0	No recent trigger.	–
1	Accepted event trigger.	The event-based triggering occurred outside of an active minimum delay time.
2	Accepted cyclic trigger.	The cyclic triggering occurred outside of an active minimum delay time.
3	Accepted event trigger, discarded cyclic trigger.	The simultaneous event-based and cyclic triggering occurred outside of an active minimum delay time.
4	Discarded event trigger.	The event-based triggering occurred during the active minimum delay time.
8	Discarded cyclic trigger.	The cyclic triggering occurred during the active minimum delay time.
12	Discarded event trigger, discarded cyclic trigger.	The simultaneous event-based and cyclic triggering occurred during the active minimum delay time.

Refer to How to Configure PDUs for Minimum Delay Time Support (FlexRay Configuration Tool Guide 🕮).

SW Enable (Available only for static PDUs.) Lets you enable the transmission of static PDUs.

If a static PDU is disabled via software, old data or null frames can be sent. The data that is sent depends on the database version, on the SWEnable Configuration property (refer to General Page (FlexRay Configuration Tool Reference 🕮)), and on settings of the CHI Code Generator (refer to Generators Page (FlexRay Configuration Tool Reference 🕮)):

- FIBEX version ≤ 2.0:

  Null frames are sent if the **Static TX buffer transmission mode** property of the CHI Code Generator is set to **Event (null frame used)**.

  Old data is sent if the **Static TX buffer transmission mode** property is set to **State (old value used)**.

- FIBEX+, FIBEX 3.0, FIBEX 3.1, FIBEX 4.1.x, and AUTOSAR System Template:

  The following applies if the **SW Enable Configuration** property in the **General Properties** dialog is set to **Control of L-PDU commit to FlexRay buffer**:

  - Null frames are sent if the **Static TX buffer transmission mode** property of the CHI Code Generator is set to **Event (null frame used)**.
  - Old data is sent if the **Static TX buffer transmission mode** property is set to **State (old value used)**. The update bit has the value that was set before SWEnable was set to 0.

The following applies if the **SW Enable Configuration** property in the **General Properties** dialog is set to **Control of I-PDU payload data update**:

- The update bit is set to 0 and old data is sent, independently of the **Static TX buffer transmission mode** property settings.

  If the **Static TX buffer transmission mode** property is set to **Event (null frame used)**, keep the following information in mind:

  In some cases, the **SW Enable** setting is overruled. This can happen if the update bit of a PDU is manipulable: If **Update Bit Enable** is set, automatic calculation of the update bit of a PDU is disabled. Instead, the update bit of the PDU is set to the value specified by **Update Bit Value**. If the update bit value is 1, the PDU still sends old data. If **Update Bit Enable** is not set, **SW Enable** is used to enable or disable the transmission of static TX PDUs.

Time (s) (Available only if the minimum delay time support is enabled for the PDU.) Displays the minimum delay time that is specified for the PDU. Refer to How to Configure PDUs for Minimum Delay Time Support (FlexRay Configuration Tool Guide 🕮).

Transmission Mode - Value Lets you specify the transmission mode value.

A FlexRay configuration can offer different transmission modes for each PDU if the underlying communication matrix contains corresponding timing information. You can select the values **False**, **True**, **User defined**, and **LPDU timing triggered**.

The result of the selection depends on the underlying FIBEX or AUTOSAR communication cluster files. For more information, refer to How to Configure PDU Transmission Modes (FlexRay Configuration Tool Guide 🕮).

Trigger Lets you trigger an event PDU.

Tx Enable (Available only for cyclic PDUs.) Lets you enable the sending of cyclic PDUs.

Type Lets you select the algorithm that is used for CRC calculation. The CRC algorithms are specified in the FlexRay Configuration Tool. For more information, refer to Using User-Defined Checksum Algorithms (FlexRay Configuration Tool Guide 🕮).

Update Bit Enable Lets you enable or disable automatic calculation of the update bit in the FTCom task when **Source Switch** is set to TRC.

- Selected: The value specified by **Update Bit Value** is used. Automatic calculation is disabled.
- Cleared: Automatic calculation is enabled. This overwrites the value specified by **Update Bit Value**.

Update Bit Value Lets you specify the value of the update bit.

Validity Lets you specify validity information of the signal to be sent.

The variable is generated if an invalid value (coded and/or physical) exists for the respective signal.

The variable is essential only if data to be sent is specified in ControlDesk (**Source Switch** = TRC).

Available if **Signal Validity Control** has been selected in the **Element Selection** dialog of the FlexRay Configuration Tool. Refer to Element Selection (FlexRay Configuration Tool Reference 🕮).

Value Lets you specify the value to be sent.

Available only if **Signal RX Status Access** has been selected in the **Element Selection** dialog of the FlexRay Configuration Tool. Refer to Element Selection (FlexRay Configuration Tool Reference 🕮).

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

HowTos

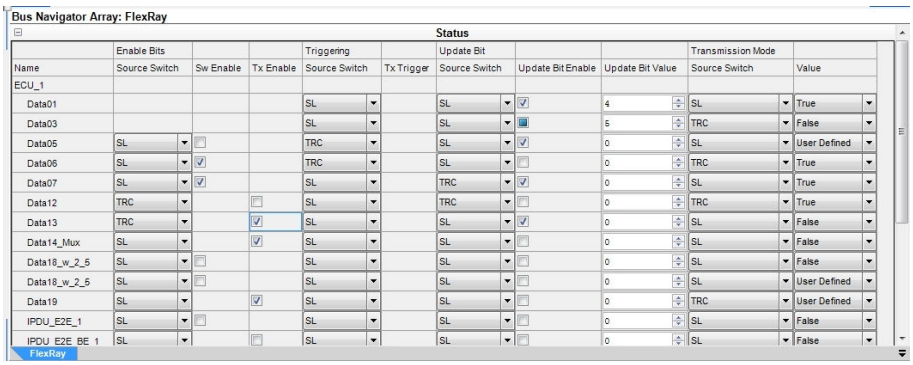
How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs..... 50

References

Generate TX Layout..... 234

Bus Instrument (TX Status Type for FlexRay)

Illustration



Purpose

To display and configure the transmission options of all the TX and TX/RX PDUs of a FlexRay communication cluster.

Description

You can create one or more Bus Instruments (TX status type for FlexRay) for each FlexRay communication cluster node in the Bus Navigator tree via the **Generate TX Status Layout** command. The instrument opens in a separate layout.

The Bus Instrument (TX status type for FlexRay) displays all the RX and TX/RX PDUs that are specified for each ECU of the selected communication cluster. The instrument lets you specify basic transmission options for the PDUs, such as enabling and disabling the sending of PDUs or disabling the automatic calculation of update bit values.

Instrument elements

Source Switch Lets you switch the data source:

- SL: The values of the Simulink model are sent.
- TRC: Data specified via the trace variable is sent.
  - If update bits are enabled, the values of the `Update Bit Enable` variables are used.
  - If CRC is enabled, the values of the `Type` and `CRC Enable` variables are used.
- TRC Dynamic:
  - If dynamic update bits are enabled, the values of the `Dynamic Value` and `Dynamic Update Bit Enable` variables are sent for the number of times specified in `Countdown Value`. When `Countdown Value` reaches 0, the `Source Switch` variable is automatically switched back to the switch state that was set before.
  - If dynamic CRC is enabled, the values of the `Dynamic Type` and `Dynamic CRC Enable` variables are sent for the number of times specified in `Countdown Value`. When `Countdown Value` reaches 0, the `Source Switch` variable is automatically switched back to the switch state that was set before.

SW Enable (Available only for static PDUs.) Lets you enable the transmission of static PDUs.

If a static PDU is disabled via software, old data or null frames can be sent. The data that is sent depends on the database version, on the SWEnable Configuration property (refer to General Page (FlexRay Configuration Tool Reference 🕮)), and on settings of the CHI Code Generator (refer to Generators Page (FlexRay Configuration Tool Reference 🕮)):

- FIBEX version ≤ 2.0:

  Null frames are sent if the **Static TX buffer transmission mode** property of the CHI Code Generator is set to **Event (null frame used)**.

  Old data is sent if the **Static TX buffer transmission mode** property is set to **State (old value used)**.
- FIBEX+, FIBEX 3.0, FIBEX 3.1, FIBEX 4.1.x, and AUTOSAR System Template:

  The following applies if the **SW Enable Configuration** property in the **General Properties** dialog is set to **Control of L-PDU commit to FlexRay buffer**:
  - Null frames are sent if the **Static TX buffer transmission mode** property of the CHI Code Generator is set to **Event (null frame used)**.

- Old data is sent if the **Static TX buffer transmission mode** property is set to **State (old value used)**. The update bit has the value that was set before SWEnable was set to 0.

The following applies if the **SW Enable Configuration** property in the **General Properties** dialog is set to **Control of I-PDU payload data update**:

- The update bit is set to 0 and old data is sent, independently of the **Static TX buffer transmission mode** property settings.

  If the **Static TX buffer transmission mode** property is set to **Event (null frame used)**, keep the following information in mind:

  In some cases, the **SW Enable** setting is overruled. This can happen if the update bit of a PDU is manipulable: If **Update Bit Enable** is set, automatic calculation of the update bit of a PDU is disabled. Instead, the update bit of the PDU is set to the value specified by **Update Bit Value**. If the update bit value is 1, the PDU still sends old data. If **Update Bit Enable** is not set, **SW Enable** is used to enable or disable the transmission of static TX PDUs.

Transmission Mode - Value Lets you specify the transmission mode value.

A FlexRay configuration can offer different transmission modes for each PDU if the underlying communication matrix contains corresponding timing information. You can select the values **False**, **True**, **User defined**, and **LPDU timing triggered**.

The result of the selection depends on the underlying FIBEX or AUTOSAR communication cluster files. For more information, refer to How to Configure PDU Transmission Modes (FlexRay Configuration Tool Guide 🕮).

Trigger Lets you trigger an event PDU.

Tx Enable (Available only for cyclic PDUs.) Lets you enable the sending of cyclic PDUs.

Update Bit Enable Lets you enable or disable automatic calculation of the update bit in the FTCom task when **Source Switch** is set to TRC.

- Selected: The value specified by **Update Bit Value** is used. Automatic calculation is disabled.
- Cleared: Automatic calculation is enabled. This overwrites the value specified by **Update Bit Value**.

Update Bit Value Lets you specify the value of the update bit.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

References

Generate TX Status Layout..... 235

LIN Instruments

Where to go from here

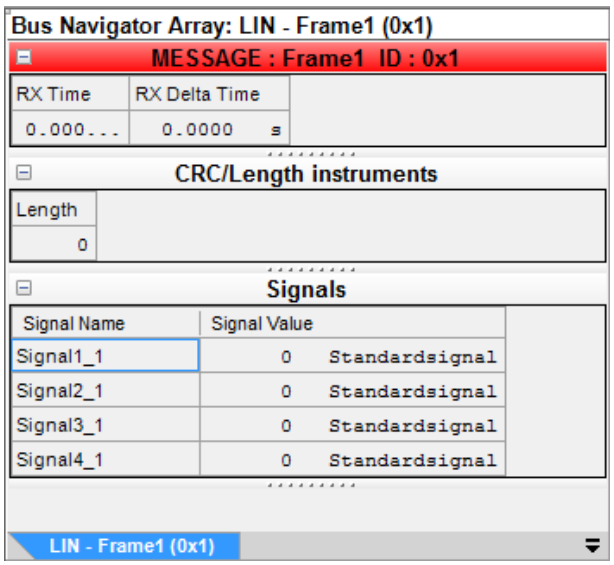
Information in this section

Bus Instrument (RX Type for LIN)..... 169
To display information on an RX message.

Bus Instrument (TX Type for LIN)..... 171
To display, configure, and transmit a TX message.

Bus Instrument (RX Type for LIN)

Illustration



Purpose

To display information on an RX message.

Description

You can create one or more Bus Instruments (RX type for LIN) for each RX and TX/RX frame/PDU in the Bus Navigator tree.The instrument opens in a separate layout when you create it via the Generate RX Layout command.

Instrument elements

The Bus Instrument (RX type for LIN) displays information on received frames (RX frames). The displayed information is view-only. The instrument content is tailor-

made, i.e., the displayed settings depend on the settings specified in the RTILINMM MainBlock and on the frame type.

Byte <number> Displays the raw data for each byte of an RX frame. This display must be enabled on the **RX Raw Data Display** page of the RTILINMM MainSetup block. Refer to TX Raw Data Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

The checksum value of the frame can be displayed to the right of the byte fields. This display must be enabled on the **RX Frames Checksum Display** page of the RTILINMM MainSetup block. Refer to RX Frames Checksum Display Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

Length Displays the number of bytes defined for the RX frame.

Length Setpoint Lets you specify the length a frame must have to be decoded. A received frame is decoded if its length corresponds to the **Length setpoint** value. Available if **Activate dynamic frame length** is selected on the **Frame length** page of the RTILINMM MainSetup block. Refer to Frame Length Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

If the frame is configured as a TX/RX frame, the **Length setpoint** setting is also used for transmission.

Message ID Displays the ID of an RX frame.

RX Delta Time Displays the delta time between the current and previous reception of the RX frame.

RX Time Displays the time the RX frame was received.

Signal Name Displays the name of the signals defined for the RX frame.

Signal Value Displays the current signal value.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

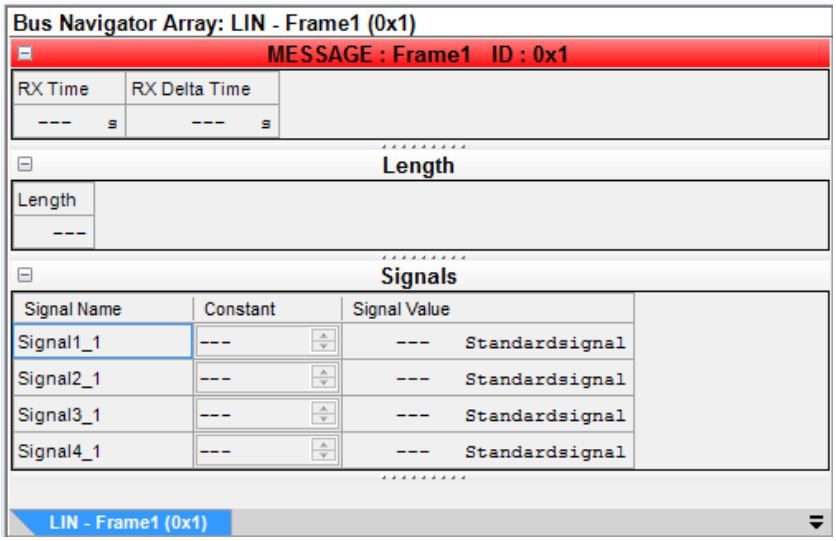
Related topics

References

Generate RX Layout..... 232

Bus Instrument (TX Type for LIN)

Illustration



Purpose

To display, configure, and transmit a TX message.

Description

You can create one or more Bus Instruments (TX type for LIN) for each TX and TX/RX frame/PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate TX Layout** command.

Instrument elements

The Bus Instrument (TX type for LIN) lets you transmit TX frames on a LIN bus. The default transmission conditions are set via various options of the RTILINMM MainBlock. If enabled, you can manipulate several frame- and signal-specific settings before the frame is transmitted on the bus.

The Bus Instrument (TX type for LIN) is tailor-made, i.e., it displays only the frame- and signal-specific settings which are enabled to be displayed and/or manipulated by ControlDesk during run time.

Byte <number> Displays the raw data for each byte of a TX frame. This display must be enabled on the **RX Raw Data Display** page of the RTILINMM MainSetup block. Refer to RX Raw Data Display Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

If enabled, you can change the raw data of the bytes via ControlDesk during run time, refer to RAW bytes on page 173.

Checksum Manipulation Lets you specify the countdown value in the range 0 ... 255. The countdown value defines the number of times the checksum value is manipulated by adding the offset value. Available if a checksum signal is defined for the frame and **Enable LIN frame checksum manipulation** is activated on the **Dynamic LIN Frame Checksum** page of the RTILINMM

MainSetup block. Refer to Dynamic LIN Frame Checksum Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

Constant Lets you specify a constant value for a signal or enable the transmission of a predefined signal constant value.

Countdown (dyn. length value) Lets you define the number of frame transmissions the **Dyn. Length** value is to be transmitted for. Available if **Activate dynamic frame length** is selected on the **Frame length** page of the RTILINMM MainSetup block. Refer to Frame Length Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

Countdown (signal) Lets you define the number of signal transmissions the **Dyn. Value** value is to be transmitted for. Available if a signal is selected on the **Dynamic Signal Values** page of the RTILINMM MainSetup block. Refer to Dynamic Signal Values Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

CRC Lets you enable checksum calculation. Available if a signal is defined as a checksum signal on the **User Checksum Frames** page of the RTILINMM MainSetup block. Refer to User Checksum Frames Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

CRC Alg. Lets you select a checksum algorithm. The checksum is calculated and transmitted by the checksum signal according to the selected algorithm. Available if a signal is defined as a checksum signal on the **User Checksum Frames** page of the RTILINMM MainSetup block. Refer to User Checksum Frames Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

CRC Dyn. Alg. / CRC Dyn. Cnt. Lets you select a checksum algorithm and specify the number of frame transmissions the selected checksum algorithm is to be transmitted for. Available if a checksum signal is defined for the frame and **Dynamic checksum** is selected on the **User Checksum Frames** page of the RTILINMM MainSetup block. Refer to User Checksum Frames Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

Dyn. Length Lets you manipulate the frame length in the range 0 ... 8 bytes for a specified number of frame transmissions. The number of frame transmissions is defined via **Countdown** (frame). Available if **Activate dynamic frame length** is selected on the **Frame length** page of the RTILINMM MainSetup block. Refer to Frame Length Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

Dyn. Value Lets you specify a signal value that is to be transmitted for a specified number of signal transmissions. The number of signal transmissions is defined via **Countdown** (signal). Available if a signal is selected on the **Dynamic Signal Values** page of the RTILINMM MainSetup block. Refer to Dynamic Signal

Values Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

Enable Lets you enable the transmission of the frame. Available if **TRC** or **BOTH** is selected on the **Enable Frames** page of the RTILINMM MainSetup block. Refer to Enable Frames Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

Error Displays the transmission of a signal's error value. If the LED is red, the error value is transmitted. Available if an error value is defined for a signal and signal errors are enabled on the **Signal Errors** page of the RTILINMM MainSetup block. Refer to Signal Errors Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

Length Lets you manipulate the frame length in the range 0 ... 8 bytes. Available if **TRC** is selected on the **Frame Length** page of the RTILINMM MainSetup block. Refer to Frame Length Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

Network Node Name Lets you enable the transmitting node of the frame. Available if **TRC** or **BOTH** is selected on the **Network Node Enable** page of the RTILINMM MainSetup block. Refer to Network Node Enable Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

RAW bytes Lets you enable the manipulation of the bytes of a frame with raw data. If selected, you can use the **Byte <number>** edit fields to manipulate each byte of a frame. Available if **TRC** and **Active** is selected on the **TX Raw Data** page of the RTILINMM MainSetup block. Refer to TX Raw Data Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮).

Signal Name Displays the names of the signals defined for the frame.

Signal Value Displays the current signal value that is transmitted on the bus.

Source Lets you select a signal value type for transmitting a signal. You can select to transmit values such as error value, counter value, etc. The available value types depend on the value types specified on one of the following pages of the RTILINMM MainSetup block:

Signal Value Type	Refer to
Counter	Counter Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮)
▪ DynVal: The countdown value is transmitted (raw signal). ▪ DynValConst: The value of the dynamic signal is transmitted.	Dynamic Signal Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮)
Input	Input Manipulation Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮)
Error	Signal Errors Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮)
Constant	Model Signals (TX) Page (RTILINMM MainSetup) (RTI LIN MultiMessage Blockset Reference 🕮)

Available if at least one value type is selected for a signal.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

References

Generate TX Layout..... 234

Bus Instruments (Related to Bus Monitoring Devices)

Where to go from here

Information in this section

CAN Instruments..... 174

Ethernet Instruments..... 178

LIN Instruments..... 180

CAN Instruments

Where to go from here

Information in this section

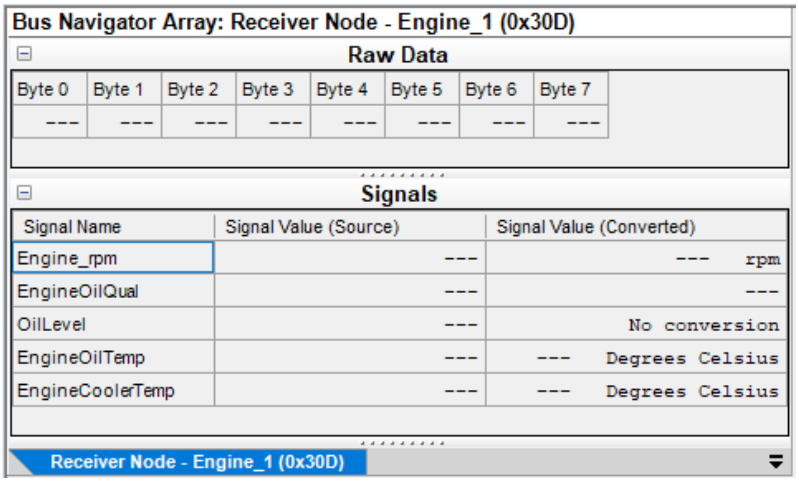
Bus Instrument (RX Type for CAN)..... 175
To display information on an RX PDU.

Bus Instrument (TX Type for CAN)..... 176
To display, configure, and transmit a TX message/PDU on a CAN bus.

Bus Instrument (RX Type for CAN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To display information on an RX PDU.

Description

You can create one or more Bus Instruments (RX type for CAN) for each RX PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the Generate RX Layout command.

Instrument elements

The Bus Instrument (RX type for CAN) lets you display information on the signals that are received via RX PDUs.

The instrument contains the following elements:

Signal Name Displays the name of the signal.

Signal Value (Source) Displays the source, i.e., the unconverted signal value.

Signal Value (Converted) Displays the signal value after it has been converted, for example, into a physical quantity according to a computation method.

You can specify to display each signal of an RX PDU or RX multiplexed PDU in the Bus Instrument (RX type for CAN), regardless of whether the signal is actually sent or received by an ECU. Refer to Include all signals in RX layouts on the Bus Navigator Page on page 207.

The Raw Data region is displayed only if you selected Import Options – Bus Import Options – Generate raw byte variables for each CAN message on the Variables page of the ControlDesk Options dialog. The region displays the raw data for each byte of an RX message.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

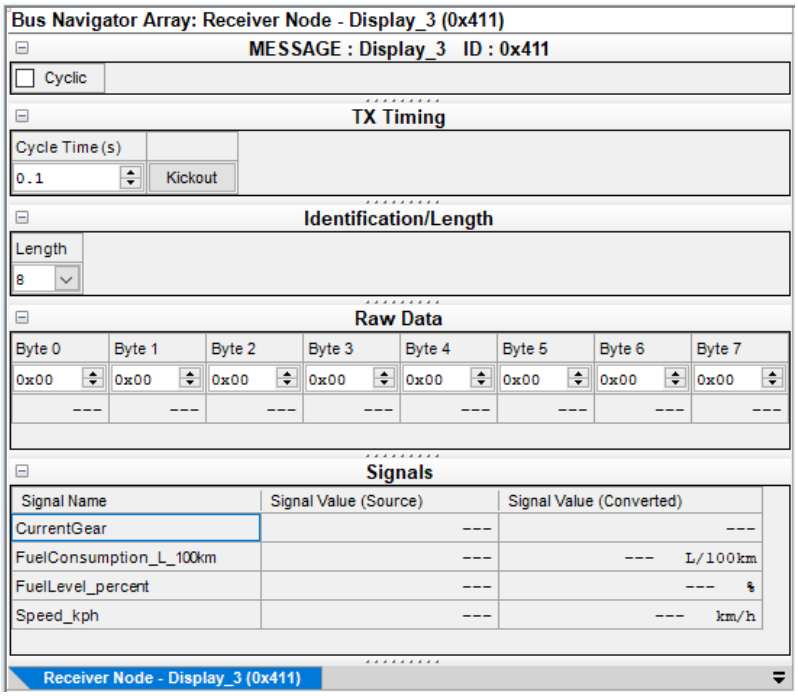
References

Generate RX Layout..... 232

Bus Instrument (TX Type for CAN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To display, configure, and transmit a TX message/PDU on a CAN bus.

Description

You can create one or more Bus Instruments (TX type for CAN) for each TX message/PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the **Generate TX Layout** command.

Instrument elements

The Bus Instrument (TX type for CAN) lets you display and manipulate the signals that are transmitted via TX PDUs.

The instrument contains the following elements:

Signal Name Displays the name of the signal.

Signal Value (Source) Displays the source, i.e., the unconverted signal value.

Signal Value (Converted) Displays the signal value after it has been converted, for example, to a physical quantity according to a computation method.

The Raw Data region is displayed only if you selected Import Options – Bus Import Options – Generate raw byte variables for each CAN message on the Variables page of the ControlDesk Options dialog. It lets you manipulate each byte of a TX message.

Depending on the variable description, the following regions and elements can also be displayed:

Cyclic Lets you select cyclic transmission of the message.

TX Timing Lets you configure transmission options:

- Cycle Time (s): Lets you specify a Cycle Time in seconds. The message is transmitted cyclically on the bus according to the specified cycle time.
- Kickout: Lets you transmit the message once immediately.

Identification/Length Lets you manipulate the message length in the range 0 ... 8 bytes (CAN) or 0 ... 64 bytes (CAN FD).

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

References

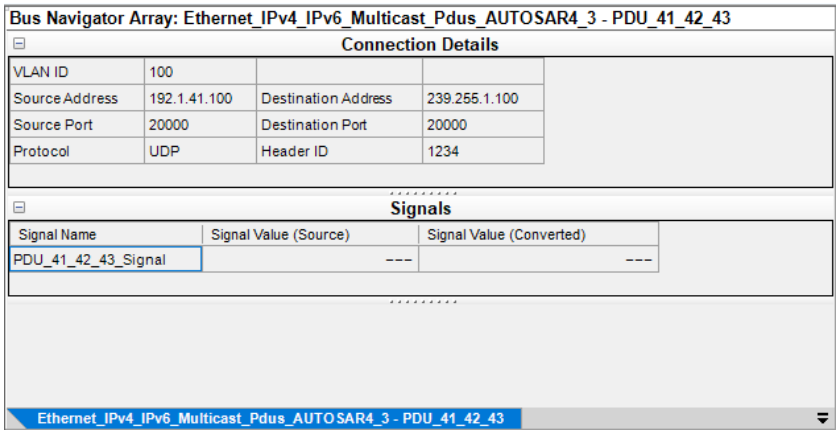
Generate TX Layout..... 234

Ethernet Instruments

Bus Instrument (RX Type for Ethernet)

Illustration

The following illustration shows an example of the instrument:



Purpose

To display information on an Ethernet RX PDU.

Description

You can create one or more Bus Instruments (RX type for Ethernet) for each Ethernet RX PDU:

- If the Show Socket Hierarchy option for the Bus Navigator tree is disabled, ControlDesk creates the bus instrument with one tab for each instance of the selected PDU.
- If the Show Socket Hierarchy option is enabled, ControlDesk creates the bus instrument only for the selected instance of the PDU.

The bus instrument opens in a separate layout when you create it via the Generate RX Layout command.

Note

You cannot manipulate Ethernet PDUs via bus instruments. For a TX PDU, ControlDesk generates a Bus Instrument (RX type for Ethernet) that is identical to the Bus Instrument (RX type for Ethernet) for the corresponding RX PDU.

Instrument elements

The Bus Instrument (RX type for Ethernet) lets you display information on the signals that are received via RX PDUs.

The information is segmented into the following regions:

Region	Description
Connection Details	**Connection Details** VLAN ID: 100 Source Address: 1080:0:0:0:41:800:200C:64 Destination Address: ff0e:0:0:0:0:0:0:64 Source Port: 30000 Destination Port: 30000 Protocol: UDP Header ID: 1239 Displays details on the connection over which the Ethernet PDU is sent.
Signals	**Signals** Signal Name / Signal Value (Source) / Signal Value (Converted) bool_Signal: --- / --- PDU_65_63_Signal: --- / --- Displays information on the signals received via the PDU.

The instrument regions can contain the following elements:

Signal Name Displays the name of the signal.

Signal Value (Source) Displays the source, i.e., the unconverted signal value.

Signal Value (Converted) Displays the signal value after it has been converted, for example, into a physical quantity according to a computation method.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

Basics

Basics on Ethernet PDUs in the Bus Navigator.....41

References

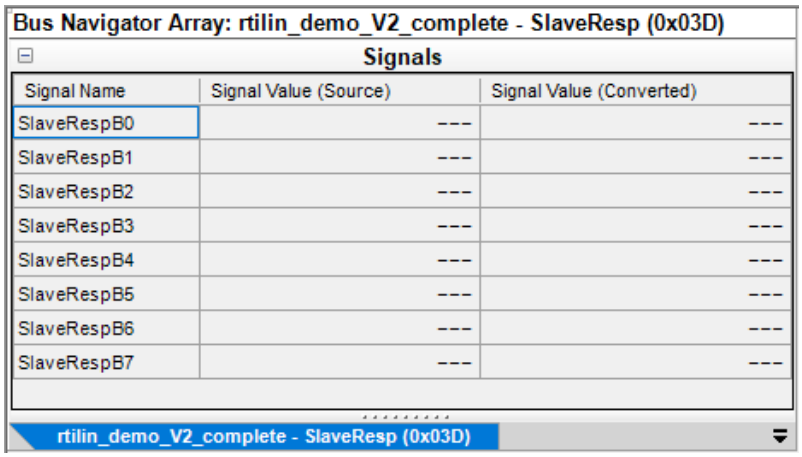
Bus Navigator.....204
Generate RX Layout.....232

LIN Instruments

Bus Instrument (RX Type for LIN)

Illustration

The following illustration shows an example of the instrument:



Purpose

To display information on an RX PDU.

Description

You can create one or more Bus Instruments (RX type for LIN) for each RX PDU in the Bus Navigator tree. The instrument opens in a separate layout when you create it via the Generate RX Layout command.

Instrument elements

The Bus Instrument (RX type for LIN) lets you display information on the signals that are received via RX PDUs.

The instruments contains the following elements:

Signal Name Displays the name of the signal.

Signal Value (Source) Displays the source, i.e., the unconverted signal value.

Signal Value (Converted) Displays the signal value after it has been converted, for example, into a physical quantity according to a computation method.

Instrument properties and commands

For information on the properties and commands of the Bus Instrument, refer to Bus Instrument (ControlDesk Instrument Handling 🕮).

Related topics

References

Generate RX Layout..... 232

Bus Navigator Commands and Dialogs

Where to go from here

Information in this section

Activate (Database) 185
To activate a real-time database, i.e., communication matrix, of a controller.

Activate (Logger) 186
To activate or deactivate the selected logger node.

Activate (Monitor) 187
To enable or disable automatic start of monitoring for the selected monitor node.

Activate (Replay) 188
To activate or deactivate the selected replay node.

Add (CAN/Ethernet) Replay 189
To replay the raw data of logged bus communication.
Not available in operator mode.

Add (CAN/LIN/Ethernet) Filter 190
To add a filter for monitoring bus communication, logging data, or replaying logged data to the selected node.
Not available in operator mode.

Add (CAN/LIN/Ethernet) Logger 191
To log communication data of a bus configuration.
Not available in operator mode.

Add (CAN/LIN/Ethernet) Monitor 193
To monitor messages/frames/PDUs.
Not available in operator mode.

Add/Edit Filter Dialog (CAN/LIN) 195
To configure settings of a filter node.

Add/Edit Filter Dialog (Ethernet) 197
To configure settings of a filter node for Ethernet communication. The filter reduces the amount of data captured by the Ethernet Bus Monitoring device.

Add/Edit Logger Dialog 199
To configure settings of a logger node.

Add/Edit Monitor Dialog 201
To configure settings of a monitor node.

Add/Edit Replay Dialog 203
To configure settings of a replay node.

Bus Navigator 204
To show the **Bus Navigator** controlbar.

Bus Navigator Page....207
To specify general Bus Navigator settings.

Case-Sensitive Search (Bus Navigator)....213
To enable differentiation between uppercase and lowercase letters for search operations.

Collapse All....213
To collapse all the subnodes of a node in the tree.

Configure Column Sets Dialog (Monitoring List)....214
To configure the column sets that are available for a bus type.

Customize Columns (Bus Navigator)....215
To add columns to or remove them from the **Bus Navigator** controlbar.

Decoding View....217
To view detailed information on an Ethernet frame selected in the monitoring list.

Delete (Filter)....219
To remove the selected filter from the Bus Navigator tree.
Not available in operator mode.

Delete (Logger)....220
To remove the selected logger node from the Bus Navigator tree.
Not available in operator mode.

Delete (Monitor)....221
To remove the selected monitor node from the Bus Navigator tree.
Not available in operator mode.

Delete (Replay)....222
To remove the selected replay node from the Bus Navigator tree.
Not available in operator mode.

Delete Column (Monitoring List)....223
To delete a column from the monitoring list.

Edit (Filter)....223
To configure the selected filter.

Edit (Logger)....224
To edit the settings of the selected logger.

Edit (Monitor)....225
To configure settings of the selected monitor node.

Edit (Replay)....226
To edit the settings of the selected replay.

Expand All....227
To expand all the subnodes of a node in the tree.

Generate Error Status Layout....228
To generate a new layout with a Bus Instrument (Error status type) for displaying the error status of all the RX and TX/RX PDUs of a communication matrix.

Not available in operator mode.

Generate Gateway Layout....229
To generate a new layout with a Bus Instrument (Gateway type).
Not available in operator mode.

Generate Inspection Layout....230
To generate a new layout with a Bus Instrument (Inspection type).
Not available in operator mode.

Generate Layout....231
To generate a new layout with a Bus Instrument (Global Layout type).
Not available in operator mode.

Generate Manipulation Layout....231
To generate a new layout with a Bus Instrument (Manipulation type).
Not available in operator mode.

Generate RX Layout....232
To generate a new layout with a Bus Instrument (RX type).
Not available in operator mode.

Generate TX Layout....234
To generate a new layout with a Bus Instrument (TX type).
Not available in operator mode.

Generate TX Status Layout....235
To generate a new layout with a Bus Instrument (TX status type) for specifying the enable options of all TX and TX/RX messages of a communication matrix.
Not available in operator mode.

Highlight Monitor....236
To highlight the selected monitor in the working area.

Import Mappings Dialog....237
To select a mapping file that contains an IP-MAC mapping list to replace IP or MAC addresses with ECU names in a monitoring list.

Load All Bus Signals....238
To load all the bus signals of a bus system to the bus configuration.
Not available in operator mode.

Locate Generated Layout....239
To locate open layouts that contain an instrument for the selected message/frame/PDU.

Monitoring List....240
To display the raw or the physical data of the monitored messages/frames/PDUs.

Save....249
To save the current monitoring data buffer to a log file.

Set as Favorite Column Set (Monitoring List)....249
To use the column set of the selected monitoring list as the default column set.

Start (Logger)..... 250
To start bus or stop data logging of the selected logger node.

Start (Monitor)..... 251
To start or stop bus monitoring of the selected monitor node.

Start (Replay)..... 252
To start or stop replaying logged data of the selected replay node.

Unload All Bus Signals..... 253
To unload all bus signals of a bus system from the bus configuration.
Not available in operator mode.

Information in other sections

Bus Configuration Structure in the Bus Navigator Tree..... 19
The Bus Navigator tree displays all the controllers, databases, messages/frames/PDUs etc. for a bus configuration hierarchically.

Activate (Database)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - communication cluster node
Shortcut key	None
Icon	None

This command is available for:

- CAN (bus communication modeled with a blockset)
- LIN (bus communication modeled with a blockset)

Purpose

To activate a real-time database, i.e., communication matrix, of a controller.

Result

The selected communication matrix is activated. All other communication matrices of the same controller are deactivated.

Description

A database, i.e., communication matrix, contains the message configurations of a CAN/LIN controller. The communication of a controller can be configured by different real-time databases (RTDB). For more information, refer to Activating Communication Matrices for Blockset-Based Bus Communication on page 44.

Related topics

Basics

Activating Communication Matrices for Blockset-Based Bus Communication..... 44
Bus Configuration Structure in the Bus Navigator Tree..... 19

Activate (Logger)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - logger node
Shortcut key	None
Icon	None
Dialog	Add/Edit Logger

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To activate or deactivate the selected logger node.

Monitoring, logging, replaying, and filtering bus communication depends on the bus type and the interface you use. Refer to Basics on Monitoring, Logging, and Filtering Bus Communication on page 77 and Basics on Replaying CAN and Ethernet Bus Communication on page 99.

Result

When the command is selected, data logging is activated.

The logger node switches to yellow in the Bus Navigator tree:

The **Start Logger** command is enabled when the simulation application for the current configuration is loaded on the platform.

Description

To start logging via the **Start Logger** command, logging must be activated.

You can add several logger nodes to one system, hardware, controller, or filter node. However, only one logger node per system, hardware, controller, or filter node can be active at the same time. When you select **Activate** for one logger node, all the other logger nodes of the same system, hardware, controller, or filter node are deactivated automatically.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Basics on Replaying CAN and Ethernet Bus Communication.....99
Bus Configuration Structure in the Bus Navigator Tree.....19

References

Start (Logger).....250

Activate (Monitor)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - monitor node (except for monitor nodes below the system node)
Shortcut key	None
Icon	None

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To enable or disable automatic start of monitoring for the selected monitor node.

Monitoring, logging, replaying, and filtering bus communication depends on the bus type and the interface you use. Refer to Basics on Monitoring, Logging, and Filtering Bus Communication on page 77 and Basics on Replaying CAN and Ethernet Bus Communication on page 99.

Result

When Activate is selected, monitoring starts automatically for the selected monitor node when monitoring is started for a higher-level monitor node.

Description

Monitoring can be started for a monitor node via:

- Start on the monitor node's context menu.
- A higher-level monitor node in the Bus Navigator tree.

For monitoring to be started via a higher-level monitor node, **Activate** must be selected. Monitoring then starts automatically when you select **Start (Monitor)** from the context menu of the higher-level node.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Basics on Replaying CAN and Ethernet Bus Communication.....99
Bus Configuration Structure in the Bus Navigator Tree.....19

References

Start (Monitor).....251

Activate (Replay)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - replay node
Shortcut key	None
Icon	None
Dialog	Add/Edit Replay

This command is available for:

- CAN
- Ethernet

Purpose

To activate or deactivate the selected replay node.

Monitoring, logging, replaying, and filtering bus communication depends on the bus type and the interface you use. Refer to Basics on Monitoring, Logging, and Filtering Bus Communication on page 77 and Basics on Replaying CAN and Ethernet Bus Communication on page 99.

Result

When the command is selected, data replay is activated and the **Start Replay** command is available.

The replay node switches to yellow in the Bus Navigator tree:

For RTICANMM-based bus communication, the activation can take some time since a MAT file must be generated for each replay.

Description

To start replaying via the **Start Replay** command, replaying must be activated.

You can add several replay nodes to one controller node. However, only one of these replay nodes can be active at the same time. When you select **Activate Replay** for one of these replay nodes, all the other replay nodes of the controller node are deactivated automatically.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Basics on Replaying CAN and Ethernet Bus Communication.....99
Bus Configuration Structure in the Bus Navigator Tree.....19

References

Start (Replay).....252

Add (CAN/Ethernet) Replay

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - controller node
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN
- Ethernet

Purpose

To replay the raw data of logged bus communication.

Monitoring, logging, replaying, and filtering bus communication depends on the bus type and the interface you use. Refer to Basics on Monitoring, Logging, and Filtering Bus Communication on page 77 and Basics on Replaying CAN and Ethernet Bus Communication on page 99.

Result

A replay node is added below the selected controller node and the Add Replay dialog opens for you to specify settings for the replay. Refer to Add/Edit Replay Dialog on page 203.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication..... 77
Basics on Replaying CAN and Ethernet Bus Communication..... 99
Bus Configuration Structure in the Bus Navigator Tree..... 19

HowTos

How to Replay Logged CAN or Ethernet Bus Communication..... 101

References

Activate (Replay)..... 188
Add/Edit Replay Dialog..... 203

Add (CAN/LIN/Ethernet) Filter

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Bus Navigator tree - system node ▪ Bus Navigator tree - hardware node ▪ Bus Navigator tree - controller node ▪ Bus Navigator tree - replay node (CAN)
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To add a filter for monitoring bus communication, logging data, or replaying logged data to the selected node.

Monitoring, logging, replaying, and filtering bus communication depends on the bus type and the interface you use. Refer to Basics on Monitoring, Logging, and Filtering Bus Communication on page 77 and Basics on Replaying CAN and Ethernet Bus Communication on page 99.

Result

A filter dialog opens for you to define the filter settings.

The content of the filter dialog depends on the bus system.

- For CAN and LIN, refer to Add/Edit Filter Dialog (CAN/LIN) on page 195.
- For Ethernet, refer to Add/Edit Filter Dialog (Ethernet) on page 197.

Related topics

Basics

Basics on Filtering Ethernet Bus Communication.....91
Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Basics on Replaying CAN and Ethernet Bus Communication.....99
Bus Configuration Structure in the Bus Navigator Tree.....19

HowTos

How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN).....85

Examples

Example of Specifying Filters for Bus Monitoring or Logging (CAN, LIN).....88
Examples of Filtering Ethernet Bus Communication.....94

References

Add/Edit Filter Dialog (CAN/LIN).....195
Add/Edit Filter Dialog (Ethernet).....197

Add (CAN/LIN/Ethernet) Logger

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Bus Navigator tree - system node ▪ Bus Navigator tree - hardware node ▪ Bus Navigator tree - controller node ▪ Bus Navigator tree - filter node

Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To log the communication data of a bus configuration.

If you add the logger to a filter node, only the filtered communication data is logged.

Monitoring, logging, replaying, and filtering bus communication depends on the bus type and the interface you use. Refer to Basics on Monitoring, Logging, and Filtering Bus Communication on page 77 and Basics on Replaying CAN and Ethernet Bus Communication on page 99.

Result

A logger node is added below the selected node and the **Add Logger** dialog opens. The dialog lets you specify settings for logging data. Refer to Add/Edit Logger Dialog on page 199.

Log file You can log the data of messages, frames, and PDUs that are active on the bus and save it to a log file. The generated log file provides information on the logged messages, frames, and PDUs in raw data format, such as the controller ID, message ID, delta time, message direction, etc.

The generated log file can be reloaded to replay the logged data.

ControlDesk uses the following log file formats:

Bus System	Log File Format	Description
CAN LIN	CSV	In a CSV file, commas are used to separate the items in each message. With an English version of Microsoft Excel, each message or frame is written to a new row and each item in a message or frame is written to a new cell. Dots are used as decimal separators.
	ASC	You can also log the data to an ASC file. The ASC format is used for log files created with CANalyzer (Vector Informatik GmbH).
Ethernet	PCAPNG	PCAPNG is a widely used file format for storing captured packet traces. A PCAPNG file contains several blocks that contain different types of

Bus System	Log File Format	Description
		information. The blocks can be used to rebuild captured packets.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Basics on Replaying CAN and Ethernet Bus Communication.....99
Bus Configuration Structure in the Bus Navigator Tree.....19

HowTos

How to Log Raw Bus Communication Data.....95

References

Add/Edit Logger Dialog.....199

Add (CAN/LIN/Ethernet) Monitor

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Bus Navigator tree - system node ▪ Bus Navigator tree - hardware node ▪ Bus Navigator tree - controller node ▪ Bus Navigator tree - filter node
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To monitor messages/frames/PDUs.

Monitoring, logging, replaying, and filtering bus communication depends on the bus type and the interface you use. Refer to Basics on Monitoring, Logging, and Filtering Bus Communication on page 77 and Basics on Replaying CAN and Ethernet Bus Communication on page 99.

Result

A monitor node is added below the selected node and a monitoring list opens. The **Add Monitor** dialog opens for you to specify the settings. Refer to Add/Edit Monitor Dialog on page 201.

When monitoring starts, the monitoring list displays all the message/frame/PDU instances. If you added the monitor to a filter node, the monitoring list displays all message/frame instances that are filtered according to the filter settings.

The monitor node is removed from the Bus Navigator tree if the monitoring list is closed.

Note

When you add a monitor node to an experiment, save the experiment to ensure that the node is saved even if you close the experiment without other changes.

LIN monitoring: automatic adaptation of frame length and checksum algorithm values To precisely check the validity of a LIN frame received during LIN monitoring, information on the frame's length and checksum algorithm specified in the LIN database (LDF file) is required. If a LIN frame was not configured in the RTI LIN MultiMessage Blockset, the frame's length and checksum algorithm values are not known. Instead, they are estimated. If a valid combination can be found using the estimated values, these values are automatically assigned to the frame. They will be used for subsequent validity checks of the frame, i.e., checksum errors etc. on subsequent frame receptions are identified according to these values.

Note

To reset the automatically assigned frame length and checksum algorithm values, you must reload the application.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77

Basics on Replaying CAN and Ethernet Bus Communication.....99

HowTos

How to Monitor Bus Communication.....82

References

Add/Edit Monitor Dialog.....201
Monitoring List.....240

Add/Edit Filter Dialog (CAN/LIN)

Access

This dialog opens when you execute one of the following commands:

- Add (CAN/LIN/Ethernet) Filter on page 190
- Edit (Filter) on page 223

Purpose

To configure settings of a filter node.

Filter tree

The filter tree on the left side of the dialog displays nodes of the Bus Navigator tree and a Range node to define an ID range for the filter. You can drag some nodes from the filter tree to the filter list of the Filter Settings page to use them as a filter.

Nodes of the Bus Navigator tree displayed in the filter tree Which nodes of the Bus Navigator tree are displayed in the filter tree depends on the bus type and the node you add the filter to.

Example: You select Add Filter from the hardware node context menu of a CAN system. In this case, the filter tree displays the following nodes: hardware node, controller node(s), ECU node(s), and message/frame node(s).

Nodes of the filter tree The filter tree consists of the following nodes:

- Events: Contains the Range node.
- Events - Range: You can drag this node to the filter list and specify an ID range to pass or block messages/frames with an ID of the specified range.
- Nodes: Contains the relevant Bus Navigator tree nodes.
- Nodes - <node_name>: Contains the name of the Bus Navigator tree node you are defining the filter for: the system node, hardware node, and/or controller node(s).

- Nodes - <ECU_name>/<Network_name>: You can drag ECU/Network nodes to the filter list and specify the direction of all the messages/frames of an ECU/Network node. The messages/frames pass the filter or are blocked according to the specified direction.
- Nodes - <message_name>/<frame name>: You can drag message/frame nodes to the filter list and specify the direction of the message/frame. The message/frame passes the filter or is blocked according to the specified direction.

Filter Settings page

Lets you specify the settings for a filter.

A filter consists of one or more filter items. Filter items with the same type (pass/pass or block/block) are combined using an implicit OR operator, filter items with different types (pass/block) are combined using an implicit AND operator.

Filter Settings for CAN/LIN

Filter name Lets you specify a name for the filter.

Note

- The filter name has to be unique on the current hierarchy level of the Bus Navigator tree.
- The permitted characters are 'A - Z', 'a - z', '0 - 9', '-' and '_'.
- Blanks are not allowed.

Name Displays the name of the range, message, or ECU node the filter item is based on.

Identifier The contents depend on the node used for the filter item:

Type	Description
Message	Displays the message ID.
ECU	Blank
Range	Lets you specify the range of message IDs in the format "<lower_limit> - <upper_limit>". The values of <lower_limit> and <upper_limit> can be specified in decimal or hexadecimal format. They must be separated by a hyphen, for example, "100-150" or "0x64-0x96". If you specify the values in decimal format they are automatically converted and displayed in hexadecimal format.

Type Displays the filter item type as Message, ECU, or Range.

Direction Lets you specify the direction of the filter item. The possible directions are RX, TX, or RX/TX.

Filter type Lets you specify the filter type. The possible types are Pass and Block.

Controller (Available only for filters of a system or hardware node) Displays the controller name or ECU node the filter item is based on.

Delete Lets you delete the selected filter item from the list.

Up Lets you move the selected filter item up the list.

Down Lets you move the selected filter item down the list.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Basics on Replaying CAN and Ethernet Bus Communication.....99
Dependencies Regarding Monitoring, Logging, and Filtering Bus Communication.....80

HowTos

How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN).....85

Add/Edit Filter Dialog (Ethernet)

Access

This dialog opens when you execute one of the following commands:

- Add (CAN/LIN/Ethernet) Filter on page 190
- Edit (Filter) on page 223

Purpose

To configure settings of a filter node for Ethernet communication. The filter reduces the amount of data captured by the Ethernet Bus Monitoring device.

Dialog settings

Filter name Lets you specify a name for the filter. A red frame is displayed if you enter an unsupported special character, such as ".

Expression Lets you enter a filter expression to specify the Ethernet packets to be captured. Filter expressions have to be defined according to the Berkeley Packet Filter (BPF) syntax. For filter examples, refer to Examples of Filtering Ethernet Bus Communication on page 94.

The **Expression** field supports *Intellisense*: Depending on characters you enter, you can select an appropriate keyword from a list.

The following buttons are available:

Button	Description
+	Opens a list with user-defined filters (if available), predefined filters, and keywords. Double-click an item to add it to the **Expression** field.

Button	Description
	Opens the **Manage Capture Filter Expressions** (refer to Manage Capture Filter Expressions dialog on page 198) dialog for you to specify, import or export user-defined Ethernet capture filters.

OK Lets you save the changes and close the dialog. If a running monitoring process uses this filter, the specified changes are applied to its subordinate monitors and loggers the next time monitoring is restarted on them.

Apply Lets you save the changes without closing the dialog. If a running monitoring process uses this filter this filter, the specified changes are applied to its subordinate monitors and loggers the next time monitoring is restarted on them.

Manage Capture Filter Expressions dialog

Lets you specify, export, and import user-defined Ethernet filters.

New Lets you add a new entry to the **Filters** list. Select **Properties – Name** and **Properties – Expression** to define the filter.

Copy Lets you copy the selected filter and add it to the **Filters** list. Select **Properties – Name** and **Properties – Expression** to define the filter.

Delete Lets you delete the selected filter.

Filters Displays the name and expressions of all the user-defined filters and lets you select one to copy or delete it.

The context menu of the **Filters** list contains all the commands of the dialog.

Properties – Name Lets you specify a name for the user-defined filter. A red frame is displayed if the filter name is not unique or if you enter an unsupported special character, such as **"**.

Properties – Expression Lets you enter a filter expression to specify the Ethernet packets to be captured. Filter expressions have to be defined according to the Berkeley Packet Filter (BPF) syntax. For filter examples, refer to Examples of Filtering Ethernet Bus Communication on page 94.

The expression field supports *Intellisense*: Depending on the characters you enter, you can select an appropriate keyword from a list.

The following buttons are available:

Button	Description
	Opens a list with user-defined filters (if available), predefined filters, and keywords. Double-click an item to add it to the **Expression** field.

Import/Export

- Import

  Opens a standard file selection dialog for you to select an Ethernet filter set (EFS) file. An EFS file contains one or more Ethernet filters.

  When you open an EFS file, the **Import Ethernet Filters** dialog opens. It displays all the filters that are contained in the EFS file and lets you select the filters to be imported. Invalid filters are ignored.

You can specify how to handle filters that have the same name as existing ones.

- **Rename** adds an underscore and a consecutive number to the name of the imported filter.
- **Overwrite** replaces the existing filters.

- **Export**

  Opens the **Export Ethernet Filters** dialog for you to select the filters to be exported to an Ethernet filter set (EFS) file.

OK Lets you save the changes and close the dialog.

Apply Lets you save the changes without closing the dialog.

Related topics

Basics

Basics on Filtering Ethernet Bus Communication.....91

Examples

Examples of Filtering Ethernet Bus Communication.....94

Add/Edit Logger Dialog

Access

The dialog opens when you execute one of the following commands:

- Add (CAN/LIN/Ethernet) Logger on page 191
- Edit (Logger) on page 224

Purpose

To configure settings of a logger node.

Dialog settings

Lets you specify logger-specific settings.

Name Lets you specify the name of the logger.

Note

- The logger name has to be unique on the current hierarchy level of the Bus Navigator tree.
- The permitted characters are 'A - Z', 'a - z', '0 - 9', '-' and '_'.
- Blanks are not allowed.

Path Lets you generate or open a log file to save incoming messages to. You can click the Browse button to open a **Specify Logfile** dialog. This dialog lets you specify a new log file or select an existing log file.

Note

Conditions for the log file are:

- The directory must exist.
- The log file name extension must be CSV, ASC or PCAPNG.
- The path must not contain any of the following characters: 'ASCII 0-31' and ", <, >, |.
- The file name must not contain any of the following characters: 'ASCII 0-31' and ", <, >, |, :, *, ?, /, \.

Format Lets you specify the format of the log file.

Overwrite file Lets you overwrite the data of an existing log file with the data of this logger.

Append data Lets you append the data of this logger to the end of an existing log file. The beginning of the appended data section is marked by an extra section header.

This option is available only for CSV log files.

Time axis (Available only for ASC files) Lets you select the time stamp relation of the time axis.

Time Axis	Time Stamp Relation
Application Time	The time stamps relate to the start of the application.
Synchronization Time	The time stamps relate to the start of the PC.

Enable Bus Statistics (Available only for controller nodes of CAN or LIN Bus Monitoring devices in connection with CAN or LIN interfaces of a SCALEXIO system, MicroAutoBox III, or VEOS) Lets you save the bus statistics together with the logged data. Both the CSV and ASC log file formats are supported.

For ASC files, the export of bus statistics is possible only if you choose Application Time as the time stamp relation of the time axis.

For information on bus statistics, refer to How to Display Bus Statistics on page 97.

Duration Lets you specify for how many seconds messages are logged.

Duration	Seconds
Minimum	1
Default	60
Maximum	3600000 (1000 hours)

Continuous logging Lets you log messages continuously. If **Continuous logging** is selected, **Duration** is disabled.

Activate Lets you activate logging.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Dependencies Regarding Monitoring, Logging, and Filtering Bus Communication.....80

HowTos

How to Log Raw Bus Communication Data.....95

Add/Edit Monitor Dialog

Access

This dialog opens when you execute one of the following commands:

- Add (CAN/LIN/Ethernet) Monitor on page 193
- Edit (Monitor) on page 225

Purpose

To configure settings of a monitor node.

Dialog settings

Monitor Name Lets you specify the name of the monitor.

Note

- The permitted characters are 'A - Z', 'a - z', '0 - 9', '-' and '_'.
- Blanks are not allowed.

Ring buffer settings Lets you specify the size and the behavior of the ring buffer.

- Fixed: If the ring buffer size is exceeded, the monitoring of the selected monitor node stops and the monitored data is not overwritten.
- Continuous: If the ring buffer size is exceeded, the monitoring of the selected monitor node continues and the monitored data is overwritten.
- Records: Lets you specify the size of the ring buffer. The specified value is the number of messages/frames/PDUs.

Number of Records	Value
Minimum	10
Default	10 000
Maximum	1 000 000 Memory requirements: One message/frame/PDU requires approx. 200 bytes. This results in a maximum of approx. 190 MB (1 000 000 * 200 bytes).

Capture settings Lets you specify the update rate in the range from 100 to 1000 ms.

J1939 settings – Use PGN to resolve messages (Available only for CAN monitoring) Use the parameter group number to resolve J1939 messages.

Name resolution settings – Import (Available only for Ethernet monitoring) Opens the Import Mappings dialog for you to select a file that contains mapping information for replacing IP or MAC addresses with ECU names in the monitoring list.

The number of imported mappings is displayed next to the Import button. If at least one valid mapping is available, the Name Resolution button in the monitoring list is activated and you can toggle between addresses and ECU names in the monitoring list.

Each imported mapping list overwrites the previous one. For more information on the dialog, refer to Import Mappings Dialog on page 237.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Dependencies Regarding Monitoring, Logging, and Filtering Bus Communication.....80

References

Import Mappings Dialog.....237
Monitoring List.....240

Add/Edit Replay Dialog

Access

This dialog opens when you execute one of the following commands:

- Add (CAN/Ethernet) Replay on page 189
- Edit (Replay) on page 226

Purpose

To configure settings of a replay node.

Dialog settings

Replay name Lets you specify the name of the replay.

Log file Lets you specify the log file you want to replay.

The following file formats are supported:

Bus Type	Supported Log File Formats
CAN	▪ ASC ▪ CSV
Ethernet	PCAPNG

Note

If more than one message per sampling step is to be transmitted on the bus during replay, the messages might be transmitted later than indicated in the log file. How many messages per sampling step can be transmitted without delay depends on a number of things, such as the bandwidth of the bus and the messages sent by the loaded application.

Section Lets you select the log file section to be replayed. A log file has at least one section. If a log file consists of different data blocks, the section headers of the data blocks are displayed to select one. In ControlDesk, different data blocks are created in a log file if you select **Append** when you specify the logger settings.

For ASC and PCAPNG files, the field is inactive, because these file formats do not support sections.

Auto repeat Lets you select whether replay is repeated continuously.

Passes Lets you select whether replay is repeated in passes and specify how often the replay is passed.

Duration Lets you specify for how many seconds the log file is replayed.

Activate Lets you activate the replay. You can also activate the replay later via its context menu in the Bus Navigator tree.

You can add several replay nodes to one controller node. However, only one replay node can be active at a time. When you select **Activate** for a replay node, all the other replay nodes of the controller node are deactivated automatically.

Start monitoring when replay starts Lets you specify to start monitoring when replay starts so that you can visualize the replayed communication data in ControlDesk.

Note

You have to stop the monitoring manually.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication..... 77
Basics on Replaying CAN and Ethernet Bus Communication..... 99

HowTos

How to Replay Logged CAN or Ethernet Bus Communication..... 101

Bus Navigator

Access

You can access this command via:

Ribbon	View - Controlbar
Context menu of	None
Shortcut key	None
Icon	None

Purpose

To show the **Bus Navigator** controlbar.

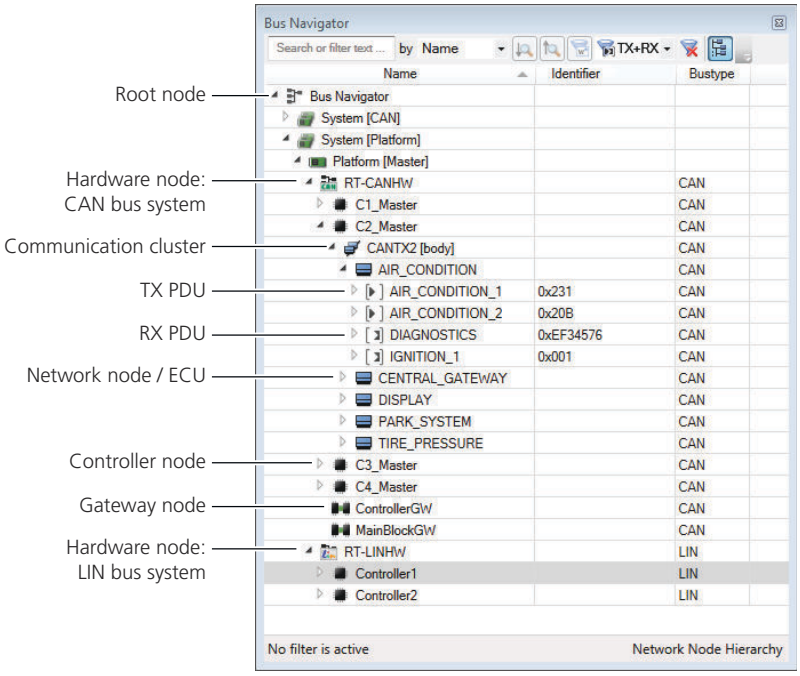
Result

The **Bus Navigator** controlbar opens.

Bus Navigator controlbar

The **Bus Navigator** controlbar displays a tree that lets you handle the bus communication in the following cases:

- The experiment contains a platform with an SDF file generated for a simulation application that contains one or more bus configurations.
- The experiment contains a bus monitoring device.



The Bus Navigator tree displays the bus configuration hierarchically.

- You can open different instruments to observe the transmission and reception of individual messages or manipulate their transmission options. Refer to Working with Bus Instruments on page 48.
- You can monitor the entire bus communication filtered or unfiltered, log the monitored data, and replay logged data. Refer to Monitoring and Logging Bus Communication on page 76.

Function buttons The following function buttons are available below the Bus Navigator header.

Function Button	Description
Search or filter text ...	Lets you enter a text string for searching and filtering the tree. You can select the column that is used for searching in the column list to the right.
by Name	Lets you select the column that is used for searching or filtering.
	Selects the next occurrence of the search string.
	Selects the previous occurrence of the search string.
	Enables a wildcard filter that filters the tree by the string that is currently entered in the search text field. The wildcard filter is active until the button is pressed again.
TX+RX	Displays all the TX and RX messages/frames/PDUs in the tree. Click the down arrow to filter the tree.

Function Button	Description
TX	Displays only TX messages/frames/PDUs. Click the down arrow to change the filter setting.
RX	Displays only RX messages/frames/PDUs. Click the down arrow to change the filter setting.
	Clears all the filter settings.
	Shows/hides the network nodes in the tree.
	Shows/hides Ethernet sockets in the tree.

Incremental search

You can search for an item in the tree via incremental search by selecting an item of a column and then typing the first characters of the entry you want to find. The current keystroke sequence is shown in the column header.

- You can use the wildcards ***** and **?** in the keystroke sequence.
- To select the next item that matches the keystroke sequence, press **F3**.

  To select the previous one, press **Shift+F3**.
- To delete the last keystroke from the keystroke sequence, press **Backspace**. The selection is modified.

  To delete the whole keystroke sequence, press **Ctrl+Backspace** or **Delete**. The selection remains.

Case-sensitive search

You can enable differentiation between uppercase and lowercase letters for search operations. Refer to Case-Sensitive Search (Bus Navigator) on page 213.

Related commands

Refer to Bus Navigator Commands and Dialogs on page 182.

Related topics

Basics

Basics on Working with the Bus Navigator 34
Bus Configuration Structure in the Bus Navigator Tree..... 19

Bus Navigator Page

Access

This page is part of the ControlDesk Options dialog.

The dialog can be opened via the **Options Command** (ControlDesk User Interface Handling 🕮).

Purpose

To specify general Bus Navigator settings.

Description

You can configure the names of the layouts that are generated automatically with a Bus Instrument. You can use predefined macros or typed text to define the layout names.

You can define the layout name of each layout type in a separate edit field. A position mark indicates where the cursor is placed in the edit field to configure the layout name. The typed text or macros are inserted at this position.

Note

Make sure that the cursor is not placed within the braces of a macro when configuring the layout name.

Startup configuration

Load Bus Navigator configurations automatically Lets you specify whether to load the bus configuration into the Bus Navigator automatically when the variable description references bus configuration files. You must restart ControlDesk to let the changes take effect.

Load all bus signals automatically (only RTIxxMM-based applications) Lets you specify whether to load all bus signals to a bus configuration when it is initially created, i.e. when the related variable description is initially added to an experiment.

Preventing the bus signals from loading is useful for bus configurations containing many bus signals which may cause memory exceptions.

This option affects only bus configurations modeled with:

- RTI CAN MultiMessage Blockset
- RTI LIN MultiMessage Blockset

After creating the bus configuration, you can manually load or unload bus signals. Refer to the following commands:

- Load All Bus Signals on page 238
- Unload All Bus Signals on page 253

Monitoring settings

Disable time synchronization for DS1006 and MicroAutoBox Lets you enable/disable the time synchronization of bus data displayed in a

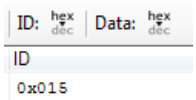
monitor/logger, regardless of the platform/device it is read from. By default, time synchronization is disabled. Keep in mind that enabling time synchronization requires bandwidth of the host interface connection.

Configure View Settings Opens the Monitoring View Settings dialog for you to specify the default settings of a monitoring list.

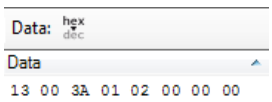
Manage Ethernet Capture Filters Opens the Manage Capture Filter Expressions (refer to Manage Capture Filter Expressions dialog on page 198) dialog for you to specify, import or export user-defined Ethernet capture filters.

Display format

ID Format Lets you specify whether to display IDs in hexadecimal or decimal format. This default setting applies to various elements related to the Bus Navigator, such as monitoring lists, variable lists, Bus Navigator tree, or bus instruments. Some of these elements let you change the setting locally. A monitoring list, for example, offers a button in its toolbar:



Data Byte Format Lets you specify whether to display data bytes in hexadecimal or decimal format. This default setting applies to various elements related to the Bus Navigator, such as monitoring lists, variable lists, Bus Navigator tree, or bus instruments. Some of these elements allow you to change the settings locally. A monitoring list, for example, offers a button in its toolbar:



Layout naming macros

RX layout Lets you specify the default names of layouts with an Bus Instrument (RX type). You can type text in the edit field to specify a name or select predefined macros from the list.

List Entry	Description
Platform Name	Adds a %PlatformName% macro to the list, and the name of the platform is added to the RX layout names.
Controller Name	Adds a %ControllerName% macro to the list, and the name of the relevant controller is added to the RX layout names.
Bus Type	Adds a %BusType% macro to the list, and the name of the relevant bus type is added to the RX layout names.
Database Name	Adds a %DatabaseName% macro to the list, and the name of the relevant database is added to the RX layout names.

List Entry	Description
Message - Name	Adds a %MessageName% macro to the list, and the name of the relevant message is added to the RX layout names.
Message - ID	Adds a %MessageID% macro to the list, and the relevant message ID is added to the RX layout names.
Reset to Default	Resets the RX layout names to the default, RX{%MessageName%}{%MessageID%}{%BusType%}.

TX layout Lets you specify the default names of layouts with a Bus Instrument (TX type). You can type text in the edit field to specify a name or select predefined macros from the list.

List Entry	Description
Platform Name	Adds a %PlatformName% macro to the list, and the name of the platform is added to the TX layout names.
Controller Name	Adds a %ControllerName% macro to the list, and the name of the relevant controller is added to the TX layout names.
Bus Type	Adds a %BusType% macro to the list, and the name of the relevant bus type is added to the TX layout names.
Database Name	Adds a %DatabaseName% macro to the list, and the name of the relevant database is added to the TX layout names.
Message - Name	Adds a %MessageName% macro to the list, and the name of the relevant message is added to the TX layout names.
Message - ID	Adds a %MessageID% macro to the list, and the relevant message ID is added to the TX layout names.
Reset to Default	Resets the TX layout names to the default, TX{%MessageName%}{%MessageID%}{%BusType%}.

TX status layout Lets you specify the default names of layouts with a Bus Instrument (TX status type). You can type text in the edit field to specify a name or select predefined macros from the list.

List Entry	Description
Bus Type	Adds a %BusType% macro to the list, and the name of the relevant bus type is added to the TX status layout names.
Database Name	Adds a %DatabaseName% macro to the list, and the name of the relevant database is added to the TX status layout names.

List Entry	Description
Reset to Default	Resets the TX layout names to the default, ST{%MatrixName%}{%BusType%}.

Gateway layout Lets you specify the default names of layouts with a Bus Instrument (Gateway type). You can type text in the edit field to specify a name or select predefined macros from the list.

List Entry	Description
Bus Type	Adds a %BusType% macro to the list, and the name of the relevant bus type is added to the gateway layout names.
Gateway Name	Adds a %GatewayName% macro to the list, and the name of the relevant gateway is added to the gateway layout names.
Reset to Default	Resets the TX layout names to the default, GW{%GatewayName%}{%BusType%}.

Inspection layout Lets you specify the default names of layouts with a Bus Instrument (Inspection type). You can type text in the edit field to specify a name or select predefined macros from the list.

List Entry	Description
Platform Name	Adds a %PlatformName% macro to the list, and the name of the platform is added to the Inspection layout names.
Controller Name	Adds a %ControllerName% macro to the list, and the name of the relevant controller is added to the Inspection layout names.
Bus Type	Adds a %BusType% macro to the list, and the name of the relevant bus type is added to the Inspection layout names.
Database Name	Adds a %DatabaseName% macro to the list, and the name of the relevant database is added to the Inspection layout names.
Message - Name	Adds a %MessageName% macro to the list, and the name of the relevant message is added to the Inspection layout names.
Message - ID	Adds a %MessageID% macro to the list, and the relevant message ID is added to the Inspection layout names.
Reset to Default	Resets the Inspection layout names to the default, TX{%MessageName%}{%MessageID%}{%BusType%}.

Manipulation layout Lets you specify the default names of layouts with a Bus Instrument (Manipulation type). You can type text in the edit field to specify a name or select predefined macros from the list.

List Entry	Description
Platform Name	Adds a %PlatformName% macro to the list, and the name of the platform is added to the Manipulation layout names.
Controller Name	Adds a %ControllerName% macro to the list, and the name of the relevant controller is added to the Manipulation layout names.
Bus Type	Adds a %BusType% macro to the list, and the name of the relevant bus type is added to the Manipulation layout names.
Database Name	Adds a %DatabaseName% macro to the list, and the name of the relevant database is added to the Manipulation layout names.
Message - Name	Adds a %MessageName% macro to the list, and the name of the relevant message is added to the Manipulation layout names.
Message - ID	Adds a %MessageID% macro to the list, and the relevant message ID is added to the Manipulation layout names.
Reset to Default	Resets the Manipulation layout names to the default, TX{%MessageName%}{%MessageID%}{%BusType%}.

Layout generation settings

Min. instrument width Lets you specify a minimum width for newly added bus instruments (in pixels).

Automatically extend the width of the instrument to fit its content Lets you specify to extend the width of newly added bus instruments to avoid scrollbars if the content is larger than the instrument's minimum width.

Min. instrument height Lets you specify a minimum height for newly added bus instruments (in pixels).

Automatically extend the width of the instrument to fit its content Lets you specify to extend the height of newly added bus instruments to avoid scrollbars if the content is larger than the instrument's minimum height.

Include all signals in RX layouts (only for CAN Bus Monitoring devices) Lets you display each signal of an RX PDU or RX multiplexed PDU in a bus instrument (RX type), regardless of whether the signal is actually sent or received by an ECU.

Monitoring View Settings dialog

To specify the default settings of the monitoring view and the column selection of monitoring lists. The default settings are used when you add a new monitoring list.

View Mode (Available only for CAN and LIN) Lets you select a view for the monitoring list.

- Static View: In static view mode, the monitoring list contains one row for each message/frame, and the row is updated with the latest received data.
- Continuous View: In continuous view mode, every new instance of a message/frame is added to the monitoring list in a new row.
- Hierarchical View: With hierarchical sorting by message/frame ID, the monitoring list is displayed in a tree view with the ID entries at the top level. Every new instance of a message/frame is added to the list in a new row. Hierarchical sorting by message/frame ID sorts message/frame instances according to their IDs.

Scroll Mode Lets you specify whether to scroll the lines of the monitoring list up or down when incoming lines are added to the visible part of the list in the continuous view.

- Tail scrolling: The lines are scrolled up. If you have sorted the list by time stamps, for example, the latest incoming messages are placed in the bottom lines and the older ones are scrolled up.
- Head scrolling: The lines are scrolled down. If you have sorted the list by time stamps, for example, the latest incoming messages are placed in the top lines and the older ones are scrolled down.

View Configuration – Configure Column Sets Opens the Configure Column Sets dialog for you to configure the column sets that are available for a bus type. Refer to Configure Column Sets Dialog (Monitoring List) on page 214.

View Configuration – Manage SOME/IP-SD Ports (Available only for Ethernet) Opens the Manage Ports dialog for you to specify the ports that are used to decode SOME/IP-SD messages. Refer to Manage Ports dialog (refer to Manage Ports dialog on page 212).

Manage Ports dialog

Lets you specify the ports that are used to decode SOME/IP-SD messages.

Decoding SOME/IP-SD messages is not possible without specified ports. The standard ports that are defined in the SOME/IP protocol are listed by default: 30490, 30491, 30501, 30502, 30503, 30504.

Decoding SOME/IP-SD messages is independent from the transport protocol (TCP or UDP).

Add Lets you add a new port to the list of ports. In the edit field, you can enter the port value in the range of 0 to 65535.

Ports used for SOME/IP-SD decoding Displays the specified ports.

Remove Lets you remove the selected port from the list of specified ports.

Remove All Lets you clear the list of ports.

Reset to Default Lets you reset the list to the default ports.

Related topics

References

Options Command (ControlDesk User Interface Handling 🕮)

Case-Sensitive Search (Bus Navigator)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator
Shortcut key	None
Icon	None

Purpose

To enable differentiation between uppercase and lowercase letters for search operations.

Description

If **Case-Sensitive Search** is disabled, you can find the "XYZ" string with the "xyz" search string. If it is enabled, you must enter "XYZ" to find it.

Related topics

References

Bus Navigator 204

Collapse All

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator
Shortcut key	None
Icon	None

Purpose

To collapse all the subnodes of a node in the tree.

Related topics

References

Expand All..... 227

Configure Column Sets Dialog (Monitoring List)

Access

- This dialog opens when you click on the toolbar of a monitoring list. Refer to Monitoring List on page 240.
- You can also open the dialog via the ControlDesk **Options** dialog: On the **Bus Navigator** page, click **Configure View Settings**. Select the bus type and click **Default Columns – Configure**. Refer to Bus Navigator Page on page 207.

Purpose

To configure the column sets that are available for a bus type.

Dialog settings

The following dialog elements are available:

Available Columns and Visible Columns lists The **Available Columns** list displays the columns that are available for the selected bus type.

The **Visible Columns** list shows which of the available columns are visible in the monitoring list.

You can move an entry between the lists as follows:

- Drag the list entry to the other list.
- Select the list entry and click ➡ or ⬅.
- Double-click the list entry.

You can sort the entries in the Visible Columns list as follows:

- Drag the list entry to the desired position.
- Select a list entry and click ⬆ or ⬇.

Column Set Lets you select a predefined or user-defined column set. The settings of the selected column set are displayed in the dialog.

Copy Lets you copy the selected column set and add it to the column set list as new user-defined column set. A dialog opens for you to change the column set name. You cannot enter a name that is already used for an existing column set.

Delete (Available only for user-defined column sets) Lets you delete the selected column set.

Favorite Lets you set the selected column set as the favorite column set. You can only specify one favorite column set per bus type.

The favorite column set is used in the Bus Navigator when you add a new monitoring list.

Import/Export

- Import

  Opens a standard file selection dialog for you to select a column set (CSET) file. A CSET file contains one or more column sets.

  When you open a CSET file, the **Import Column Sets** dialog opens. It displays all the column sets that are contained in the CSET file and lets you select the column sets to be imported.

  You can specify how to handle filters that have the same name as existing ones.

  - **Rename** adds a **#** and a consecutive number to the name of the imported column set.
  - **Overwrite** replaces the existing column set.
- Export

  Opens the **Export Column Sets** dialog for you to select the column sets to be exported to a column set (CSET) file.

Rename (Available only for user-defined column sets) Lets you change the name of the selected column set.

Reset (Available only for predefined column sets) Lets you reset the visible columns of the column set to the default state.

Related topics

References

Bus Navigator Page 207
Delete Column (Monitoring List) 223
Monitoring List 240
Set as Favorite Column Set (Monitoring List) 249

Customize Columns (Bus Navigator)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator (column header)
Shortcut key	None
Icon	None

Purpose

To add columns to or remove them from the Bus Navigator controlbar.

Result

The added columns appear in the Bus Navigator controlbar, the removed columns are hidden.

Customize Columns dialog

The columns list contains all the columns that can be displayed in the Bus Navigator controlbar. You can add a column to the Bus Navigator controlbar by selecting its checkbox. To remove a column, clear its checkbox.

The following columns are available:

Column	Description
Name	Displays the name of the bus element. The Name column is always visible and cannot be moved.
Identifier	Displays the identifier of the bus element (if available).
AST	Displays information about the absolutely scheduled timing (slot ID, base cycle, cycle repetition) (if available).
Channel	Displays the channel of the bus element (if available).
Bustype	Displays the bus type of the bus element.

You can use the following commands/settings to configure the columns.

Hide Lets you specify to hide the column in the Bus Navigator controlbar.

Move Down Moves the selected column down the list (to the right in the Bus Navigator controlbar).

Move Up Moves the selected column up the list (to the left in the Bus Navigator controlbar).

Show Lets you specify to display the column in the Bus Navigator controlbar.

Visible Lets you specify to display/hide the column in the Bus Navigator controlbar by activating a checkbox.

Width Lets you enter a width for the selected column (in pixels).

Related topics

References

Bus Navigator..... 204

Decoding View

Access

You can show and hide the decoding view via the **Decoding View** button in the toolbar of the monitoring list. Refer to Monitoring List on page 240.

The decoding view is only available for Ethernet monitoring.

Purpose

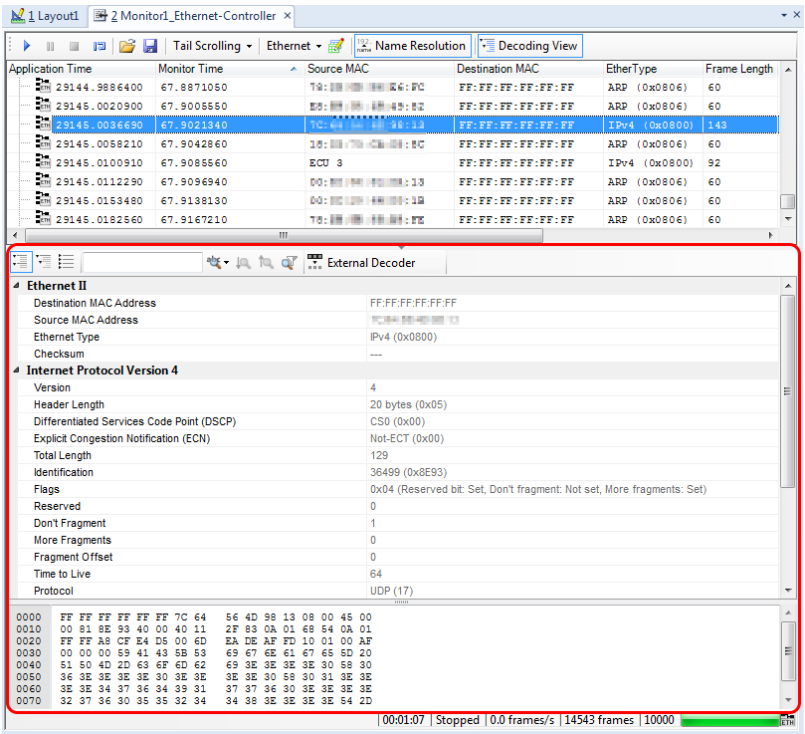
To view detailed information on the Ethernet frame that is selected in the monitoring list.

Result

Opens the decoding view to display detailed information on the selected Ethernet frame.

Description

Decoding view ControlDesk's monitoring list provides a decoding view that displays protocol data for the Ethernet frame selected in the monitoring list. In addition, Ethernet II raw data is displayed in the byte view at the bottom of the decoding view.



For information on the displayed protocol data items, refer to the description of the columns in the monitoring list. Refer to Configure Column Sets Dialog (Monitoring List) on page 214.

Tip

You can use the byte view at the bottom of the decoding view to copy raw data to the Clipboard. Select the data with the mouse pointer and open the context menu or press `Ctrl+C`.

Buttons The decoding view provides the following buttons:

Button	View
	All categories are expanded (showing the first-level items). All categories and all items are expanded. All categories are collapsed.
(Search by name)	Lets you search for item names. Enter the search string in the edit field and click or to find the previous or the next matching item. To change the search scope, click .
(Search by value)	Lets you search for item values. Enter the search string in the edit field and click or to find the previous or the next matching value. To change the search scope, click .
(Search by name and value)	Lets you search for item names and values. Enter the search string in the edit field and click or to find the previous or the next matching item. To change the search scope, click .
	Lets you search all color properties for a name that contains the search phrase. Enter the search string in the edit field and click or to find the previous or the next matching item. To change the search scope, click .
	Lets you search all font properties for a name that contains the search phrase. Enter the search string in the edit field and click or to find the previous or the next matching item. To change the search scope, click .
	Lets you jump to the next occurrence of the search string.
	Lets you jump to the previous occurrence of the search string.
	Lets you filter the list by the search results. All the matching items including their parent objects are displayed.

Button	View
External Decoder (External Decoder)	Lets you use the Wireshark plug-ins for decoding. **Note** Keep in mind the Important Information on Using an External Decoder for Interpreting Ethernet Raw Data (ControlDesk Introduction and Overview 🕮).

Related topics

References

Monitoring List..... 240

Delete (Filter)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - filter node
Shortcut key	`Delete`
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To remove the selected filter from the Bus Navigator tree.

Result

A dialog opens, asking you whether you want to delete the filter node or not. If you click Yes in the dialog, the filter node and all its monitor, logger, and replay nodes are removed from the Bus Navigator tree.

Note

You cannot delete a filter while its subordinate monitor, logger, or replay nodes are active. If you do so, an error message informs you to stop monitoring, logging, or replaying first.

Related topics

Basics

Basics on Filtering Ethernet Bus Communication.....91
Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Bus Configuration Structure in the Bus Navigator Tree.....19

HowTos

How to Replay Logged CAN or Ethernet Bus Communication.....101
How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN).....85

Delete (Logger)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - logger node
Shortcut key	**`Delete`**
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To remove the selected logger node from the Bus Navigator tree.

Result

A dialog opens, asking you whether you want to delete the logger node or not. If you click **Yes** in the dialog, the logger node is removed from the Bus Navigator tree. If data logging was activated, logging stops automatically.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Bus Configuration Structure in the Bus Navigator Tree.....19

Delete (Monitor)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - monitor node
Shortcut key	**`Delete`**
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To remove the selected monitor node from the Bus Navigator tree.

Tip

Alternatively, you can close the corresponding monitoring list to remove the monitor node from the Bus Navigator tree.

Result

The monitor node is removed from the Bus Navigator tree and the corresponding monitoring list is closed immediately. If monitoring was activated, it stops automatically.

Note

You are not asked whether or not you want to remove the monitor.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Bus Configuration Structure in the Bus Navigator Tree.....19

Delete (Replay)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - replay node
Shortcut key	**`Delete`**
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN
- Ethernet

Purpose

To remove the selected replay node from the Bus Navigator tree.

Result

A dialog opens, asking you whether you want to delete the replay node or not. If you click **Yes** in the dialog, the replay node is removed from the Bus Navigator tree. If a replay is in progress, it is stopped automatically.

Related topics

Basics

Basics on Replaying CAN and Ethernet Bus Communication.....99
Bus Configuration Structure in the Bus Navigator Tree.....19

Delete Column (Monitoring List)

Access

You can access this command via:

Ribbon	None
Context menu of	Monitoring list (column header)
Shortcut key	None
Icon	None

Purpose

To delete a column from the monitoring list.

Description

By deleting a column from the monitoring list, you modify the column set. If the column set is a predefined column set, you can reset it to the default settings in the **Configure Column Sets** dialog. Refer to Configure Column Sets Dialog (Monitoring List) on page 214.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77

Edit (Filter)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - filter node
Shortcut key	None
Icon	None

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To configure the selected filter.

Result

The **Edit Filter** dialog opens for you to configure the filter settings.

The content of the filter dialog depends on the bus system.

- For CAN and LIN refer to Add/Edit Filter Dialog (CAN/LIN) on page 195.
- For Ethernet, refer to Add/Edit Filter Dialog (Ethernet) on page 197.

Related topics

Basics

Basics on Filtering Ethernet Bus Communication.....91
Bus Configuration Structure in the Bus Navigator Tree.....19

HowTos

How to Specify a Filter for Bus Monitoring and Logging (CAN, LIN).....85

References

Add/Edit Filter Dialog (CAN/LIN).....195
Add/Edit Filter Dialog (Ethernet).....197

Edit (Logger)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - logger node
Shortcut key	None
Icon	None

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To edit the settings of the selected logger.

Description

The **Edit Logger** dialog opens for you to edit the specified logger settings. Refer to Add/Edit Logger Dialog on page 199.

Log file You can log the data of messages, frames, and PDUs that are active on the bus and save it to a log file. The generated log file provides information

on the logged messages, frames, and PDUs in raw data format, such as the controller ID, message ID, delta time, message direction, etc.

The generated log file can be reloaded to replay the logged data.

ControlDesk uses the following log file formats:

Bus System	Log File Format	Description
CAN LIN	CSV	In a CSV file, commas are used to separate the items in each message. With an English version of Microsoft Excel, each message or frame is written to a new row and each item in a message or frame is written to a new cell. Dots are used as decimal separators.
	ASC	You can also log the data to an ASC file. The ASC format is used for log files created with CANalyzer (Vector Informatik GmbH).
Ethernet	PCAPNG	PCAPNG is a widely used file format for storing captured packet traces. A PCAPNG file contains several blocks that contain different types of information. The blocks can be used to rebuild captured packets.

Related topics

Basics

Basics on Replaying CAN and Ethernet Bus Communication.....99
Bus Configuration Structure in the Bus Navigator Tree.....19

HowTos

How to Log Raw Bus Communication Data.....95

References

Activate (Logger).....186
Add/Edit Logger Dialog.....199

Edit (Monitor)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - monitor node
Shortcut key	None
Icon	None

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To configure settings of the selected monitor node.

Result

The **Edit Monitor** dialog opens for you to configure the settings of the selected monitor node. Refer to Add/Edit Monitor Dialog on page 201.

Related topics

Basics

Bus Configuration Structure in the Bus Navigator Tree..... 19

HowTos

How to Monitor Bus Communication..... 82

References

Add/Edit Monitor Dialog..... 201
Monitoring List..... 240

Edit (Replay)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - replay node
Shortcut key	None
Icon	None

This command is available for:

- CAN
- Ethernet

Purpose

To edit the settings of the selected replay.

Result

The **Edit Replay** dialog opens for you to edit the replay settings. Refer to Add/Edit Replay Dialog on page 203.

Related topics

Basics

Bus Configuration Structure in the Bus Navigator Tree..... 19

HowTos

How to Replay Logged CAN or Ethernet Bus Communication..... 101

References

Activate (Replay)..... 188
Add/Edit Replay Dialog..... 203

Expand All

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator
Shortcut key	None
Icon	None

Purpose

To expand all the subnodes of a node in the tree.

Related topics

References

Collapse All..... 213

Generate Error Status Layout

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - FlexRay communication matrix node
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- FlexRay (bus communication modeled with a blockset)

Purpose

To generate a new layout with a Bus Instrument (Error status type) for displaying the error status of all the RX and TX/RX PDUs of a communication matrix.

Result

The Bus Instrument (Error status type) opens.

Description

The instrument displays the error status of all the RX and TX/RX PDUs of a communication matrix. Refer to Bus Instrument (Error Status Type for FlexRay) on page 155.

Related topics

HowTos

How to Display the Error Status for all the RX PDUs of a FlexRay Controller..... 71

References

Bus Instrument (Error Status Type for FlexRay)..... 155

Generate Gateway Layout

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - gateway node
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN (bus communication modeled with a blockset)

Purpose

To generate a new layout with a Bus Instrument (Gateway type).

Result

The Bus Instrument (Gateway type) opens.

Description

The instrument displays information on the gateway and the gatewayed messages (refer to Bus Instrument (Gateway Type for CAN) on page 140).

Tip

You can configure the default layout name, refer to Bus Navigator Page on page 207.

Related topics

HowTos

How to Create a Bus Instrument (Gateway Type) for a CAN Controller Gateway..... 67

References

Bus Instrument (Gateway Type for CAN)..... 140

Generate Inspection Layout

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Bus Navigator tree - RX PDU node (configured for inspection purposes) ▪ Bus Navigator tree - TX/RX PDU node (configured for inspection purposes)
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN (bus communication modeled with the Bus Manager)
- LIN (bus communication modeled with the Bus Manager)

Purpose

To generate a new layout with a Bus Instrument (Inspection type).

Result

The Bus Instrument (Inspection type) opens.

Description

The instrument displays the values of the signals (refer to Bus Instruments on page 104).

You can change the properties and the content of generated layouts and instruments. For example, you can place more instruments on the layout and drag other signals from the Bus Navigator to the instrument.

Tip

You can configure the default layout name, refer to Bus Navigator Page on page 207.

Related topics

References

Bus Instrument (Inspection Type for CAN)..... 107
Bus Instrument (Inspection Type for LIN)..... 126

Generate Layout

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - Platform node
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN (bus communication modeled with the Bus Manager)
- LIN (bus communication modeled with the Bus Manager)

Purpose

To generate a new layout with a Bus Instrument (Global Layout type).

Result

The Bus Instrument (Global Layout type) opens.

Description

The instrument displays the bus configurations and lets you enable or disable the entire bus communication of a bus configuration. Refer to Bus Instrument (Global Layout Type) on page 105).

Related topics

References

Bus Instrument (Global Layout Type)..... 105

Generate Manipulation Layout

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - TX/ PDU node (configured for manipulation purposes)
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN (bus communication modeled with the Bus Manager)
- LIN (bus communication modeled with the Bus Manager)

Purpose

To generate a new layout with a Bus Instrument (Manipulation type).

Result

The Bus Instrument (Manipulation type) opens.

Description

The instrument displays the values of the signals. Refer to Bus Instruments on page 104.

You can change the properties and the content of generated layouts and instruments. For example, you can place more instruments on the layout and drag other signals from the Bus Navigator to the instrument.

Tip

You can configure the default layout name, refer to Bus Navigator Page on page 207.

Related topics

References

Bus Instrument (Manipulation Type for CAN)..... 110
Bus Instrument (Manipulation Type for LIN)..... 128

Generate RX Layout

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree: ▪ RX message/frame/PDU node ▪ TX/RX message/frame/PDU node ▪ RX PDU node (configured for simulation purposes) ▪ TX/RX PDU node (configured for simulation purposes)

	▪ TX PDU node (Ethernet)
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN
- Ethernet
- FlexRay
- LIN

Purpose

To generate a new layout with an Bus Instrument (RX type).

Result

The Bus Instrument (RX type) opens.

Description

The instrument displays information on the RX message/frame/PDU and its signals (refer to Bus Instruments on page 104).

You can change the properties and the content of generated layouts and instruments. For example, you can place more instruments on the layout and drag other signals from the Bus Navigator to the instrument.

Tip

You can configure the default layout name, refer to Bus Navigator Page on page 207.

Related topics

HowTos

How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs..... 50

References

Bus Instrument (RX Type for CAN)..... 114
Bus Instrument (RX Type for CAN)..... 142
Bus Instrument (RX Type for CAN)..... 175
Bus Instrument (RX Type for Ethernet)..... 178
Bus Instrument (RX Type for FlexRay)..... 157
Bus Instrument (RX Type for LIN)..... 131
Bus Instrument (RX Type for LIN)..... 169

Generate TX Layout

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree: ▪ TX message/frame/PDU node ▪ TX/RX message/frame/PDU node ▪ TX PDU node (configured for simulation purposes) ▪ TX/RX PDU node (configured for simulation purposes)
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN
- FlexRay
- LIN

Purpose

To generate a new layout with a Bus Instrument (TX type).

Result

The Bus Instrument (TX type) opens.

Description

The instrument displays information on the TX message/frame/PDU and its signals (refer to Bus Instruments on page 104).

The data source for the layout depends on the online calibration status:

- When online calibration is stopped, the Bus Navigator loads initial data from the SDF file.
- When online calibration is started, the Bus Navigator loads data from the application running on the platform.

  You can configure message/frame/PDU data during run time.

You can change the properties and the content of generated layouts and instruments. For example, you can place more instruments on the layout and drag other signals from the Bus Navigator to the instrument.

Tip

You can configure the default layout name, refer to Bus Navigator Page on page 207.

Related topics

Basics

Bus Instrument (TX Type for CAN)..... 176

HowTos

How to Configure a TX Message/Frame/PDU and Transmit It on the Bus..... 59
How to Configure a TX Message/Frame/PDU and Transmit It on the Bus..... 64
How to Create Bus Instruments for Handling RX and TX Messages/Frames/PDUs..... 50

References

Bus Instrument (TX Type for CAN)..... 121
Bus Instrument (TX Type for CAN)..... 147
Bus Instrument (TX Type for FlexRay)..... 160
Bus Instrument (TX Type for LIN)..... 136
Bus Instrument (TX Type for LIN)..... 171

Generate TX Status Layout

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - communication matrix node
Shortcut key	None
Icon	None

Note

This command is not available in operator mode.

This command is available for:

- CAN
- FlexRay
- LIN (bus communication modeled with the Bus Manager)

Purpose

To generate a new layout with a Bus Instrument (TX status type) for specifying the enable options of all TX and TX/RX messages of a communication cluster.

Result

The Bus Instrument (TX status type) of the database opens.

Description

The Bus Instrument (TX status type) lets you enable transmission of all the TX and TX/RX messages or just for a subset of the messages. In addition, you can specify transmission options of the messages, such as cycle time, cyclic transmission, and kickout.

Tip

You can configure the default layout name, refer to Bus Navigator Page on page 207.

Related topics

HowTos

How to Specify Global Transmission Settings for a CAN or FlexRay Controller..... 69
How to Specify Global Transmission Settings for a CAN or LIN Controller..... 61

References

Bus Instrument (TX Status Type for CAN)..... 145
Bus Instrument (TX Status Type for CAN)..... 119
Bus Instrument (TX Status Type for FlexRay)..... 166
Bus Instrument (TX Status Type for LIN)..... 134

Highlight Monitor

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - monitor node
Shortcut key	None
Icon	None

Purpose

To highlight the selected monitor in the working area.

Result

ControlDesk highlights the selected monitor in the working area.

Related topics

References

Activate (Monitor)..... 187
Add (CAN/LIN/Ethernet) Monitor..... 193

Delete (Monitor) 221
Edit (Monitor) 225
Monitoring List 240
Start (Monitor) 251

Import Mappings Dialog

Access

(Available only for Ethernet monitoring) This dialog opens when you open an **Add/Edit Monitor** dialog and then click **Import** to import a name resolution file.

Purpose

To select a mapping file that contains an IP-MAC mapping list to replace IP or MAC addresses with ECU names in a monitoring list.

Description

Lets you select a mapping file that contains an IP/MAC mapping list to replace IPv4, IPv6, or MAC addresses with ECU names in a monitoring list.

The mapping file is a text file (without a specific file name extension) and lists IP/MAC addresses and ECU names as shown in the following example. You can also enter comment lines by starting them with a `#`.

```
10.0.8.15 ECU_1
192.0.8.15 ECU_2
#This is my comment
24:6E:66:1E:8E:52 ECU_3
3C:D9:2B:5E:BD:EB ECU_4
```

The following alternative spellings are supported:

- MAC and IPv6 addresses can be written with `-`.
- IPv6 addresses can be written with a double colon (`::`) as shortcut for contiguous zeros. Only one double colon is allowed in an address.

The addresses are replaced by ECU names in following columns of the monitoring list:

- Source MAC
- Destination MAC
- Source Address (IPv4)
- Destination Address (IPv4)
- Source Address (IPv6)
- Destination Address (IPv6)

During the import process, ControlDesk checks the validity of the IP/MAC addresses. Failures during the import are prompted to the user and written to the log file.

Related topics

References

Add/Edit Monitor Dialog..... 201

Load All Bus Signals

Access

This command is available only if the signals of a bus system are not loaded (grayed out). You can access it via:

Ribbon	None
Context menu of	▪ Bus Navigator tree - system node ▪ Bus Navigator tree - hardware node
Shortcut key	None
Icon	None

This command is available for:

- CAN (bus communication modeled with the CAN MultiMessage Blockset)
- FlexRay (bus communication modeled with the FlexRay Configuration Package)
- LIN (bus communication modeled with the LIN MultiMessage Blockset)

Purpose

To load all the bus signals of a bus system to the bus configuration.

Result

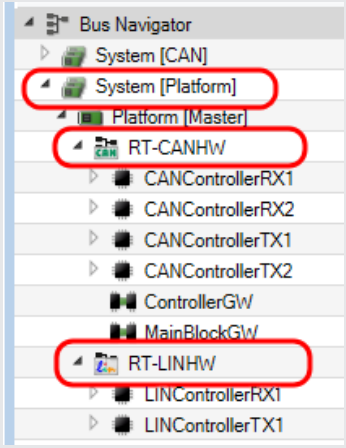
All the bus signals of the selected bus system are loaded to the bus configuration.

Description

The Bus Navigator lets you load and unload bus signals. This lets you control the memory consumption of ControlDesk for bus configurations containing many bus signals.

Tip

Depending on the selected node, you can load or unload the bus signals of a complete bus configuration or of a specific bus system.



ControlDesk lets you specify the default start-up behavior for loading bus signals on the Bus Navigator Page.

Related topics

References

Bus Navigator Page..... 207
Unload All Bus Signals..... 253

Locate Generated Layout

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - PDU node
Shortcut key	None
Icon	None

Purpose

To locate open layouts that contain an instrument for the selected message/frame/PDU.

Result

The result depends on whether there are any layouts containing instruments for the selected PDU:

- One layout: ControlDesk activates the corresponding layout.
- Several layouts: ControlDesk displays a list of all corresponding layouts and lets you choose which one to activate.
- None: ControlDesk displays a message informing you that no layouts containing instruments generated for the searched item could be found.

Related topics

References

Generate Gateway Layout..... 229
Generate Inspection Layout..... 230
Generate Layout..... 231
Generate Manipulation Layout..... 231
Generate RX Layout..... 232
Generate TX Layout..... 234
Generate TX Status Layout..... 235

Monitoring List

Access

A monitoring list is created in the working area when you execute the following command: Add (CAN/LIN/Ethernet) Monitor on page 193.

Purpose

To display the raw or physical data of the monitored messages/frames/PDUs.

Note

Without a variable description file, only the monitoring of raw data is possible.

Description

The monitoring list displays the monitored messages/frames/PDUs:

- As raw data
- As data in a physical format if a variable description file with computation methods was added to the platform or device performing the monitoring

Columns For information on the available columns, refer to Columns on page 243.

For information on how to customize the columns, refer to Customizing the columns on page 248.

Toolbar The toolbar is displayed at the top of the monitoring list. For more information, refer to Toolbar on page 241.

Status bar The status bar is displayed at the bottom right of the monitoring list. For more information, refer to Status bar on page 248.

Time cursor in the monitoring list If the time cursor is enabled in ControlDesk, it is visible in the bus monitoring list. The row that is related to the selected time position is highlighted. Selecting another row also moves the time cursor position. Refer to How to View Values at a Specific Point in Time (Time Cursor) (ControlDesk Measurement and Recording).

Toolbar

The toolbar consists of the following buttons:

Button	Description
	Starts bus monitoring by the monitor node that is associated to the monitoring list.
	Freezes the visualization of the displayed information without stopping the monitoring. To enable visualization again, click .
	Stops bus monitoring by the monitor node that is associated with the monitoring list.
	Deletes all entries from the monitoring list.
Continuous View	Selects a view for the monitoring list: ▪ Static View: In static view mode, the monitoring list contains one row for each message/frame, and the row is updated with the latest received data. ▪ Continuous View: In continuous view mode, every new instance of a message/frame is added to the monitoring list in a new row. ▪ Hierarchical View by ID: With hierarchical sorting by message/frame ID, the monitoring list is displayed in a tree view with the ID entries as the top level. Every new instance of a message/frame is added to the list in a new row. Hierarchical sorting by message/frame ID sorts message/frame instances according to their IDs.
	(Available only if monitoring is stopped or paused) To load an existing log file. When you load a log file that consists of several appended data blocks, only the last data block is

Button	Description
	loaded and displayed in the monitoring list, refer to Add (CAN/LIN/Ethernet) Logger on page 191.
	(Available only if monitoring is stopped or paused) To save the current monitoring list to a log file, which you can use for postprocessing or replay. If there are bus statistics, the **Save as Logfile** dialog lets you specify whether to export the bus statistics. For more information, refer to Save on page 249.
Tail Scrolling	Specifies whether to scroll the lines of the monitoring list up or down when incoming lines are added to the visible part of the list in the continuous view. ▪ Tail scrolling: The lines are scrolled up. If you have sorted the list by time stamps, for example, the latest incoming messages are placed in the bottom lines and the older ones are scrolled up. ▪ Head scrolling: The lines are scrolled down. If you have sorted the list by time stamps, for example, the latest incoming messages are placed in the top lines and the older ones are scrolled down.
hex dec ID	Specifies whether to display IDs in hexadecimal or decimal format in the monitoring list.
hex dec Data	Specifies whether to display data bytes in hexadecimal or decimal format in the monitoring list.
CAN	Displays the selected column set and lets you switch to another one in the list. A column set specifies the columns that are visible in a monitoring list. A column set can be marked with the following additional symbols: ▪ A d symbol indicates that a column set is a predefined column set: If you change a predefined column set, you can reset it to the predefined settings later on. ▪ A star symbol indicates that a column set is the favorite column set for newly added monitoring lists:
	Opens the **Configure Column Sets** dialog. Refer to Configure Column Sets Dialog (Monitoring List) on page 214.
Name Resolution	(Available only for Ethernet monitoring if an IP/MAC address mapping list was imported.) Toggles between IP or MAC addresses and ECU names in the monitoring list. You can import an IP/MAC address mapping list when you add or edit an Ethernet

Button	Description
	monitor. Refer to Add/Edit Monitor Dialog on page 201.
Decoding View	(Available only for Ethernet monitoring) Shows or hides the decoding view at the bottom of the monitoring list. Refer to Decoding View on page 217.
Bus Statistics View	(Available only for CAN, CAN FD and LIN controller nodes in connection with CAN or LIN interfaces of a SCALEXIO system, MicroAutoBox III, or VEOS.) Specifies whether to display bus statics values in the monitoring list. Refer to How to Display Bus Statistics on page 97.

Columns

The following columns are available (sorted by categories):

General This category contains the following columns:

Column	Bus	Description
Application Time	CAN LIN Ethernet	The time stamps are relative to the time at which the simulation application was started.
BRS	CAN FD	Displays whether the bit rate switch (BRS) is set.
Bus Status	CAN	The following states are possible: ▪ Active: Normal state. The microcontroller reacts to bus errors by outputting an error frame consisting of 6 dominant bits. ▪ Warning: The microcontroller has reached the warning stage (96 error frames). This is displayed in the Bus Statistics view. The controller is still *Error Active*. ▪ Passive: The controller has entered the *Error Passive* state. It reacts to bus errors with an error frame consisting of 6 recessive bits. Refer to the CAN specification for further details on this behavior. ▪ Off: The microcontroller has been disconnected from the bus. It will not be reactivated until a restart has been executed.
Checksum	LIN	Displays the LIN checksum value.
Checksum Model	LIN	Displays the LIN checksum model.
Controller ID	CAN LIN Ethernet	Displays the identifier of the controller that sends the message/frame.
Controller Name	CAN LIN Ethernet	Displays the name of the controller.
Controller Time	Ethernet	Displays the time at which the packet was received. The time stamps are relative to the time at which the controller was started.

Column	Bus	Description
Count	CAN LIN Ethernet	Displays the count of a message/frame/PDU.
Data	CAN LIN	Displays the signal value(s) (with unit, if available) of the message/frame instance.
Database ID	CAN LIN	Displays the database identifier.
Database Name	CAN LIN	Displays the database name.
Date	CAN LIN Ethernet	Displays the date of the message/frame/PDU instance.
Delta Time	CAN LIN Ethernet	Displays the difference between the receipt time of a message/frame/PDU instance and the receipt time of the previous message/frame/PDU instance *with the same ID* received by the same communication controller.
Delta Time (Last Frame)	CAN	Displays the difference between the receipt time of a message/frame/PDU instance and the receipt time of the previous message/frame/PDU instance, *irrespective of the message ID* but *received by the same communication controller*.
Dir	CAN LIN	Displays the message/frame/PDU direction (TX or RX).
DLC	CAN LIN	Data length code: Displays the data length of the message/frame/PDU instance. For signals, the bit length is displayed.
Event	LIN	Displays bus status information from the RTI LIN MultiMessage Blockset: Wake up, Sleep, Header error, Bus error, System error.
Event Detail	LIN	Displays additional information on the event: Wake up signal received, Header received, Sleep command received, No bus activity, Forced by internal function call, Header sync field error, Header PID parity error, Header bit error, Extra bytes detected, Data lost.
Frame Length	Ethernet	Displays the frame length of the frame instance.
Frame Type	Ethernet	Displays the type of the frame instance.
ID	CAN LIN	Displays the message/frame ID.
ID Format	CAN	Displays whether the extended ID format (29 bits) or the standard ID format (11 bits) is used for CAN or CAN FD messages and whether the bit rate switch (BRS) is set.
Identifier	CAN LIN Ethernet	Displays the identifier of the message/frame/PDU, together with the controller ID and the message/frame/PDU direction.
Measurement Time	CAN LIN Ethernet	Displays the time at which the message/frame/PDU instance was received relative to the measurement start time.

Column	Bus	Description
Monitor Time	CAN LIN Ethernet	Displays the time at which the message/frame/PDU instance was received relative to the monitoring start time.
Name	CAN LIN	Displays the message/frame/PDU name.
Platform Name	CAN LIN Ethernet	Displays the platform name.
Protocol	Ethernet	Displays the protocol used for the monitored communication.
Raw frame	Ethernet	Displays the raw data of a frame.
Service ID	CAN LIN	Displays the service ID.
Status	LIN	Displays the LIN frame status: OK, Checksum error, Slave not responding, Data lost, Framing error.
Summary	Ethernet	Displays brief information on the frame.
Synchronization Time	CAN LIN Ethernet	Displays the time at which the message/frame/PDU instance was received relative to the PC start time.
Time	CAN LIN Ethernet	Displays the PC system time at which the message/frame/PDU instance was received.
Type	LIN	Displays the frame type: Unconditional, Event-triggered, Sporadic, Diagnostic.

Ethernet This category contains the following columns:

Column	Bus	Description
802.1Q Tag	Ethernet	Displays the IEEE 802.1Q tag (if applicable).
Checksum	Ethernet	Displays the frame check sequence value (32-bit CRC checksum).
Destination MAC	Ethernet	Displays the MAC address of the destination interface.
EtherType	Ethernet	Displays the content of the EtherType field.
Ethernet Payload	Ethernet	Displays the payload of the Ethernet frame.
Source MAC	Ethernet	Displays the MAC address of the source interface.

IPv4 This category contains the following columns:

Column	Bus	Description
Destination Address	Ethernet	Displays the IPv4 address of the receiver of the packet.
Differentiated Services Code Point (DSCP)	Ethernet	Displays the DSCP value in the differentiated services field.
Explicit Congestion Notification (ECN)	Ethernet	Displays the value of the ECN bits in the differentiated services field (optional).
Flags	Ethernet	Displays the values of the three flag bits.
Fragment Offset	Ethernet	Displays the size of the fragment offset.

Column	Bus	Description
Header Checksum	Ethernet	Displays the value of the header checksum.
Identification	Ethernet	Displays the content of the Identification field.
IHL	Ethernet	Displays the content of the Internet Header Length field.
IPv4 Payload	Ethernet	Displays the payload of the packet.
Options	Ethernet	Displays the content of the Options field of the IP packet.
Protocol	Ethernet	Displays the protocol corresponding to the payload of the current packet.
Source Address	Ethernet	Displays the IPv4 address of the sender of the packet.
Time To Live	Ethernet	Displays the content of the Time To Live field of the IP packet.
Total Length	Ethernet	Displays the length of the entire IP packet in bytes.
Version	Ethernet	Displays the IP version.

IPv6 This category contains the following columns:

Column	Bus	Description
Destination Address	Ethernet	Displays the destination address of the packet.
Flow Label	Ethernet	Displays the content of the Flow Label field.
Hop Limit	Ethernet	Displays the content of the Hop Limit field.
IPv6 Payload	Ethernet	Displays the payload of the packet.
Next Header	Ethernet	Displays the content of the Next Header field.
Payload Length	Ethernet	Displays the size of the payload in octets.
Source Address	Ethernet	Displays the source address of the packet.
Traffic Class	Ethernet	Displays the traffic class values of the packet.
Version	Ethernet	Displays the IP version.

TCP/IP This category contains the following columns:

Column	Bus	Description
Acknowledgment Number	Ethernet	Displays the value of the acknowledgment number (if the ACG flag is set).
Checksum	Ethernet	Displays the value of the checksum.
Data Offset	Ethernet	Displays the size of the TCP header.
Destination Port	Ethernet	Displays the receiving port.
Flags	Ethernet	Displays the values of the flags (control bits).
Options	Ethernet	Displays the options of the packet.
Reserved	Ethernet	Displays the values of the reserved bits.
Sequence Number	Ethernet	Displays the value of the sequence number.
Source Port	Ethernet	Displays the sending port.
TCP/IP Payload	Ethernet	Displays the payload of the packet.
Urgent Pointer	Ethernet	Displays the value of the urgent pointer (if the URG flag is set).

Column	Bus	Description
Window Size	Ethernet	Displays the size of the receive window.

UDP/IP This category contains the following columns:

Column	Bus	Description
Checksum	Ethernet	Displays the value of the checksum.
Destination Port	Ethernet	Displays the receiving port.
Length	Ethernet	Displays the length of the UDP header and UDP data in bytes.
Source Port	Ethernet	Displays the sending port.
UDP/IP Payload	Ethernet	Displays the payload of the packet.

SOME/IP To display information in the SOME/IP columns, you have to specify ports to decode the SOME/IP-SD PDUs in the **Manage Ports** dialog. Refer to Bus Navigator Page on page 207.

This category contains the following columns:

Column	Bus	Description
Client ID	Ethernet	Displays the value of the Client ID bits.
Interface Version	Ethernet	Displays the version of the service interface.
Length	Ethernet	Displays the packet length.
Message Type	Ethernet	Displays the message type.
Method ID/Event ID	Ethernet	Displays the value of the method or event ID (0-32767 for methods, 32768-65535 for events).
Protocol Version	Ethernet	Displays the protocol version.
Return Code	Ethernet	Displays the return code, which is used to signal whether a request was successfully processed.
Service ID	Ethernet	Displays the service ID.
Session ID	Ethernet	Displays the session ID.

SOME/IP–SD To display information in the SOME/IP columns, you have to specify ports to decode the SOME/IP-SD PDUs in the **Manage Ports** dialog. Refer to Bus Navigator Page on page 207.

This category contains the following columns:

Column	Bus	Description
Entries Array	Ethernet	Displays the content of the Entries Array field.
Flags	Ethernet	Displays the values of the flag bits.
Length of Entries Array	Ethernet	Displays the length of the Entries Array field (in bytes).
Length of Options Array	Ethernet	Displays the length of the Options Array field (in bytes).
Options Array	Ethernet	Displays the content of the Options Array field.

Column	Bus	Description
Reserved	Ethernet	Displays the values of the reserved bits.

Null/Loopback This category contains the following columns:

Column	Bus	Description
Family	Ethernet	Displays the protocol/address family.

Customizing the columns

You can customize the columns of the monitoring list as follows:

- You can sort the list entries by clicking a column header.
- You can change the column sequence by dragging column headers and change the column width by dragging column borders.
- You can choose which columns are displayed. Refer to Delete Column (Monitoring List) on page 223 and Configure Column Sets Dialog (Monitoring List) on page 214.
- You can save your column selection. Refer to Set as Favorite Column Set (Monitoring List) on page 249.

Status bar

The status bar is displayed at the bottom right of the monitoring list.

Depending on your setup, the following information is displayed in the status bar from left to right:

- Total monitored time
- Current state (stopped, running, frozen)
- Frames per second as measured on the bus
- Number of total monitored frames
- Bus load (available only if you are using a bus monitoring device in connection with a CAN or LIN interface of a SCALEXIO system, MicroAutoBox III, or VEOS.)
- Buffer size
- Fill status of the buffer

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77

References

Add (CAN/LIN/Ethernet) Monitor.....193
Decoding View.....217

Save

Access

You can access this command via:

Ribbon	None
Context menu of	Monitoring list
Shortcut key	None
Icon	None

This command is available for:

- CAN
- LIN
- Ethernet

Purpose

(Available only if monitoring is stopped) To save the current monitoring data buffer to a log file which you can use for postprocessing and replay.

Result

ControlDesk opens the **Save as Logfile** dialog, which lets you specify where and in which file format the log file is saved.

If there are bus statistics, you can also specify whether to export the bus statistics. For ASC log files, the time axis selection is restricted to Application Time.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Basics on Replaying CAN and Ethernet Bus Communication.....99

References

Monitoring List.....240

Set as Favorite Column Set (Monitoring List)

Access

You can access this command via:

Ribbon	None
Context menu of	Monitoring list (column header)

Shortcut key	None
Icon	None

Purpose

To use the column set of the selected monitoring list as the default column set.

Description

ControlDesk uses the current column set as a default setting for new monitoring lists of the same bus type. The default setting contains the column selection, the column sequence, and the column width.

If you want to specify a global-specific column setting, you must use the **Bus Navigator** page in the **ControlDesk Options** dialog. Refer to Bus Navigator Page on page 207.

Result

The next time you add a monitoring list for the same bus type, it uses the new default settings.

Related topics

References

Bus Navigator Page..... 207
Configure Column Sets Dialog (Monitoring List)..... 214

Start (Logger)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - active logger node
Shortcut key	None
Icon	None

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To start or stop data logging of the selected logger node.

You can only start logging if the logger is active. An active logger node is displayed in yellow in the Bus Navigator tree: .

Result

When the command is selected, the selected logger node starts data logging and the property is displayed on the logger node in the Bus Navigator tree.

Description

Data logging starts according to the specified logger settings. You can edit the logger settings via the **Edit Logger** dialog. Refer to Edit (Logger) on page 224.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication..... 77
Bus Configuration Structure in the Bus Navigator Tree..... 19

References

Activate (Logger)..... 186
Add (CAN/LIN/Ethernet) Logger..... 191
Edit (Logger)..... 224

Start (Monitor)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - monitor node
Shortcut key	None
Icon	/ (in a monitoring list)

This command is available for:

- CAN
- LIN
- Ethernet (monitoring device in connection with a PC-based interface or a bus interface of SCALEXIO, MicroAutoBox III, or VEOS)

Purpose

To start or stop bus monitoring of the selected monitor node.

Note

The **Start Monitor** command automatically starts monitoring for all the lower hierarchy monitor nodes that are activated.

Result

Bus monitoring starts for the selected monitor node and the property is displayed on the monitor node in the Bus Navigator tree.

Description

Bus monitoring starts according to the settings specified via the **Add/Edit Monitor** dialog. Refer to Add/Edit Monitor Dialog on page 201.

Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication.....77
Bus Configuration Structure in the Bus Navigator Tree.....19

HowTos

How to Monitor Bus Communication.....82

References

Activate (Monitor).....187
Add (CAN/LIN/Ethernet) Monitor.....193
Edit (Monitor).....225

Start (Replay)

Access

You can access this command via:

Ribbon	None
Context menu of	Bus Navigator tree - replay node
Shortcut key	None
Icon	None

This command is available for:

- CAN
- Ethernet

Purpose

To start or stop replaying logged data of the selected replay node.

You can only start replaying if the replay is active. An active replay node is displayed in yellow in the Bus Navigator tree: .

Result

When the command is selected, the selected replay node starts replaying the logged data and the property is displayed on the replay node in the Bus Navigator tree.

Description

Replaying the logged data of the selected replay node starts according to the specified replay settings. You can edit the replay settings via the **Edit Replay** dialog. Refer to Add/Edit Replay Dialog on page 203.

Related topics

Basics

Basics on Replaying CAN and Ethernet Bus Communication..... 99
Bus Configuration Structure in the Bus Navigator Tree..... 19

References

Activate (Replay)..... 188
Add (CAN/Ethernet) Replay..... 189
Edit (Replay)..... 226

Unload All Bus Signals

Access

You can access this command via:

Ribbon	None
Context menu of	▪ Bus Navigator tree - system node ▪ Bus Navigator tree - hardware node
Shortcut key	None
Icon	None

This command is available for:

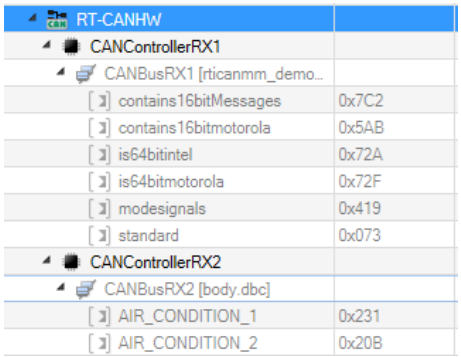
- CAN (bus communication modeled with the RTI CAN MultiMessage Blockset)
- FlexRay (bus communication modeled with the FlexRay Configuration Package)
- LIN (bus communication modeled with the RTI LIN MultiMessage Blockset)

Purpose

To unload all bus signals of a bus system from the bus configuration.

Result

All the bus signals of the selected bus system are unloaded from the bus configuration. PDUs with unloaded signals are displayed in gray.

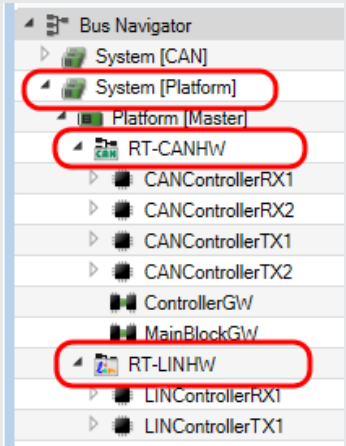


Description

The Bus Navigator lets you load and unload bus signals. This lets you control the memory consumption of ControlDesk for bus configurations containing many bus signals.

Tip

Depending on the selected node, you can load or unload the bus signals of a complete bus configuration or of a specific bus system.



ControlDesk lets you specify the default start-up behavior for loading bus signals on the Bus Navigator Page.

Related topics

References

Bus Navigator Page..... 207
Load All Bus Signals..... 238

Automation

Where to go from here

Information in this section

Programming ControlDesk Automation.....256
Bus Navigator and Bus Instrument Handling.....260

Programming ControlDesk Automation

Where to go from here

Information in this section

Automating the Bus Navigator.....256
ControlDesk's Bus Navigator lets you handle bus communication on dSPACE platforms and on PC-based interfaces supported by ControlDesk. For example, you can monitor, log and replay bus communication, and manipulate signals via specific Bus Instruments.

Information in other sections

Tool Automation Demos (ControlDesk Automation 🕮)
Demonstrate how to automate ControlDesk and use ControlDesk events.

Automating the Bus Navigator

Introduction

ControlDesk's Bus Navigator lets you handle bus communication on dSPACE platforms and on PC-based interfaces supported by ControlDesk. For example, you can monitor, log and replay bus communication, and manipulate signals via specific Bus Instruments.

Unless otherwise indicated, the program listings below consist of excerpts from the `BusNavigatorHandling.py` demo script. The demo uses the Bus Navigator Demo on page 24.

Configuring and starting bus communication replay

The following listing shows you how to configure the replay node in the `transceiverSystem` node of the Bus Navigator.

```
class MainDemoController(object):
   (...)
   def Replay(self):
      busNavigator = self.ControlDeskApplication.BusNavigator
      receiverSystem = busNavigator.Systems.Item(1)
      transceiverSystem = busNavigator.Systems.Item(0)
      for replay in transceiverSystem.GetAllReplays(self.Enums.BusType.CAN):
         if replay.State == self.Enums.RunState.Running:
            replay.Stop()
         replay.Activated = False
         replay.Configuration.Mode = self.Enums.ReplayMode.Infinite
         replay.Activated = True
         replay.Start()
```

Configuring and starting bus communication monitoring

The following listing shows you how to configure the monitor node in the `receiverSystem` node of the Bus Navigator.

```
class MainDemoController(object):
   (...)
   def Monitor(self):
      busNavigator = self.ControlDeskApplication.BusNavigator
      receiverSystem = busNavigator.Systems.Item(1)
      transceiverSystem = busNavigator.Systems.Item(0)
      for monitor in receiverSystem.GetAllMonitors(self.Enums.BusType.CAN):
         if monitor.State == self.Enums.RunState.Running:
            monitor.Stop()
         monitor.Configuration.UpdateRate = 200
         monitor.Configuration.BufferSize = 20000
         monitor.Start()
```

Configuring and starting bus communication logging

The following listing shows you how to configure the logger node in the `receiverSystem` node of the Bus Navigator.

```
class MainDemoController(object):
   (...)
   def Logger(self):
      busNavigator = self.ControlDeskApplication.BusNavigator
      receiverSystem = busNavigator.Systems.Item(1)
      transceiverSystem = busNavigator.Systems.Item(0)
      for logger in receiverSystem.GetAllLoggers(self.Enums.BusType.CAN):
         if logger.State == self.Enums.RunState.Running:
            logger.Stop()
         logger.Activated = False
         logger.Configuration.LogFileFullPath = self.ControlDeskApplication.ActiveExperiment.
                                                DirectoryName +  r'\Bus Replays\demo_log.asc'
         logger.Configuration.Duration = 10.0
         logger.Configuration.OverwriteMode = True
         logger.Activated = True
         logger.Start()
```

Creating a custom Bus Instrument

The following listing shows you how to create a Bus Instrument.

```
class MainDemoController(object):
   (...)
   def BusInstrument(self):
      # Add a Bus Instrument to the currently active layout
      instrument = self.ControlDeskApplication.LayoutManagement.ActiveLayout.Instruments.
                   Add("Bus Navigator Instrument", "Instrument_demo", 0, 0, 800, 400)
      # Create a subinstrument
      subInstrument = instrument.SubInstruments.Item(0)
      subInstrument.Name = "instrument automation demo"
      # Add a grid with 3 columns
      grid = subInstrument.Grids.Add()
      grid.Columns.Add()
      grid.Columns.Add()
      grid.Columns.Add()
      # Add a row
      row = grid.Rows.Add()
      # Change the cell type
      cell1 = row.ChangeCellType(0,1)
      cell2 = row.ChangeCellType(1,2)
      cell3 = row.ChangeCellType(2,3)
      cell1.Text = 'Test'
      grid.OptimizeHeight()
```

Generating layouts with Bus Instruments

The following listing shows you how to generate a TX status layout and layouts for all the TX and RX messages of a CAN controller's communication matrix. The generated layout objects are added to a list.

Generating a layout means inserting a new layout in the experiment and placing a Bus Instrument of the appropriate type on it:

- TX layout: Bus Instrument (TX type)
- RX layout: Bus Instrument (RX type)
- TX status layout: Bus Instrument (TX status type for CAN/LIN/FlexRay)
- Error status layout: Bus Instrument (Error status type for FlexRay)

```
class MainDemoController(object):
   (...)
   def    Init   (self):
   (...)
   # List of layouts
   self.BNVLayouts = []
   (...)
   def GenerateLayouts(self):
      # Generate layouts from matrices
      matrix = self.ControlDeskApplication.BusNavigator.Systems[0].BusPlatforms[0].
               CANBusSystem.PhysicalBusAccesses[0].CommunicationMatrices[0]
      self.BNVLayouts.append(matrix.GenerateStatusLayout())
      for i in matrix.NetworkNodes[0].CommunicationChannels[0].LPDUs:
         #Generate layouts from TX-PDU
         if i.TxIPDUs.Count > 0:
         self.BNVLayouts.append(i.TxIPDUs[0].GenerateLayout())
         # Generate layouts from RX-PDU
         if i.RxIPDUs.Count > 0:
            self.BNVLayouts.append(i.RxIPDUs[0].GenerateLayout())
```

Extending the context menu of Bus Navigator tree nodes

To execute custom commands or scripts, you can extend the context menu of Bus Navigator tree nodes.

Refer to Extending the Context Menu of Elements (ControlDesk Customization 🕮).

Related topics

Basics

Bus Navigator Demo.....24
Introduction to the Bus Navigator.....14
Tool Automation Demos (ControlDesk Automation 🕮)

Bus Navigator and Bus Instrument Handling

Where to go from here

Information in this section

Bus Navigator and Bus Instrument Handling Interfaces.....260
The Bus Navigator lets you handle bus messages. You can manipulate messages, frames, and PDUs before transmission, exclude them from being transmitted, etc.

Bus Navigator-Related Interfaces.....260
The Bus Navigator lets you handle bus messages, such as CAN messages, LIN frames, and FlexRay PDUs.

Bus Instrument-Related Interfaces.....261
The Bus instrument lets you display information on received messages, and transmit messages.

Bus Navigator and Bus Instrument Handling Interfaces

Introduction

The Bus Navigator lets you handle bus messages. You can manipulate messages, frames, and PDUs before transmission, exclude them from being transmitted, etc.

Bus Navigator-Related Interfaces

Introduction

The Bus Navigator lets you handle bus messages, such as CAN messages, LIN frames, and FlexRay PDUs.

Description

The Bus Navigator object implements the *IBnBusNavigator* interface. It is used to access the bus configuration in the Bus Navigator tree.

Related interfaces

Interface	Description
IBnBusNavigator (refer to BusNavigator / IBnBusNavigator <<Interface>> (ControlDesk Automation 🕮))	Root interface of BusNavigator Automation. Provides access to the loaded bus configurations (systems).
IBnSystems (refer to Systems / IBnSystems <<Collection>> (ControlDesk Automation 🕮))	A collection without methods for modification.

Interface	Description
IBnSystem (refer to System / IBnSystem <<Interface>> (ControlDesk Automation 🕮))	Bus configuration of a system.
IBnBusPlatforms (refer to BusPlatforms / IBnBusPlatforms <<Collection>> (ControlDesk Automation 🕮))	A collection without methods for modification.
IBnMonitors (refer to Monitors / IBnMonitors <<Collection>> (ControlDesk Automation 🕮))	A collection without methods for modification.
IBnLoggers (refer to Loggers / IBnLoggers <<Collection>> (ControlDesk Automation 🕮))	A collection without methods for modification.
IBnLogger (refer to Logger / IBnLogger <<Interface>> (ControlDesk Automation 🕮))	Bus logger (generic for all bus systems).
IBnMonitor (refer to Monitor / IBnMonitor <<Interface>> (ControlDesk Automation 🕮))	Bus monitor (generic for all bus systems).

Related documentation

Topic	Description
Automating the Bus Navigator on page 256	ControlDesk's Bus Navigator lets you handle bus communication on dSPACE platforms and on PC-based interfaces supported by ControlDesk. For example, you can monitor, log and replay bus communication, and manipulate signals via specific Bus Instruments.

Bus Instrument-Related Interfaces

Introduction

The Bus instrument lets you display information on received messages, and transmit messages.

Description

The Bus Navigator instrument implements the *IBnBusInstrument* interface. You can use this object to manipulate Bus Instrument properties.

Related interfaces

Interface	Description
IBnBusInstrument (refer to BusInstrument / IBnBusInstrument <<Interface>> (ControlDesk Automation 🕮))	This interfaces provides access to the Bus Navigator instrument.

Interface	Description
IBnGrids (refer to Grids / IBnGrids <<Collection>> (ControlDesk Automation 🕮))	This interface provides access to the grid collection.
IBnGrid (refer to Grid / IBnGrid <<Interface>> (ControlDesk Automation 🕮))	This interface provides access to a grid, which is a table-like array of cells.

Related documentation

Topic	Description
Automating the Bus Navigator on page 256	ControlDesk's Bus Navigator lets you handle bus communication on dSPACE platforms and on PC-based interfaces supported by ControlDesk. For example, you can monitor, log and replay bus communication, and manipulate signals via specific Bus Instruments.

Troubleshooting

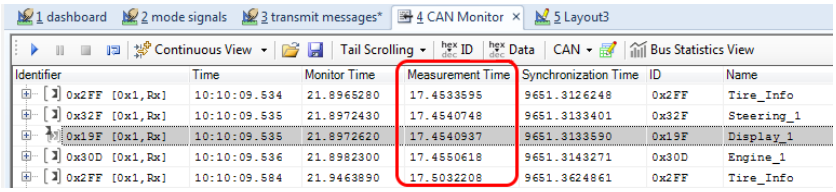
Synchronizing Bus Monitoring Data with Recorded Data

Problem

If you compare the measurement values of a recording with the measurement values of a bus monitoring, the measured values at the same measurement time stamp might be different.

Description

If you have monitored bus data simultaneously to a running measurement, the monitoring list can display the measurement time for each measured value. If the time cursor is enabled in ControlDesk, the row that is related to the selected time position is highlighted. The following illustration shows this for CAN bus monitoring.

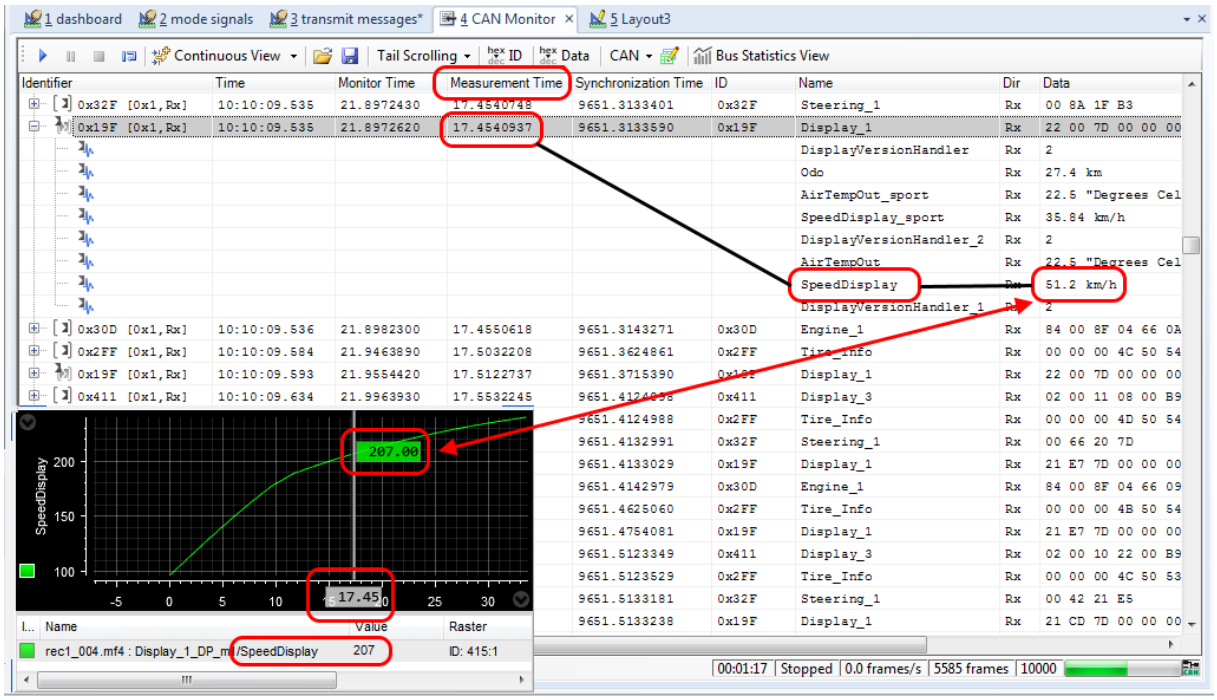


Identifier	Time	Monitor Time	Measurement Time	Synchronization Time	ID	Name
0x2FF [0x1,Rx]	10:10:09.534	21.8965280	17.4533595	9651.3126248	0x2FF	Tire_Info
0x32F [0x1,Rx]	10:10:09.535	21.8972430	17.4540748	9651.3133401	0x32F	Steering_1
0x19F [0x1,Rx]	10:10:09.535	21.8972620	17.4540937	9651.3133590	0x19F	Display_1
0x30D [0x1,Rx]	10:10:09.536	21.8982300	17.4550618	9651.3143271	0x30D	Engine_1
0x2FF [0x1,Rx]	10:10:09.584	21.9463890	17.5032208	9651.3624861	0x2FF	Tire_Info

If the **Measurement Time** column is not visible in the monitoring list, click to activate it via the **Configure Column Sets** dialog.

Suppose you have performed a recording after the measurement was running for a while. If you now open the measurement data file and compare signals that were simultaneously recorded and monitored, you might see different values at the same time stamps.

The following illustration shows an example. The recorded value is visible in a Time Plotter, the monitored value in the CAN Monitor monitoring list:



At time stamp 17.45 s, the recorded value for the `SpeedDisplay` signal (207.0 km/h) differs from the monitored value (51.2 km/h). This shows that monitoring and recording measurement times are not synchronized.

The problem occurs because the measurement time stamps in the recording start with 0 at the start of the recording, whereas the monitoring list uses the measurement time stamps from the running measurement that was started earlier.

There are two solutions for this problem.

- Solution 1: Saving the recording with the original measurement time stamps
- Solution 2: Adding an offset to the time stamps in the recorded data

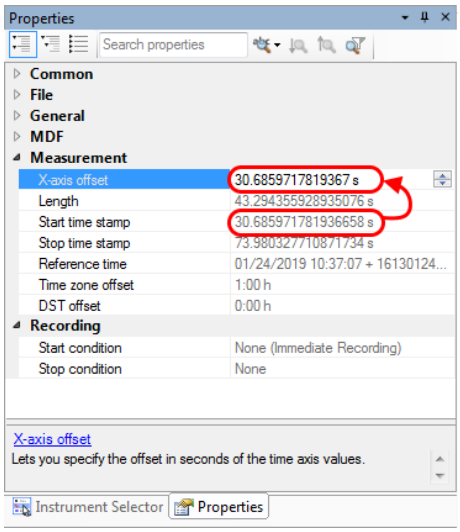
Solution 1: Saving the recording with the original measurement time stamps

Before you start the recording, activate Use measurement time stamps in the property list of the recorder. The recorder then uses the original time stamps of the running measurement.

Solution 2: Adding an offset to the time stamps in the recorded data

In the Project controlbar or in the Measurement Data Pool, right-click the measurement data file and select Properties to open the Properties controlbar.

Open the **Measurement** category, right-click the value of the **Start time stamp** property and copy it to the **X-axis offset** property.



Related topics

Basics

Basics on Monitoring, Logging, and Filtering Bus Communication..... 77

Limitations

Limitations for the Bus Navigator

Introduction	There are some limitations for the Bus Navigator in ControlDesk.
CAN: Limitations	**Log files with messages from several buses** Replaying log files with CAN messages from *several* CAN buses is not possible if the time stamps of the CAN messages do not increase monotonously. **Replay during high bus traffic** If there is high traffic on the CAN bus, some messages of the log file might not be replayed to avoid delaying the bus communication. **Replay on VEOS: Incorrect timing for real-time acceleration factors other than 1** If you use real-time acceleration factors other than 1, the timing of the replayed messages is not correct.
Ethernet: Limitations	**No filtering of replayed communication** You cannot filter replayed Ethernet communication with the Bus Navigator. **Replay on VEOS: Incorrect timing for real-time acceleration factors other than 1** If you use real-time acceleration factors other than 1, the timing of the replayed messages is not correct.
FlexRay: No display of signal update bit value	No signal update bit values are displayed for signals of an RX PDU.
LIN: Limitations in connection with real-time applications configured with the RTI LIN MultiMessage Blockset	**Controller-specific settings cannot be configured via ControlDesk** You cannot configure the following controller-specific settings via ControlDesk. You have to configure them using the RTI LIN MultiMessage Blockset. ▪ Baud rate ▪ Break length

- Break delimiter
- Transceiver type
- Termination

Tip

Except for the transceiver type, you can configure the controller-specific settings listed above via ControlDesk during run time if you add corresponding inports to the RTILINMM ControllerSetup block in the related Simulink model.

LIN monitoring: No display of the frame type for unconfigured LIN frames No frame type is displayed for monitored LIN frames if they were not configured via the RTI LIN MultiMessage Blockset.

LIN monitoring: No display of the frame type for sporadic frames The RTI LIN MultiMessage Blockset does not support sporadic frames.As a consequence, No frame type is displayed for monitored LIN frames if they are defined as *sporadic* in the related communication database.

No display of slave-not-responding errors for TX frames In connection with real-time applications configured with the RTI LIN MultiMessage Blockset, the slave-not-responding error cannot be displayed for TX frames – frames transmitted by dSPACE LIN hardware – even if the frame is disrupted, for example, by a bus error such as a short circuit.

Related topics

Basics

Introduction to the Bus Navigator..... 14

Glossary

Introduction

Briefly explains the most important expressions and naming conventions used in the ControlDesk documentation.

Where to go from here

Information in this section

Numerics....270
A....270
B....271
C....272
D....276
E....280
F....283
G....284
H....284
I....285
K....287
L....287
M....288
N....291
O....291
P....293
Q....295
R....296

S.....297
T.....300
U.....301
V.....302
W.....304
X.....305

Numerics

3-D Viewer An instrument for displaying items in a 3-D environment.

A

A2L file A file that contains all the relevant information on measurement and calibration variables in an ECU application and the ECU's communication interface(s). This includes information on the variables' memory addresses and conversion methods, the memory layout and data structures in the ECU as well as interface description data (IF_DATA).

Acquisition An object in the Measurement Configuration controlbar that specifies the variables to be measured and their measurement configuration.

Active variable description The variable description that is currently active for a platform/device. Multiple variable descriptions can be assigned to one platform/device, but only one of them can be active at a time.

Additional write variable A scalar parameter or writable measurement variable that can be connected to an instrument in addition to the main variable. When the value of the main variable changes, the changed value is also applied to all the additional write variables connected to the instrument.

Airspeed Indicator An instrument for displaying the airspeed of a simulated aircraft.

Altimeter An instrument for displaying the altitude of a simulated aircraft.

Animated Needle An instrument for displaying the value of a connected variable by a needle deflection.

Application image An image file that contains all the files that are created when the user builds a real-time application. It particularly includes the variable

description (SDF) file. To extend a real-time application, ControlDesk lets the user create an updated application image from a data set. The updated application image then contains a real-time application with an additional set of parameter values.

Artificial Horizon An instrument displaying the rotation on both the lateral and the longitudinal axis to indicate the angle of pitch and roll of a simulated aircraft. The Artificial Horizon has a pitch scale and a roll scale.

Automatic Reconnect Feature for automatically reconnecting to platform/device hardware, for example, when the ignition is turned off and on, or when the physical connection between the ControlDesk PC and the ECU is temporarily interrupted.

If the feature is enabled for a platform/device and if the platform/device is in the 'unplugged' state, ControlDesk tries to re-establish the logical connection to the platform/device hardware. After the logical connection is re-established, the platform/device has the same state as before the unplugged state was detected. A measurement started before the unplugged state was detected is resumed.

Automation A communication mechanism that can be used by various programming languages. A client can use it to control a server by calling methods and properties of the server's automation interface.

Automation script A script that uses automation to control an automation server.

Axis point object Common axis

B

Bar An instrument (or a value cell type of the Variable Array) for displaying a numerical value as a bar deflection on a horizontal or vertical scale.

Bitfield A value cell type of the Variable Array for displaying and editing the source value of a parameter as a bit string.

Bookmark A marker for a certain event during a measurement or recording.

Browser An instrument for displaying HTML and TXT files. It also supports Microsoft Internet Explorer© plug-ins that are installed on your system.

Bus communication replay A feature of the Bus Navigator that lets you replay logged bus communication data from a log file. You can add replay nodes

to the Bus Navigator tree for this purpose. You can specify filters to replay selected parts of the logged bus communication.

Bus configuration A configuration of all the controllers, communication matrices, and messages/frames/PDUs of a specific communication bus such as CAN. ControlDesk lets you display and experiment with bus configurations in the Bus Navigator.

Bus connection A mode for connecting dSPACE real-time hardware to the host PC via bus. The list below shows the possible bus connections:

- dSPACE real-time hardware installed directly in the host PC
- dSPACE real-time hardware installed in an expansion box connected to the host PC via dSPACE link board

Bus Instrument An instrument available for the Bus Navigator. It can be configured for different purposes, for example, to display information on received messages (RX messages) or to manipulate and transmit messages (TX messages). The instrument is tailor-made and displays only the message- and signal-specific settings which are enabled for display and/or manipulation by ControlDesk during run time.

Bus logging A feature of the Bus Navigator that lets you log raw bus communication data. You can add logger nodes on different hierarchy levels of the Bus Navigator tree for this purpose. You can specify filters to log filtered bus communication. The logged bus communication can be replayed.

Bus monitoring A feature of the Bus Navigator that lets you observe bus communication. You can open monitoring lists and add monitor nodes on different hierarchy levels of the Bus Navigator tree for this purpose. You can specify filters to monitor filtered bus communication.

Bus Navigator A controlbar for handling bus messages, such as CAN messages, LIN frames, and Ethernet packets.

Bus statistics A feature of the Bus Navigator that lets you display and log statistical information on the bus load during bus monitoring.

Bypassing A method for replacing an existing ECU function by running a new function.

C

Calculated variable A scalar variable that can be measured and recorded, and that is derived from one or more *input variables*.

The following input variable types are supported:

- Measurement variables
- Single elements of measurement arrays or value blocks
- Scalar parameters, or existing calculated variables

The value of a calculated variable is calculated via a user-defined *computation formula* that uses one or more input variables.

Calculated variables are represented by the symbol.

CalDemo ECU A demo program that runs on the same PC as ControlDesk. It simulates an ECU on which the Universal Measurement and Calibration (XCP) protocol and the Unified Diagnostic Services (UDS) protocol are implemented.

The CalDemo ECU allows you to perform parameter calibration, variable measurement, and ECU diagnostics with ControlDesk under realistic conditions, but without having to have a real ECU connected to the PC. Communication between the CalDemo ECU and ControlDesk can be established via XCP on CAN or XCP on Ethernet, and UDS on CAN.

Tip

If communication is established via XCP on Ethernet, the CalDemo ECU can also run on a PC different from the PC on which ControlDesk is running.

The memory of the CalDemo ECU consists of two areas called memory page. Each page contains a complete set of parameters, but only one page is accessible by the CalDemo ECU at a time. You can easily switch the memory pages of the CalDemo ECU to change from one parameter to another in a single step.

Two ECU tasks run on the CalDemo ECU:

- ECU task #1 runs at a fixed sample time of 5 ms. In ControlDesk's **Measurement Configuration**, ECU task #1 is related to the time-based **5 ms**, **10 ms**, **50 ms** and **100 ms** measurement rasters of the CalDemo ECU.
- ECU task #2 has a variable sample time. Whenever the CalDemo ECU program is started, the initial sample time is 5 ms. This can then be increased or decreased by using the **dSPACE CalDemo** dialog.

  ECU task #2 is related to the **extEvent** measurement raster of the CalDemo ECU.

The CalDemo ECU can also be used to execute diagnostic services and jobs, handle DTCs and perform measurement and calibration via ECU diagnostics.

The CalDemo ECU program is run by invoking `CalDemo.exe`. The file is located in the `.\Demos\CalDemo` folder of the ControlDesk installation.

Calibration Changing the parameter values of real-time application s or ECU application s.

Calibration memory segment Part of the memory of an ECU containing the calibratable parameters. Memory segments can be defined as `MEMORY_SEGMENT` in the A2L file. ControlDesk can use the segments to evaluate the memory pages of the ECU.

ControlDesk lets you perform the calibration of:

- Parameters inside memory segments
- Parameters outside memory segments
- Parameters even if no memory segments are defined in the A2L file.

CAN Bus Monitoring device A device that monitors the data stream on a CAN bus connected to the ControlDesk PC.

The CAN Bus Monitoring device works, for example, with PC-based CAN interfaces such as the DCI-CAN2 or the DCI-CAN/LIN1.

The device supports the following variable description file types:

- DBC
- FIBEX
- AUTOSAR system description (ARXML)

CANGenerator A demo program that simulates a CAN system, that is, it generates signals that can be measured and recorded with ControlDesk. The program runs on the same PC as ControlDesk.

The CANGenerator allows you to use the CAN Bus Monitoring device under realistic conditions, but without having to have any device hardware connected to the PC.

The CAN (Controller Area Network) protocol is used for communication between the CANGenerator and ControlDesk. However, since the CANGenerator runs on the same PC as ControlDesk, ControlDesk does not communicate with the device via a real CAN channel, but via a *virtual CAN channel* implemented on the host PC.

You can start the CAN generator program by running `CANGenerator.exe`. The file is located in the `.\Demos\CANGenerator` folder of the ControlDesk installation.

Capture A data packet of all the measurement variables assigned to a measurement raster. The packet comprises the data that results from a single triggering of the raster.

CCP Abbreviation of CAN Calibration Protocol. This protocol can be implemented on electronic control units (ECUs) and allows users to access ECUs with measurement and calibration systems (MCS) such as ControlDesk.

The basic features of CCP are:

- Read and write access to the ECU memory, i.e., providing access for calibration
- Synchronous data acquisition
- Flash programming for ECU development purposes

The CCP protocol was developed by ASAM e.V. (Association for Standardization of Automation and Measuring Systems e.V.). For the protocol specification, refer to http://www.asam.net.

The following device supports ECUs with an integrated CCP service:

- CCP device

CCP device A device that provides access to an ECU with CCP connected to the ControlDesk PC via CAN, for example, for measurement and calibration purposes via CCP (CAN Calibration Protocol).

Check Button An instrument (or a cell type of the Variable Array) for displaying whether the value of a connected variable matches predefined values or for writing a predefined value to a connected variable.

cmdloader A command line tool for handling applications without using the user interface of an experiment software.

Common axis A parameter that consists of a 1-dimensional array containing axis points. A common axis can be referenced by one or more curves and/or maps. Calibrating the data points of a common axis affects all the curves and/or maps referencing the axis.

Common axes are represented by the symbol.

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

```
%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>
```

or

```
%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>
```

Computation method A formula or a table that defines the transformation of a source value into a converted value (and vice versa). In addition to the computation methods defined in the variable description file, ControlDesk provides the *__Identity* computation method which means the converted and the source value are equal.

Connected A platform/device state defined by the following characteristics:

- A continuous logical connection is established between ControlDesk and the platform/device hardware.
- A platform/device must be in the 'connected' state before it can change to the 'measuring/recording' or 'online calibration started' state.
- Online calibration is impossible. ControlDesk did not yet adjust the memory segments containing calibration data in the platform/device and on the corresponding hardware. Offline calibration is possible.
- Platform/device configuration is not possible. However, you can invoke platform/device configuration for a platform/device that is in the connected state. ControlDesk temporarily sets the platform/device to the disconnected state.

The 'connected' platform/device state is indicated by the icon.

Connection mode dSPACE real-time systems can be installed within the host PC or connected to the host via a bus interface and/or via Ethernet. When the Ethernet is being used, different network clients might exist. The connection type being used and, in the case of Ethernet, the network client being used, determine the dSPACE systems that can be accessed.

Control primitive A special diagnostic communication object for changing communication states or protocol parameters, or for identifying (ECU) variants.

Controlbar A window or pane outside the working area. Can be docked to an edge of the main window or float in front of it. A controlbar can contain a

document, such as a layout, or a tool, such as the Bus Navigator. It can be grouped with other controlbars in a window with tabbed pages.

ControlDesk The main version of ControlDesk for creating and running experiments, and for accessing dSPACE real-time hardware and VEOS. The functionality can be extended by optional software modules.

ControlDesk - Operator Version A version of ControlDesk that provides only a subset of functionality for running existing experiments. The functionality can be extended by optional software modules.

ControlDesk Bus Navigator Module An optional software module for ControlDesk for handling bus messages, such as CAN, LIN, and FlexRay messages, frames, and PDUs and Ethernet packets.

ControlDesk ECU Diagnostics Module An optional software module for ControlDesk that facilitates the calibration and validation of ECU diagnostic functions.

ControlDesk ECU Interface Module An optional software module for ControlDesk for calibration and measurement access to electronic control units (ECUs). The module is also required for calibration and measurement access to virtual ECUs (V-ECUs) used in SIL testing scenarios.

ControlDesk Signal Editor Module An optional software module for ControlDesk for the graphical definition and execution of signal generators for stimulating model variables of real-time/offline simulation applications.

Controller board Single-board hardware computing the real-time application. Contains a real-time processor for fast calculation of the model and I/O interfaces for carrying out the control developments.

Conversion table A table that specifies the value conversion of a source value into a converted value. In the case of verbal conversion, the converted value is a string that represents one numerical value or a range of numerical values.

Conversion type The type of a computation method, for example a linear function or a verbal computation method.

Curve A parameter that consists of

- A 1-dimensional array containing the axis points for the x-axis. This array can also be specified by a reference to a common axis.
- Another 1-dimensional array containing data points. The curve assigns one data point to each axis point.

Curves are represented by the symbol.

D

DAQ module A hardware module for the acquisition of physical quantities

Data Cursor One or two cursors that are used to display the values of selected chart positions in a Time Plotter or an Index Plotter.

Data logger An object in the Measurement Configuration controlbar that lets you configure a data logging.

Data logger signal list A list that contains the variables to be included in subsequent data loggings on real-time hardware.

Data logging The recording of data on dSPACE real-time hardware that does not require a physical connection between the host PC and the real-time hardware. In contrast to flight recording, data logging is configured in ControlDesk.

Data set A set of the parameters and their values of a platform/device derived from the variable description of the platform/device. There are different types of data sets:

- Reference data set
- Sub data set
- Unassigned data set
- Working data set

DCI-CAN/LIN1 A dSPACE-specific interface between the host PC and the CAN/CAN FD bus and/or LIN bus. The DCI-CAN/LIN1 transfers messages between the CAN-/LIN-based devices and the host PC via the universal serial bus (USB).

DCI-CAN2 A dSPACE-specific interface between the host PC and the CAN bus. The DCI-CAN2 transfers CAN and CAN FD messages between the CAN-based devices and the host PC via the universal serial bus (USB).

DCI-GSI2 Abbreviation of *dSPACE Communication Interface - Generic Serial Interface 2*. A dSPACE-specific interface for ECU calibration, measurement and ECU interfacing.

DCI-GSI2 device A device that provides access to an ECU with DCI-GSI2 connected to the ControlDesk PC for measurement, calibration, and bypassing purposes via the ECU's debug interface.

DCI-KLine1 Abbreviation of *dSPACE Communication Interface - K-Line Interface*. A dSPACE-specific interface between the host PC and the diagnostics bus via K-Line.

Debug interface An ECU interface for diagnostics tasks and flashing.

Default raster A platform-/device-specific measurement raster that is used when a variable of the platform/device is connected to a plotter or a recorder, for example.

Deposition definition A definition specifying the sequence in which the axis point values of a curve or map are deposited in memory.

Device A software component for carrying out calibration and/or measurement, bypassing, ECU flash programming, or ECU diagnostics tasks.

ControlDesk provides the following devices:

- Bus devices:
  - CAN Bus Monitoring device
  - Ethernet Bus Monitoring device
  - LIN Bus Monitoring device
- ECU Diagnostics device
- GNSS device
- Measurement and calibration devices:
  - CCP device
  - DCI-GSI2 device
  - XCP on CAN device
  - XCP on Ethernet device

Each device usually has a variable description that specifies the device's variables to be calibrated and measured.

Diagnostic interface Interface for accessing the fault memory of an ECU.

Diagnostic job (often called Java job) Programmed sequence that is usually built from a sequence of the diagnostic service. A diagnostic job is either a single-ECU job or a multiple-ECU job, depending on whether it communicates with one ECU or multiple ECUs.

Diagnostic protocol A protocol that defines how an ECU communicates with a connected diagnostic tester. The protocol must be implemented on the ECU and on the tester. The diagnostics database specifies the diagnostic protocol(s) supported by a specific ECU.

ControlDesk's ECU Diagnostics device supports CAN and K-Line as the physical layers for communication with an ECU connected to the ControlDesk PC. For information on the supported diagnostic protocols with CAN and K-Line, refer to Basics of ECU Diagnostics with ControlDesk (ControlDesk ECU Diagnostics).

Diagnostic service A service implemented on the ECU as a basic diagnostic communication element. Communication is performed by selecting a service, configuring its parameters, executing it, and receiving the ECU results. When a service is executed, a defined request is sent to the ECU and the ECU answers with a specific response.

Diagnostic trouble code (DTC) A hexadecimal index for the identification of vehicle malfunctions. DTCs are stored in the fault memory of ECUs and can be read by diagnostic testers.

Diagnostics database A database that completely describes one or more ECUs with respect to diagnostics communication. ControlDesk supports the ASAM MCD-2 D ODX database format, which was standardized by ASAM e.V. (Association for Standardisation of Automation and Measuring Systems e.V.). For the format specification, refer to http://www.asam.net.

Proprietary diagnostics database formats are not supported by ControlDesk.

Diagnostics Instrument An instrument for communicating with an ECU via the diagnostic protocol using diagnostic services, diagnostic jobs, and control primitives.

Disabled A platform/device state defined by the following characteristics:

- No logical connection is established between ControlDesk and the platform/device hardware.
- When a platform/device is disabled, ControlDesk does not try to establish the logical connection for that platform/device. Any communication between the platform/device hardware and ControlDesk is rejected.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.

The 'disabled' platform/device state is indicated by the icon.

Disconnected A platform/device state defined by the following characteristics:

- No logical connection is established between ControlDesk and the platform/device hardware.
- When a platform/device is in the disconnected state, ControlDesk does not try to re-establish the logical connection for that platform/device.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.

The 'disconnected' platform/device state is indicated by the icon.

Display An instrument (or a value cell type of the Variable Array) for displaying the value of a scalar variable or the text content of an ASCII variable.

Documents folder A standard folder for user-specific documents.

```
%USERPROFILE%\Documents\dSPACE\<ProductName>\
<VersionNumber>
```

DS1006 Processor Board platform A platform that provides access to a DS1006 Processor Board connected to the host PC for HIL simulation and function prototyping purposes.

DS1007 PPC Processor Board platform A platform that provides access to a single multicore DS1007 PPC Processor Board or a DS1007 multiprocessor system consisting of two or more DS1007 PPC Processor Boards, connected to the host PC for HIL simulation and function prototyping purposes.

DS1104 R&D Controller Board platform A platform that provides access to a DS1104 R&D Controller Board installed in the host PC for function prototyping purposes.

DS1202 MicroLabBox platform A platform that provides access to a MicroLabBox connected to the host PC for function prototyping purposes.

DsDAQ service A service in a real-time application or offline simulation application (OSA) that provides measurement data from the application to the

host PC. Unlike the host service, the DsDAQ service lets you perform, for example, triggered measurements with complex trigger conditions.

The following platforms support applications that contain the DsDAQ service:

- DS1007 PPC Processor Board platform
- DS1202 MicroLabBox platform
- MicroAutoBox III platform
- SCALEXIO platform
- VEOS platform
- XIL API MAPort platform

dSPACE Calibration and Bypassing Service An ECU service for measurement, calibration, bypassing, and ECU flash programming. The dSPACE Calibration and Bypassing Service can be integrated on the ECU. It provides access to the ECU application and the ECU resources and is used to control communication between an ECU and a calibration and/or bypassing tool.

With the dSPACE Calibration and Bypassing Service, users can run measurement, calibration, bypassing, and flash programming tasks on an ECU via the DCI-GSI2. The service is also designed for bypassing ECU functions using dSPACE prototyping hardware by means of the RTI Bypass Blockset in connection with DPMEM PODs. The dSPACE Calibration and Bypassing Service allows measurement, calibration, and bypassing tasks to be performed in parallel.

dSPACE Internal Bypassing Service An ECU service for on-target prototyping. The dSPACE Internal Bypassing Service can be integrated in the ECU application. It lets you add additional functions to be executed in the context of the ECU application without the need for recompiling the ECU application.

dSPACE Log A collection of errors, warnings, information, questions, and advice issued by all dSPACE products and connected systems over more than one session.

dSPACE system A hardware system such as a MicroAutoBox III or SCALEXIO system on which the real-time application runs.

Duration trigger A trigger that defines a duration. Using a duration trigger, you can, for example, specify the duration of data acquisition for a measurement raster. A duration trigger can be used as a stop trigger.

E

ECU Abbreviation of *electronic control unit*.

ECU application A sequence of operations executed by an ECU. An ECU application is mostly represented by a group of files such as ECU Image files, MAP files, A2L files and/or software module description files.

ECU calibration interface Interface for accessing an ECU by either emulating the ECU's memory or using a communication protocol (for example, XCP on CAN).

ECU diagnostics Functions such as:

- Handling the ECU fault memory: Entries in the ECU´s fault memory can be read, cleared, and saved.
- Executing diagnostic services and jobs: Users can communicate with an ECU via a diagnostic protocol using diagnostic services, diagnostic jobs, and control primitives.

ControlDesk provides the ECU Diagnostics device device to access ECUs for diagnostic tasks. Communication is via diagnostic protocols implemented on the ECUs.

ECU diagnostics with ControlDesk are completely based on Open Diagnostic Data Exchange (ODX), the ASAM MCD-2 D diagnostics standard.

ControlDesk provides the Fault Memory Instrument and the Diagnostics Instrument for ECU diagnostics tasks.

ECU Diagnostics device A device that provides access to ECUs connected to the ControlDesk PC via CAN or K-Line for diagnostics or flash programming purposes.

ControlDesk provides the *ECU Diagnostics v2.0.2* device, which supports the ASAM MCD-3 D V2.0.2 standard.

ControlDesk supports the following ODX database standards:

- ASAM MCD-2 D V2.0.1
- ASAM MCD-2 D V2.2.0 (ISO 22901-1)

ECU flash programming A method by which new code or data is stored in ECU flash memory.

ECU Image file A binary file that is part of the ECU application. It usually contains the code of an ECU application and the data of the parameters within the application. It can be stored as an Intel Hex (HEX) or Motorola S-Record (MOT or S19) file.

EESPort Configurations controlbar A controlbar for configuring error configurations.

Electrical error simulation Simulating electrical errors such as loose contacts, broken cables, and short-circuits, in the wiring of an ECU. Electrical error simulation is performed by the failure simulation hardware of an HIL simulator.

Electrical Error Simulation port (EESPort) An *Electrical Error Simulation port* (EESPort) provides access to a failure simulation hardware for simulating electrical errors in an ECU wiring according to the ASAM AE XIL API standard.

The configuration of the EESPort is described by a hardware-dependent *port configuration* and one or more *error configurations*.

Environment model A model that represents a part or all of the ECU's environment in a simulation scenario.

The environment model is a part of the simulation system.

Environment VPU The executable of an environment model built for the VEOS platform. An environment VPU is part of an offline simulation application (OSA).

Error An electrical error that is specified by:

- An error category
- An error type
- A load type

Error category The error category defines how a signal is disturbed. Which errors you can create for a signal depends on the connected failure simulation hardware.

Error configuration An XML file that describes a sequence of errors you want to switch during electrical error simulation. Each error configuration comprises error sets with one or more errors.

Error set An error set is used to group errors (pin failures).

Error type The error type specifies the way an error category – i.e., an interruption or short circuit of signals – is provided. The error type defines the disturbance itself.

Ethernet Bus Monitoring device A device that monitors the data stream on an Ethernet network connected to the ControlDesk PC.

The device supports the following variable description file type:

- AUTOSAR system description (ARXML)

Ethernet connection A mode for connecting dSPACE real-time hardware to the host PC via Ethernet. The list below shows the possible Ethernet connections:

- dSPACE real-time hardware installed in an expansion box connected to the host PC via Ethernet.
- MicroAutoBox II/III and MicroLabBox connected via Ethernet.

Ethernet decoding A feature of the Bus Navigator that lets you view protocol data and raw data of an Ethernet frame.

Event An event that is triggered by an action performed in ControlDesk.

Event context The scope of validity of event source s and event s. There is one event handler code area for each event context.

Event handler Code that is executed when the related event occurs.

Event management Functionality for executing custom code according to actions triggered by ControlDesk.

Event source An object providing and triggering event s. *LayoutManagement* is an example of an event source.

Event state State of an event. ControlDesk provides the following event states:

- No event handler is defined
- Event handler is defined and enabled
- Event handler is defined and disabled
- Event handler is defined, but no Python code is available
- Event handler is deactivated because a run-time error occurred during the execution of the Python code

Expansion box A box that hosts dSPACE boards. It can be connected to the host PC via bus connection or via network.

Experiment A container for collecting and managing information and files required for a parameter calibration and/or measurement task. A number of experiments can be collected in a project but only one of them can be active.

Extension script A Python script (PY or PYC file) that is executed each time ControlDesk starts up. An extension script can be executed for all users or user-specifically.

F

Failure insertion unit Hardware unit used with dSPACE simulators to simulate failures in the wiring of an ECU, such as broken wire and short circuit to ground.

Fault memory Part of the ECU memory that stores diagnostic trouble code (DTC) entries with status and environment information.

Fault Memory Instrument An instrument for reading, clearing, and saving the content of the ECU's fault memory.

Firmware update An update for the firmware installed in the board's flash memory. Firmware should be updated if it is older than required by the real-time application to be downloaded.

Fixed axis An axis with data points that are not deposited in the ECU memory. Unlike a common axis, a fixed axis is specified within a curve or map. The parameters of a fixed axis cannot be calibrated.

Fixed parameter A parameter that has a fixed value during a running simulation. Changing the value of a fixed parameter does not immediately affect the simulation results. The affect occurs only after you stop the simulation and

start it again. A fixed parameter is represented by an added pin in its symbol, for example: .

Flash job A specific diagnostic job for flashing the ECU memory. A flash job implements the process control for flashing the ECU memory, such as initialization, security access, writing data blocks, etc.

Flight recording The recording of data on dSPACE real-time hardware that does not require a physical connection between the host PC and the real-time hardware. In contrast to data logging, flight recording is not configured in ControlDesk but via RTI and RTLib.

Frame An instrument for adding a background frame to a layout, for example, to visualize an instrument group.

G

Gauge An instrument for displaying the value of the connected variable by a needle deflection on a circular scale.

Gigalink module A dSPACE board for connecting several processor boards in a multiprocessor system. The board allows high-speed serial data transmission via fiber-optic cable.

GNSS data Positioning and timing data that is transmitted by a Global Navigation Satellite System (GNSS), such as GPS, GLONASS, or Galileo. GNSS receivers use this data to determine their location.

GNSS device A device that provides positioning data from a GNSS receiver (e.g., a serial GPS mouse) in ControlDesk.

ControlDesk provides the *GNSS (GPS, GLONASS, Galileo, ...)* device that supports various global navigation satellite systems.

GPX file An XML file that contains geodata, such as waypoints, routes, or tracks. In ControlDesk, you can import GPX files to visualize GNSS positioning data in a Map instrument.

Group A collection of variables that are grouped according to a certain criterion.

H

Heading Indicator An instrument displaying the heading direction of a simulated aircraft on a circular scale.

Host service A service in a real-time application that provides measurement data from the application to the host PC.

The following platforms support applications that contain the host service:

- DS1006 Processor Board platform
- DS1104 R&D Controller Board platform
- MicroAutoBox platform
- Multiprocessor System platform

I

Index Plotter A plotter instrument for displaying signals that are measured in an event-based raster (index plots).

Input quantity A measurement variable that is referenced by a common axis and that provides the input value of that axis.

Instrument An on-screen representation that is designed to monitor and/or control simulator variables interactively and to display data captures. Instruments can be arranged freely on layouts.

The following instruments can be used in ControlDesk:

- 3-D Viewer
- Airspeed Indicator
- Altimeter
- Animated Needle
- Artificial Horizon
- Bar
- Browser
- Bus Instrument
- Check Button
- Diagnostics Instrument
- Display
- Fault Memory Instrument
- Frame
- Gauge
- Heading Indicator
- Index Plotter
- Invisible Switch
- Knob
- Multistate Display
- Multiswitch
- Numeric Input
- On/Off Button

- Push Button
- Radio Button
- Selection Box
- Slider
- Sound Controller
- Static Text
- Steering Controller
- Table Editor
- Time Plotter
- Variable Array
- XY Plotter

Instrument Navigator A controlbar that displays a tree with all the instrument s of the active layout and all the variables that are connected to them. The Instrument Navigator's main function is easy selection of instruments in complex layouts.

Instrument script A Python script used to extend the functionality of an instrument .

Instrument Selector A controlbar that provides access to ControlDesk's instrument s. The instruments can be placed on a layout via double-click or drag & drop.

Interface description data (IF_DATA) An information structure, mostly provided by an A2L file , describing the type, features and configuration of an implemented ECU interface.

Internal Interpreter ControlDesk's built-in programming interface for editing, running and importing Python scripts. It contains an Interpreter controlbar where the user can enter Python commands interactively and which displays output and error messages of Python commands.

Interpreter controlbar A controlbar that can be used to execute line-based commands. It is used by the Internal Interpreter to print out Python standard error messages and standard output during the execution or import of Python scripts.

Invisible Switch An instrument for defining an area that is sensitive to mouse operations.

IOCNET IOCNET (I/O carrier network) is a dSPACE-specific high-speed serial communication bus that connects all the real-time hardware in a SCALEXIO system. IOCNET can also be used to build a multiprocessor system that consists of multiple SCALEXIO processor hardware components.

K

Knob An instrument for displaying and setting the value of the connected variable by means of a knob on a circular scale.

L

Label list A list of user-defined variables that can be used for saving connected variables, etc.

Layout A window with instrument s connected to variables of one or more simulation models.

Layout Navigator A controlbar that displays all opened layout s. It can be used for switching between layouts.

Layout script A Python script used to extend the functionality of a layout .

Leading raster The measurement raster that specifies the trigger settings for the Time Plotter display. The leading raster determines the time range that is visible in the plotter if a start and stop trigger is used for displaying the signals.

LIN Bus Monitoring device A device that monitors the data stream on a LIN bus connected to the ControlDesk PC.

The LIN Bus Monitoring device works, for example, with PC-based LIN interfaces.

The device supports the following variable description file types:

- LDF
- FIBEX
- AUTOSAR system description (ARXML)

Load type The load type specifies the option to disturb a signal with or without load rejection.

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

```
%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\
<ProductName>
```

Logical link A representation of an ECU specified in the diagnostics database. A logical link contains information on the ECU itself, and all the information required for accessing it, such as the diagnostic protocol used for

communication between the ECU and ControlDesk. Each logical link is represented by a unique short name in the ODX database.

Look-up table A look-up table maps one or more input values to one output value. You have to differentiate between the following look-up table types:

- A 1-D look-up table maps one input value to one output value.
- A 2-D look-up table maps two input values to one output value.
- An n-D look-up table maps multidimensional table data with 3 or more input values to one output value.

Look-up table is a generic term for curves and maps.

M

Main variable A scalar variable that is visualized in an instrument that can be used to change parameter values. In addition to the main variable, additional write variables can also be connected to (but not visualized in) the same instrument. When you change the value of the main variable in an instrument, the changed value is also applied to all the additional write variables connected to that instrument.

Map A parameter that consists of

- A 1-dimensional array containing the axis points for the x-axis. This array can also be specified by a reference to a common axis.
- A 1-dimensional array containing the axis points for the y-axis. This array can also be specified by a reference to a common axis.
- A 2–dimensional array containing data points. The map assigns one data point of the array to each pair of x-axis and y-axis points.

Maps are represented by the symbol.

Map file A file that contains symbols (symbolic names) and their physical addresses. It is generated during a build process of an ECU application.

Map instrument A customized Browser instrument. It uses an instrument script to open a web map and connect positioning data to the map. The Map instrument offers prepared connection nodes to connect variables with GNSS data.

Measurement Viewing and analyzing the time traces of variables, for example, to observe the effects of ECU parameter changes.

ControlDesk provides various instruments for measuring variables.

Measurement (variable type) A scalar variable that can be measured, including individual elements of a measurement array.

Measurement variables are represented by the symbol.

Measurement array A 1-, 2-, or 3-dimensional array of measurement variables. In variable lists, ControlDesk displays entries for the measurement array itself and for each array element.

Measurement arrays are represented by the symbol.

Measurement buffer A ring buffer that buffers measurement data at the start of a measurement. The measurement buffer size determines the amount of data that can be buffered. Earlier values are overwritten by later values when the buffer capacity is exceeded (buffer overflow).

Measurement Configuration A controlbar that allows you to configure measurement, recording and data logging.

Measurement Data API Application programming interface for accessing measurement data. The API lets the user access measurement data without having to use ControlDesk.

Measurement Data Pool A controlbar that provides access to measurement data recorded in measurement data files.

Measurement raster Specification of how often a value of a variable is updated during a measurement. A measurement raster can be derived from a measurement service.

Measurement service The generic term for the following services:

- CCP service
- DsDAQ service
- Host service
- XCP service

Measurement signal list A list containing the variables to be included in subsequent measurements and recording. The list is global for all platforms/devices of the current experiment. The measurement signal list is available in the configuration area of the **Measurement Configuration** controlbar.

Measurement variable Any variable type that can be measured but not calibrated.

Measuring/recording A platform/device state defined by the following characteristics:

- A continuous logical connection is established between ControlDesk and the platform/device hardware.
- Online calibration is possible. Parameter values can be changed directly on the platform/device hardware.
- A measurement (or recording) is running.
- Platform/device configuration is not possible.

The 'measuring' / 'recording' platform/device state is indicated by the icon.

Memory page An area of a calibration memory. Each page contains a complete set of parameters of the platform/device hardware, but only one of the pages is “visible” to the microcontroller of the ECU or the real-time processor (RTP) of the platform hardware at a time.

ControlDesk supports platform/device hardware with up to two memory pages. These are usually the working page and the reference page. The parameter values on the two memory pages usually are different. ControlDesk lets you switch from one page to the other, so that when parameters are changed on one page, the changes can be made available to the ECU or prototyping hardware via a single page switch.

Messages controlbar A controlbar displaying a history of all error and warning messages that occur during work with ControlDesk.

MicroAutoBox III platform A platform that provides access to a MicroAutoBox III connected to the host PC for function prototyping purposes such as Bypassing.

MicroAutoBox platform A platform that provides access to a MicroAutoBox II connected to the host PC for function prototyping purposes such as bypassing.

Mirrored memory A memory area created by ControlDesk on the host PC that mirrors the contents of the available memory pages of calibration and prototyping hardware. For hardware with two memory pages, the mirrored memory is divided into a reference and a working page, each of them containing a complete set of parameters. When a calibration or prototyping platform/device is added to an experiment, ControlDesk initially fills the available memory pages of the mirrored memory with the contents of the ECU Image file (initial filling for calibration devices) or with the contents of the SDF file (initial filling for platforms).

- Mirrored memory for offline calibration

  Parameter values can even be changed offline. Changes to parameter values that are made offline affect only the mirrored memory.
- Offline-to-online transition for online calibration

  For online calibration, an offline-to-online transition must be performed. During the transition, ControlDesk compares the memory pages of the hardware of each platform/device with the corresponding pages of the mirrored memory. If the pages differ, the user has to equalize them by uploading them from the hardware to the host PC, or downloading them from the host PC to the hardware.
- Mirrored memory for online calibration

  When ControlDesk is in the online mode, parameter value changes become effective synchronously on the memory pages of the hardware and in the mirrored memory. In other words, parameter values on the hardware and on the host PC are always the same while you are performing online calibration.

Modular system A dSPACE processor board and one or more I/O boards connected to it.

Multi-capture history The storage of all the captures acquired during a triggered measurement. The amount of stored data depends on the measurement buffer.

Multi-pin error A feature of the SCALEXIO concept for electrical error simulation that lets you simulate a short circuit between three or more signal

channels and/or bus channels. The channels can be located on the same or different boards or I/O units. You can simulate a short circuit between:

- Channels of the same signal category (e.g., four signal generation channels)
- Channels of different signal categories (e.g., three signal generation channels and two signal measurement channels)
- Signal channels and bus channels (e.g., two signal generation channels, one signal measurement channel, and one bus channel)

Multiple electrical errors A feature of the SCALEXIO concept for electrical error simulation that lets you switch electrical errors at the same time or in succession. For example, you can simulate an open circuit for one channel and a short circuit for another channel at the same time, without deactivating the first error.

Multiprocessor System platform A platform that provides access to:

- A multicore application running on a multicore DS1006 board
- A multiprocessor application on a multiprocessor system consisting of two or more DS1006 processor boards interconnected via Gigalink.

ControlDesk handles a multiprocessor/multicore system as a unit and uses one system description file (SDF file) to load the applications to all the processor boards/cores in the system.

Multistate Display An instrument for displaying the value of a variable as an LED state and/or as a message text.

Multistate LED A value cell type of the Variable Array for displaying the value of a variable as an LED state.

Multiswitch An instrument for changing variable values by clicking sensitive areas in the instrument and for visualizing different states depending on the current value of the connected variable.

N

Numeric Input An instrument (or a value cell type of the Variable Array) for displaying and setting the value of the connected variable numerically.

O

Observing variables Reading variable values cyclically from the dSPACE real-time hardware and displaying their current values in ControlDesk, even if no measurement is running. Variable observation is performed without using a measurement buffer, and no value history is kept.

For platforms that support variable observation, variable observation is available for parameters and measurement variables that are visualized in single-shot instruments (all instruments except for a plotter). If you visualize a variable in a single-shot instrument, the variable is not added to the measurement signal list. Visualizing a parameter or measurement variable in a plotter automatically adds the variable to the measurement signal list.

ControlDesk starts observing variables if one of the following conditions is true:

- Online Calibration is started for the platform.

  All the parameters and measurement variables that are visualized in single-shot instruments are observed.
- Measurement is started for the platform.

  All the visualized parameters and measurement variables that are not activated for measurement in the measurement signal list are observed. Data of the activated parameters and measurement variables is acquired using measurement rasters.

ODX database Abbreviation of Open Diagnostic Data Exchange, a diagnostics database that is the central ECU description for working with an ECU Diagnostics device in ControlDesk. The ODX database contains all the information required to perform diagnostic communication between ControlDesk and a specific ECU or set of ECUs in a vehicle network. ControlDesk expects the database to be compliant with ASAM MCD-2 D (ODX).

Offline State in which the parameter values of platform/device hardware in the current experiment cannot be changed. This applies regardless of whether or not the host PC is physically connected to the hardware.

The mirrored memory allows parameter values to be changed even offline.

Offline simulation A PC-based simulation in which the simulator is not connected to a physical system and is thus independent of the real time.

Offline simulation application (OSA) An offline simulation application (OSA) file is an executable file for VEOS. After the build process with a tool such as the VEOS Player, the OSA file can be downloaded to VEOS.

An OSA contains one or more VPUs, such as V-ECUs and/or environment VPUs.

On/Off Button An instrument (or a value cell type of the Variable Array) for setting the value of the connected parameter to a predefined value when the button is pressed (On value) and released (Off value).

Online calibration started A platform/device state defined by the following characteristics:

- A continuous logical connection is established between ControlDesk and the platform/device hardware.
- Online calibration is possible. Parameter values can be changed directly on the platform/device hardware.
- Platform/device configuration is not possible.

Before starting online calibration, ControlDesk lets you compare the memory pages on the platform/device hardware with the corresponding pages of the mirrored memory. If the parameter values on the pages differ, they must be

equalized by uploading the values from the hardware to ControlDesk, or downloading the values from ControlDesk to the hardware. However, a page cannot be downloaded if it is read-only.

The 'online calibration started' platform/device state is indicated by the symbol.

Operation signal A signal which represents the result of an arithmetical operation (such as addition or multiplication) between two other signals.

Operator mode A working mode of ControlDesk in which only a subset of the ControlDesk functionality is provided. You can work with existing experiments but not modify them, which protects them from unintentional changes.

Output parameter A parameter or writable measurement whose memory address is used to write the computed value of a calculated variable to.

P

Parameter Any variable type that can be calibrated.

Parameter (variable type) A scalar parameter, as well as the individual elements of a value block.

Scalar parameters are represented by the P symbol.

Parameter limits Limits within which parameters can be changed. Parameters have hard and weak limits.

- Hard limits

  Hard limits designate the value range of a parameter that you *cannot* cross during calibration.

  The hard limits of a parameter originate from the corresponding variable description and cannot be edited in ControlDesk.
- Weak limits

  Weak limits designate the value range of a parameter that you *should not* cross during calibration. When you cross the value range defined by the weak limits, ControlDesk warns you.

  In ControlDesk, you can edit the weak limits of a parameter within the value range given by the parameter's hard limits.

PHS (Peripheral High Speed) bus A dSPACE-specific bus for communication between a processor board and the I/O boards in a modular system. It allows direct I/O operations between the processor board (bus master) and I/O boards (bus slaves).

PHS-bus-based system A modular dSPACE system consisting of a processor board such as the DS1006 Processor Board and I/O boards. They communicate with each other via the PHS (Peripheral High Speed) bus.

Pitch variable A variable connected to the pitch scale of an Artificial Horizon.

Platform A software component representing a simulator where a simulation application is computed in real-time (on dSPACE real-time hardware) or in non-real-time (on VEOS).

ControlDesk provides the following platforms:

- DS1006 Processor Board platform
- DS1007 PPC Processor Board platform
- DS1104 R&D Controller Board platform
- DS1202 MicroLabBox platform
- MicroAutoBox platform
- MicroAutoBox III platform
- Multiprocessor System platform
- SCALEXIO platform
- VEOS platform
- XIL API MAPort platform

Each platform usually has a variable description that specifies its variables.

Platform trigger A trigger that is available for a platform and that is evaluated on the related dSPACE real-time hardware or VEOS.

Platforms/Devices controlbar A controlbar that provides functions to handle devices, platforms, and the applications assigned to the platforms.

Plotter instrument ControlDesk offers three plotter instruments with different main purposes:

- The Index Plotter displays signals in relation to events.
- The Time Plotter displays signals in relation to measurement time.
- The XY Plotter displays signals in relation to other signals.

Port configuration To interface the failure simulation hardware, an EESPort needs the hardware-dependent *port configuration file* (PORTCONFIG file). The file's contents must fit the connected HIL simulator architecture and its failure simulation hardware.

Postprocessing The handling of measured and recorded data by the following actions:

- Displaying measured or recorded data
- Zooming into measured or recorded signals with a plotter
- Displaying the values of measurement variables and parameters as they were at any specific point in time

Processor board A board that computes real-time applications. It has an operating system that controls all calculations and communication to other boards.

Project A container for collecting and managing the information and files required for experiment/calibration/modification tasks in a number of experiments. A project collects the experiments and manages their common data.

Project controlbar A controlbar that provides access to projects and experiments and all the files they contain.

Project root directory The directory on your file system to which ControlDesk saves all the experiments and documents of a project. Every project is associated with a project root directory, and several projects can use the same project root directory. The user can group projects by specifying several project root directories.

ControlDesk uses the Documents folder as the default project root directory unless a different one is specified.

Properties controlbar A controlbar providing access to the properties of, for example, platforms/devices, layouts/instruments, and measurement/recording configurations.

Proposed calibration A calibration mode in which the parameter value changes that the user makes do not become effective on the hardware until they are applied. This allows several parameter changes to be written to the hardware together. Being in proposed calibration mode is like being in the offline calibration mode temporarily.

Push Button An instrument (or a value cell type of the Variable Array) for setting the value of the connected parameter by push buttons.

Python Editor An editor for opening and editing PY files.

Q

Quick start measurement A type of measurement in which all the ECU variables configured for measurement are measured and recorded, starting with the first execution of an ECU task. ControlDesk supports quick start measurements on ECUs with DCI-GSI2, CCP, and XCP (except for XCP on Ethernet with the TCP transmission protocol).

Quick start measurement can be used to perform cold start measurements. Cold start means that the vehicle and/or the engine are cooled down to the temperature of the environment and then started. One reason for performing cold start measurements is to observe the behavior of an engine during the warm-up phase.

R

Radio Button An instrument for displaying and setting the value of the connected parameter by radio buttons.

Real-time application An application that can be executed in real time on dSPACE real-time hardware. A real-time application can be built from a Simulink model containing RTI blocks, for example.

Record layout A record layout is used to specify a data type and define the order of the data in the memory of the target system (ECU, for example). For scalar data types, a record layout allows you to add an address mode (direct or indirect). For structured (aggregated) data types, the record layout specifies all the structure elements and the order they appear in.

The `RECORD_LAYOUT` keyword in an A2L file is used to specify the various record layouts of the data types in the memory. The structural setup of the various data types must be described in such a way that a standard application system will be able to process all data types (reading, writing, operating point display etc.).

Record layout component A component of a record layout. A structured record layout consists of several components according to the ASAP2 specification. For example, the AXIS_PTS_X component specifies the x-axis points, and the FNC_VALUES component describes the function values of a map or a curve.

Recorder An object in the Measurement Configuration controlbar that specifies and executes the recording of variables according to a specific measurement configuration.

Recorder signal list A list that contains the variables to be included in subsequent recordings.

Recording Saving the time traces of variables to a file. Both measurement variables and parameters can be recorded. Recorded data can be postprocessed directly in ControlDesk.

A recording can be started and stopped immediately or via a trigger:

- Immediate recording

  The recording is started and stopped without delay, without having to meet a trigger condition.
- Triggered recording

  The recording is not started or stopped until certain trigger conditions are met. These conditions can be defined and edited in ControlDesk.

Reduction data Additional content in an MF4 file that allows for visualizing the MF4 file data depending on the visualization resolution. Reduction data therefore improves the performance of the visualization and postprocessing of measurement data.

Reference data set A read-only data set assigned to the reference page of a device that has two memory pages. There can be only one reference data set for each device. The reference data set is read-only.

Reference page Memory area containing the parameters of an ECU. The reference page contains the read-only reference data set.

> **Note**
>
> Some platforms/devices provide only a working page. You cannot switch to a reference page in this case.

Resynchronization Mechanism to periodically synchronize the drifting timers of the platform/device hardware ControlDesk is connected to. Resynchronization means adjustment to a common time base.

Roll variable A variable connected to the roll scale of an Artificial Horizon.

S

Sample count trigger A trigger that specifies the number of samples in a data capture.

A sample count trigger can be used as a stop trigger.

SCALEXIO platform A platform that provides access to a single-core, multicore or multiprocessor SCALEXIO system connected to the host PC for HIL simulation and function prototyping purposes.

SCALEXIO system A dSPACE hardware-in-the-loop (HIL) system consisting of at least one processing hardware component, I/O boards, and I/O units. They communicate with each other via the IOCNET. In a SCALEXIO system, two types of processing hardware can be used, a DS6001 Processor Board or a real-time industry PC as the SCALEXIO Processing Unit. The SCALEXIO system simulates the environment to test an ECU. It provides the sensor signals for the ECU, measures the signals of the ECU, and provides the power (battery voltage) for the ECU and a bus interface for restbus simulation.

SDF file The system description file that describes the files to be loaded to the individual processing units of a simulation platform. It also contains the variable description of the relevant simulation application.

The SDF file is generated automatically when the TRC file is built.

Segment The minimum part a segment signal can consist of.

There are different kinds of segments to be used in segment signals:

- Segments to form synthetic signal shapes (sine, sawtooth, ramp, etc.)
- Segments to perform arithmetical operations (addition, multiplication) with other segments
- Segments to represent numerical signal data (measured data)

Segment signal A signal consisting of one or more segments.

Selection Box An instrument for selecting a text-value entry and setting the respective numerical value for the connected variable.

Signal

- Representation of a variable measured in a specific measurement raster.
- Generic term for segment signals and operation signals.

  A signal is part of a signal description set which can be displayed and edited in the working area.

Signal description set A group of one or more signals.

A signal description set and its signals can be edited in the working area by means of the Signal Editor. Each signal description set is stored as an STZ file either in the **Signal Description Sets** folder or in the **Signal Generators** folder.

Signal Editor A software component to create, configure, display, and manage signals in signal description sets.

Signal file A file that contains the wiring information of a simulator and that is part of the standard dSPACE documentation of dSPACE Simulator Full-Size. Normally, dSPACE generates this file when designing the simulator. Before using a failure simulation system, users can adapt the signal file to their needs.

Signal generator An STZ file containing a signal description set and optional information about the signal mapping, the description of variables, and the real-time platform.

The file is located in the **Signal Generators** folder and used to generate, download, and control Real-Time Testing sequences, which are executed on the real-time platform to stimulate model variables in real time.

Signal Mapping A controlbar of the Signal Editor to map model variables to signals and variable aliases of a signal generator.

Signal Selector A controlbar of the Signal Editor. The Signal Selector provides signals and segments for arranging and configuring signal description sets in the working area.

SIL testing Abbreviation of *software-in-the-loop testing*.

Simulation and testing of individual software functions, complete virtual ECUs (V-ECUs), or even V-ECU networks on a local PC or highly parallel in the cloud independently of real-time constraints and real hardware.

Simulation application The generic term for offline simulation application (OSA) and real-time application.

Simulation system A description of the composition of V-ECU models, environment models, real ECUs, and their interconnections required for simulating the behavior of a system. A simulation system is the basis for the generation of a simulation application for a given simulator platform.

Simulation time group Group of platforms/devices in an experiment whose simulation times are synchronized with each other. If resynchronization is enabled, ControlDesk synchronizes a simulation time group as a whole, not the single members of the group individually.

Simulator A system that imitates the characteristics or behaviors of a selected physical or abstract system.

Single-processor system A system that is based on one dSPACE processor or controller board.

Single-shot instrument An instrument that displays an instantaneous value of a connected variable without keeping a value history. In ControlDesk, all instruments except for a plotter are single-shot instruments. For platforms that support the variable observer functionality, you can use single-shot instruments to observe variables.

Slave application An application assigned to the slave DSP of a controller or I/O board. It is usually loaded and started together with the real-time application running on the corresponding main board.

Slave DSP A DSP subsystem installed on a controller or I/O board. Its slave application can be loaded together with the real-time application or separately.

Slider An instrument (or a value cell type of the Variable Array) for displaying and setting the value of the connected variable by means of a slide.

Sound Controller An instrument for generating sounds to be played.

Standard axis An axis with data points that are deposited in the ECU memory. Unlike a common axis, a standard axis is specified within a curve or map. The parameters of a standard axis can be calibrated, which affects only the related curve or map.

Start trigger A trigger that is used, for example, to start a measurement raster. A platform trigger can be used as a start trigger.

Static Text An instrument for displaying explanations or inscriptions on the layout.

Steering Controller An instrument for changing variable values using a game controller device such as a joystick or a steering wheel.

Stimulation Writing signals to variables in real-time models during a simulation run.

Stop trigger A trigger that is used, for example, to stop a measurement raster.

String A text variable in ASCII format.

Strings are represented by the symbol.

Struct A variable with the struct data type. A struct contains a structured list of variables that can have various data types. In ControlDesk, a struct variable can contain either parameters and value blocks or measurement variables and measurement arrays. ControlDesk supports nested structs, i.e., structs that contain further structs and struct arrays as elements.

Structs are represented by the symbol.

Struct array An array of homogeneous struct variables.

Struct arrays are represented by the symbol.

STZ file A ZIP file containing signal descriptions in the STI format. The STZ file can also contain additional MAT files to describe numerical signal data.

Sub data set A data set that does not contain the complete set of the parameters of a platform/device.

Symbol A symbolic name of a physical address in a MAP file. A symbol can be associated to a variable in the Variable Editor, for example, to support an address updates.

System variable A type of variable that represents internal variables of the device or platform hardware and that can be used as measurement signals in ControlDesk to give feedback on the status of the related device or platform hardware. For example, an ECU's power supply status or the simulation state of a dSPACE board can be visualized via system variables.

T

Table Editor An instrument for displaying and setting values of a connected curve, map, value block, or axis in a 2-D, 3-D, and grid view. The Table Editor can also display the values of a measurement array.

The Table Editor can be used for the following variable types:

- Common axis ( )
- Curve ( )
- Map ( )
- Measurement array ( )
- Value block ( )

Time cursor A cursor which is visible at the same time position in the following instruments:

- In all Time Plotters
- In all XY Plotters
- In all bus monitoring lists

You can use the time cursor to view signal values at a specific point in time. If you move the time cursor, all measured signals and the respective parameters are

updated. Instruments and bus monitoring lists display the values that are available at the selected time position.

Time Plotter A plotter instrument for displaying signals that are measured in a time-based raster (time plots).

Topology A description of the processor boards belonging to a multiprocessor system and their interconnections via Gigalinks. The topology also contains information on which Gigalink port of each processor board is connected to the Gigalink ports of other processor boards in the multiprocessor system.

Topology information is contained in the real-time application (PPC/x86/RTA) files of the multiprocessor system's processor boards.

TRC file A variable description file with information on the variables available in an environment model running on a dSPACE platform.

Trigger A condition for executing an action such as starting and stopping a measurement raster or a recorder.

The generic term for the following trigger types:

- Duration trigger
- Platform trigger
- Sample count trigger

Trigger condition A formula that specifies the condition of a trigger mathematically.

Triggered measurement The measurement of a measurement raster started by a platform trigger. The data flow between the dSPACE real-time hardware or VEOS and the host PC is not continuous.

U

Unassigned data set A data set that is assigned neither to the working page nor to the reference page of a platform/device. An unassigned data set can be defined as the new working or reference data set. It then replaces the "old" working or reference data set and is written to the corresponding memory page, if one is available on the platform/device.

Unplugged A platform/device state defined by the following characteristics:

- The logical connection between ControlDesk and the hardware was interrupted, for example, because the ignition was turned off or the ControlDesk PC and the hardware were disconnected.
- Before the state of a platform/device changes to 'unplugged', the platform/device was in one of the following states:
  - 'Connected'
  - 'Online calibration started'
  - 'Measuring' / 'Recording'

Tip

A device for which the connection between ControlDesk and the device hardware currently is interrupted is also set to the 'unplugged' state when you start online calibration if both the following conditions are fulfilled:

- The device's **Start unplugged** property is enabled.
- The **Start online calibration behavior** property is set to 'Ignore differences'.

This is possible for CCP and XCP devices. For details on the two properties listed above, refer to General Settings Properties (ControlDesk Platform Management 📖).

- If the Automatic Reconnect feature is enabled for a platform/device and if the platform/device is in the 'unplugged' state, ControlDesk periodically tries to re-establish the logical connection for that platform/device.
- Online calibration is impossible. Offline calibration is possible.
- Platform/device configuration is possible.

The 'unplugged' platform/device state is indicated by the ⚠ icon.

Untriggered measurement The measurement of a measurement raster not started by a platform trigger. The data flow between the dSPACE real-time hardware or VEOS and the host PC is continuous.

User function An external function or program that is added to the ControlDesk user interface for quick and easy access during work with ControlDesk.

User Functions Output A controlbar that provides access to the output of external tools added to the **Automation** ribbon.

V

Value block A parameter that consists of a 1- or 2-dimensional array of scalar parameters.

In variable lists, ControlDesk displays entries for the value block itself and for each array element.

Value blocks are represented by the symbol.

Value conversion The conversion of the original *source values* of variables of an application running on an ECU or dSPACE real-time hardware into the corresponding scaled *converted values*.

Variable Any parameter or measurement variable defined in a variable description. ControlDesk provides various instrument s to visualize variables.

Variable alias An alias name that lets the user control the property of a segment by a model parameter of a real-time application.

Variable Array An instrument for calibrating parameters and displaying measurement variable values.

The Variable Array can be used for the following variable types:

- Measurement ( )
- Measurement array ( )
- String ( )
- Struct ( )
- Struct array ( )
- Value ( )
- Value block ( )

Variable connection The connection of a variable to an instrument. Via the variable connection, data is exchanged between a variable and the instrument used to measure or calibrate the variable. In other words, variable connections are required to visualize variables in instrument.

Variable description A file describing the variables in a simulation application, which are available for measurement, calibration, and stimulation.

Variable Editor A tool for viewing, editing, and creating variable descriptions in the ASAM MCD-2MC (A2L) file format. The Variable Editor allows you to create A2L files from scratch, or to import existing A2L files for modification.

Variable Filter A variable filter contains the filter configuration of a combined filter, which is used to filter the variable list in the **Variables** controlbar using a combination of filter conditions.

Variables controlbar A controlbar that provides access to the variables of the currently open experiment.

V-ECU Abbreviation of *virtual ECU*.

ECU software that can be executed in a software-in-the-loop (SIL) testing environment such as a local PC or highly parallel in the cloud independently of real-time constraints and real ECU hardware.

Vehicle information The ODX database can contain information for one or more vehicles. Vehicle information data is used for vehicle identification purposes and for access to vehicles. It references the access paths (logical links) to the ECUs.

VEOS A simulator which is part of the PC and allows the user to run an offline simulation application (OSA) without relation to real time.

VEOS Player is the graphical user interface for VEOS.

VEOS platform A platform that configures and controls the offline simulation application (OSA) running in VEOS and that also provides access to the application's environment VPU.

VEOS Player An application running on the host PC for editing, configuring and controlling an offline simulation application (OSA) running in VEOS.

Verbal conversion A conversion in which a conversion table is used to specify the computation of numerical values into strings. The verbal conversion table is used when you switch the value representation from source to converted mode and vice versa.

Verbal conversion range A conversion in which a conversion table is used to specify the computation of a range of numerical values into strings. The verbal conversion range table is used when you switch the value representation from source to converted mode and vice versa.

View set A named configuration of the controlbars of ControlDesk. A view set has a default state and a current state that can differ from the default state. The configuration includes the geometry, visibility, and docking or floating state of controlbars.

Visualization The representation of variables in instruments:

- Measurement variables are visualized in instruments to view and analyze their time traces.
- Calibration parameters are visualized in instruments to change their values.

VPU Abbreviation of *virtual processing unit*. A VPU is part of an offline simulation application in VEOS. Each VPU runs in a separate process of the PC.

VPU is also the generic term for:

- V-ECUs
- Environment VPUs
- Controller VPUs
- Bus VPUs

W

Working data set The data set currently residing in the memory of a platform/device hardware. There can be only one working data set for each calibration platform/device. The working data set is read/write.

Working page Memory area containing the parameters of an ECU or prototyping hardware (memory page). The working page contains the read/write working data set.

If the platform/device also provides a reference page, ControlDesk lets you switch between both pages.

Writable measurement A scalar variable that can be measured and calibrated.

X

XCP Abbreviation of *Universal Measurement and Calibration Protocol*. A protocol that is implemented on electronic control units (ECUs) and provides access to ECUs with measurement and calibration systems (MCS) such as ControlDesk.

XCP is based on the *master-slave principle*:

- The ECU is the slave.
- The measurement and calibration system is the master.

The "X" stands for the physical layers for communication between the ECU and the MCS, such as CAN (Controller Area Network) and Ethernet.

The basic features of XCP are:

- ECU parameter calibration (CAL)
- Synchronous data acquisition (DAQ)
- Synchronous data stimulation (STIM), i.e., for bypassing
- ECU flash programming (PGM)

The XCP protocol was developed by ASAM e.V. (Association for Standardisation of Automation and Measuring Systems e.V.). For the protocol specification, refer to http://www.asam.net.

The following ControlDesk devices support ECUs with an integrated XCP service:

- XCP on CAN device
- XCP on Ethernet device

XCP on CAN device A device that provides access to an ECU with XCP connected to the ControlDesk PC via CAN. Using the XCP on CAN device, you can access the ECU for measurement and calibration purposes via XCP (*Universal Measurement and Calibration Protocol*).

XCP on Ethernet device A device that provides access to an ECU or V-ECU with XCP connected to the ControlDesk PC via Ethernet. The XCP on Ethernet device provides access to the ECU/V-ECU via XCP (*Universal Measurement and Calibration Protocol*) for measurement and calibration purposes.

XIL API EESPort Electrical Error Simulation port (EESPort)

XIL API MAPort platform A platform that provides access to a simulation platform via the ASAM XIL API implementation that is installed on your host PC.

XY Plotter A plotter instrument for displaying signals as functions of other signals.

A

Activate command
  Database 185
  Logger 186
  Monitor 187
  Replay 188
Add CAN Filter command 190
Add CAN Logger command 191
Add CAN Monitor command 193
Add Ethernet Filter command 190
Add Ethernet Logger command 191
Add Ethernet Monitor command 193
Add Filter command 190
Add LIN Filter command 190
Add LIN Logger command 191
Add LIN Monitor command 193
Add Logger command 191, 199
Add Monitor command 193
Add Replay command 189
adding a monitor node 82

B

basics
  working with the Bus Navigator 34
bus communication
  logging 76
  monitoring 76
bus configuration 14
  displaying 34
Bus Instrument
  configuring 55
Bus Instrument (Gateway type)
  creating 67
Bus Instruments
  working with 48
Bus Navigator
  Configure Column Sets dialog 214
  Customize Columns dialog 216
  displaying 34
  Import Mappings dialog 237
  introduction 14
  overview 14
  working with 34
Bus Navigator tree
  structure 19

C

Common Program Data folder 10, 275
communication matrix
  activating 44
Configure Column Sets dialog 214
configuring a Bus Instrument 55
configuring a TX message
  bus monitoring device 74
  model basis blockset 64
  model basis Bus Manager 59
configuring layout names 48
create instruments to handle RX and TX messages/frames/PDUs 50
creating a Bus Instrument (Gateway type) 67
Customize Columns dialog 216

D

Delete command
  Filter 219
  Logger 220
  Monitor 221
  Replay 222
displaying the Bus Navigator 34
Documents folder 10, 279

E

Edit command
  Filter 223
  Logger 224
  Monitor 225
  Replay 226
Edit Logger command 199
example
  specifying filters for bus monitoring and logging 88

F

filter for bus logging
  specifying 85
filter for bus monitoring
  specifying 85
filters for bus monitoring and logging
  specifying 88

G

Generate Error Status Layout command 228
Generate Gateway Layout command 229
Generate Inspection Layout command 230
Generate Layout command 231
Generate Manipulation Layout command 231
Generate RX Layout command 232
Generate TX Layout command 234
Generate TX Status Layout command 235
global time synchronization 45
global transmission settings for a controller
  specifying 69

I

Import Mappings dialog 237
Instrument Selector 286
instruments
  configuring a Bus Instrument 55
  creating instruments to handle RX and TX messages/frames/PDUs 50
introduction
  Bus Navigator 14

L

layout name
  configuring 48
Local Program Data folder 10, 287
logging bus communication 76
logging raw bus communication data 95

M

Measurement Data Pool 289
Messages controlbar 290
monitor node
  adding 82
monitoring bus communication 76
monitoring settings
  specifying 82

O

overview
  Bus Navigator 14

P

Platforms/Devices controlbar 294
Project controlbar 295
Project Manager 295
Properties controlbar 295

R

raw bus communication data
  logging 95
  replaying 101

S

Save command 249
specifying a filter for bus logging 85
specifying a filter for bus monitoring 85
specifying filters for bus monitoring and logging 88
specifying global transmission settings for a controller 69
specifying monitoring settings 82
Start command
  Logger 250
  Monitor 251
  Replay 252
structure
  Bus Navigator tree 19

T

transmitting a TX message on the bus
  bus monitoring device 74
  model basis blockset 64
  model basis Bus Manager 59
TX message
  configuring
    bus monitoring device 74
    model basis blockset 64
    model basis Bus Manager 59
  transmitting on the bus
    bus monitoring device 74
    model basis blockset 64
    model basis Bus Manager 59

W

working with Bus Instruments 48
working with the Bus Navigator 34