ConfigurationDesk

Conflicts

For ConfigurationDesk 6.7

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

Contents

About This Document

Content

This reference decribes the conflicts that can occur while you are implementing a ConfigurationDesk application. The conflicts are displayed in the Conflicts Viewer.

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.

Conflicts

Objective

ConfigurationDesk provides the **Conflict Viewer**, which displays any conflicts that occurred when the signal chain was created and configured. It also helps you to resolve these conflicts.

Where to go from here

Information in this section

Information in other sections

Resolving Conflicts (ConfigurationDesk Real-Time Implementation Guide 🕮)

Application Process Conflicts

Application process: Assigned model contains incompatible code

An application process contains an SIC file that was generated for the `dsrt64.tlc` Run-Time Target.

SIC files generated for the `dsrt64.tlc` Run-Time Target are not supported in real-time applications built by ConfigurationDesk. You can use these SIC files only to generate BSC files that are used for execution on VEOS on a Linux operating system.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	–	–	–

Remedy Remove the SIC file generated for the `dsrt64.tlc` Run-Time Target from the application process.

Application process: Default task provided and model assigned

The Provide default task checkbox in the Properties Browser of the application process is selected, and one or more model implementations are assigned to the application process.

Application processes whose Provide default task checkbox is activated must not have any model implementations assigned.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, perform one of the following actions:

- Select False from the Provide default task list of the application process.
- Clear the (Select All) checkbox from the Assigned models list of the application process, and click OK.

Application process: Duplicate DAQ raster name for tasks within common application process

An application process contains two or more tasks with the same DAQ raster name.

DAQ raster names for the tasks in an application process must be unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, enter unique names in the DAQ raster name edit field for all the tasks assigned to the application process.

Application process: Duplicate event names

An application process contains two or more timer events with the same name.

The names of timer events in an application process must be unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, enter unique names in the Name edit field for all the timer events in the application process.

Application process: Duplicate task names

An application process contains two or more tasks with the same name.

The names of all tasks in an application process must be unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, enter unique names in the Name edit field for all tasks in the application process.

Application process: Invalid number of Bus monitoring tasks

The application process contains either more than one task or no task for which bus monitoring is activated.

If you want to use bus monitoring, it must be activated for exactly one task within an application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	–	–	–

Remedy Make sure that bus monitoring is activated for exactly one task in the application process.

Application process: Mismatching active application process OS and component OS

The application process contains AUTOSAR OS tasks and user-defined tasks.

You must not mix AUTOSAR OS tasks and user-defined tasks in an application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Task Configuration table, delete the user-defined tasks affected by this conflict.

Application process: Mismatching settings for Jitter and latency optimization and Real-Time Testing

You have enabled Real-Time Testing and use Low jitter, low latency for a task in the application process.

Real-Time Testing is not supported for tasks with Jitter and latency optimization set to Low jitter, low latency.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Make sure that Real-Time Testing and Low jitter, low latency is not used for the same task in the application process. If the application process contains only tasks that are configured as Low jitter, low latency, you must create an additional low-priority standard task for that you enable Real-Time Testing.

Application process: Multiple bus configurations with the same name assigned

Two or more bus configurations within the application process have identical names.

The names of bus configurations within an application process must be unique, regardless of whether the bus configurations are contained in BSC files or are available as Bus Configuration function blocks.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Choose one of the following alternatives:

- Assign the conflicting BSC files or Bus Configuration function blocks to different application processes.
- Rename the conflicting Bus Configuration function blocks.

Application process: Multiple FlexRay configuration IDs

There are duplicate FlexRay configuration IDs in an application process.

A configuration ID is used to identify blocks from different FlexRay configurations. It must therefore be unique for each FlexRay bus in an application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Assign unique configuration IDs to each FlexRay bus within the application process.

Application process: Multiple models assigned

More than one V-ECU implementation is assigned to one application process.

Only one V-ECU implementation is supported for each application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Processing Resource Assignment table, delete all but one V-ECU implementation from the application process by using the Delete Assignment command from the context menu of the model implementations.

Application process: Multiple models providing the same global C variables assigned

Two or more model implementations that have identical global C variables are assigned to the same application process. This conflict is displayed in particular if you specify the names of C variables that are created for the parameters or signals in the relevant Simulink models by setting the storage class of these variables to ExportedGlobal, for example.

All the global C variables provided by model implementations that are assigned to the same application process must be unique. This applies to the variables that can be accessed at run time, i.e., the variables in the TRC file. If the conflict is not resolved, the related variables cannot be accessed via experiment software at run-time.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy To resolve this conflict, perform one of the following actions in the relevant model implementations:

- Specify a storage class other than ExportedGlobal for the C variables for parameters and signals.
- Specify unique names for the C variables for parameters and signals.

If the relevant model implementations are Simulink models, start the build process in ConfigurationDesk. If the relevant model implementations are SIC files, create a new SIC file and reload it in ConfigurationDesk.

Application process: Multiple models with the same name assigned to multimodel application process

A V-ECU implementation is assigned to a multimodel application process.

Each V-ECU implementation must be assigned to a separate application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Task Configuration table, remove the relevant V-ECU implementations from the multimodel application process and assign each V-ECU implementation to a separate application process.

Application process: Multiple MotionDesk models assigned to multimodel application process

The application process has two or more Simulink behavior models or SIC files assigned that contain blocks from the MotionDesk Blockset with activated platform support.

Only one Simulink model or SIC file that contains blocks from the MotionDesk Blockset with activated SCALEXIO platform support is allowed per application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Processing Resource Assignment table, delete the assignment of all but one model implementations with blocks from the MotionDesk Blockset from the multimodel application process using the Delete Assignment command from the context menu of the relevant model implementations. Assign the deleted model implementations to separate application processes.

Application process: Multiple tasks with enabled Real-Time Testing

You have enabled Real-Time Testing for more than one task within an application process.

Real-Time Testing can only be allowed for exactly one task.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, select False from the Real-Time Testing list for all but one task in the application process.

Application process: No model assigned

There is no model implementation assigned to the application process and the application process has no default task. If you want to model an executable application from scratch, you have to create an application process and assign a model implementation to it. You can then model the tasks in the application process manually using the runnable function that the model implementation provides.

Another possibility is to use no model implementation in the application process and let the application process provide a default task.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Task Configuration table, right-click the application process and assign a model implementation via the Assign Model command.

If you want the application process to run without any model implementation, activate the Provide default task property of the application process.

Application process: No task defined

There is no task defined in the application process.

An application process must have at least one task.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Task Configuration table, add at least one task to the application process. For details, refer to How to Model an Executable Application Manually from Scratch (ConfigurationDesk Real-Time Implementation Guide 🕮).

Application process: No task with DAQ raster defined

There is no task with a specified DAQ raster in the application process.

At least one task must have a specified DAQ raster.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, enter a unique name in the DAQ raster name edit field for at least one task of the application process.

Application process: No task with enabled Real-Time Testing

There is no task with Real-Time Testing enabled in the application process.

You can enable Real-Time Testing for exactly one task.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, select Real-Time Testing for exactly one task in the application process.

Application process: Too many tasks with DAQ raster name set (max. is 31)

There are too many tasks with a DAQ raster name set in the application process.

The maximum number of tasks with a DAQ raster name set is 31.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, clear the DAQ raster name edit field for excess tasks in the application process.

Application process: Unresolved model or I/O function

A model implementation assigned to the application process is unresolved. For details on unresolved model implementations, refer to Basics on Modeling Executable Applications (ConfigurationDesk Real-Time Implementation Guide 🕮). For details on unresolved I/O function blocks, refer to Basics on Function Blocks (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Add the relevant model implementation to the executable application using the Add Model context menu command in the Model Browser.

Application process: Unsupported BSC assigned to multimodel application process

A BSC file created with the Bus Manager of dSPACE Release 2019-A or earlier is assigned to the multimodel application process.

BSC files assigned to multimodel application processes must be created with the Bus Manager of dSPACE Release 2019-B.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Processing Resource Assignment table, delete the assignment of the relevant BSC file via the Delete Assignment command of the BSC file's context menu. Assign the BSC file to a separate application process.

Application process: Unsupported Jitter and latency optimization settings

There are different possible causes for this conflict, depending on the Jitter and latency optimization setting.

For a task configured as Low jitter, low latency in the application process, at least one of the following conditions is met:

- There is a task with a different Jitter and latency optimization setting and a higher priority in the application process.
- The task is triggered by a runnable function via a software event. The triggering runnable function is assigned to a task with a different Jitter and latency optimization setting.
- The task has a runnable function assigned that triggers another task via a software event. The triggered task has a different Jitter and latency optimization setting.

Or

For a task configured as No jitter, low latency in the application process, at least one of the following conditions is met:

- The task is not the only task in the application process.
- The task is not triggered by a timer event.

The settings described above are not supported.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy For a task configured as Low jitter, low latency make sure that:

- There is no task with a different Jitter and latency optimization setting and a higher priority in the application process.

- If the task is triggered by a runnable function via a software event: The Jitter and latency optimization setting of the task with the triggering runnable function is also configured as Low jitter, low latency.
- If the task has a runnable function assigned that triggers another task via a software event: The Jitter and latency optimization setting of the triggered task is also configured as Low jitter, low latency.

For a task configured as No jitter, low latency make sure that:

- The task is the only task in the application process.
- The task is triggered by a timer event.

Application process: Unsupported model assigned to multimodel application process

At least one Simulink behavior model or SIC file with blocks from unsupported blocksets is assigned to a multimodel application process.

Simulink models or SIC files that are assigned to multimodel application processes must not contain blocks from the RTI CAN MultiMessage Blockset, the RTI LIN MultiMessage Blockset, the FlexRay Configuration Package, or from the Ethernet Configuration Package.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Processing Resource Assignment table, delete the assignment of the relevant Simulink models or SIC files via the Delete Assignment command of the model implementations' context menu. Assign the Simulink models or SIC files to separate application processes.

Application process: Unsupported model assigned to multi-model application process. At least one of the models provides an ASAP2 interface

At least one behavior model that provides an ASAP2 interface is assigned to a multimodel application process.

Multimodel application processes must not include assigned behavior models that provide an ASAP2 interface.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Processing Resource Assignment table, delete the assignment of the relevant behavior models via the Delete Assignment command from the context menu of the model implementation. Assign the behavior models to separate application processes.

Application process: Unsupported precompiled FMU assigned to multimodel application process

At least one unsupported precompiled FMU is assigned to an application process together with other model implementations.

A precompiled FMU must be created with dSPACE Release 2017-A (ConfigurationDesk 5.7) or later to be used with other model implementations in the same application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Processing Resource Assignment table, delete the unsupported precompiled FMU by using the Delete Assignment command from the context menu of the FMU. To use the FMU in your real-time application, choose one of the following options:

- Assign the corresponding precompiled FMU to an application process that does not contain any model implementations.
- Create a new application process and assign the precompiled FMU to this application process.
- Precompile the FMU again, using ConfigurationDesk 6.0 (on dSPACE Release 2017-B) or later.

Application process: Unsupported SIC assigned to multimodel application process

An SIC file that was created with the Model Interface Package for Simulink 3.5 or earlier was assigned to a multimodel application process.

SIC files assigned to multimodel application processes must be created with the Model Interface Package for Simulink 3.6 or later.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Processing Resource Assignment table, delete the unsupported SIC file by using the Delete Assignment command from the context menu of the relevant application process. To use the SIC file in your real-time application, choose one of the following options:

- Assign the corresponding SIC file to an application process that does not contain any model implementations.
- Create a new application process and assign the SIC file to this application process.
- Build the SIC file again, using the Model Interface Package for Simulink 3.6 (on dSPACE Release 2018-A) or later.

Application process: Unsupported task priority value

The assigned application process has a task with a priority that is not supported by the assigned processor hardware.

The priority of the task must be in the range 0 ... 79.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, specify a priority value for the task in the range 0 ... 79.

Build Configuration Set Conflicts

Build configuration set: Duplicate name

The active ConfigurationDesk application contains two or more build configuration sets with the same name.

The names of build configuration sets must be unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, enter unique names in the Name edit field for the build configuration sets.

Bus Access Request Conflicts

Bus access request: Duplicate name of bus access request

Two or more bus access requests have identical names.

The names of bus access requests must be unique within the ConfigurationDesk application.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify unique names for the affected bus access requests.

Bus access request: No bus access assigned

A bus access request is not assigned to a suitable bus function block (CAN, LIN).

If you want to build a real-time application, you must specify the bus access for each bus access request via a suitable bus function block (CAN, LIN). If you want to generate bus simulation containers, you do not have to specify the bus accesses.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy Choose one of the following alternatives:

- Assign all bus access requests of the active ConfigurationDesk application to bus accesses in one step: Switch to the Buses view set if necessary. On the Home ribbon, click Assign Automatically.
- Assign a bus access request individually to one bus access:
  - In the Bus Configurations table, select the affected bus access request and assign it to a bus access via the Bus Access Assignment command.
  - In the Conflicts Viewer, for example, specify a bus access for the Bus Access Request property (only possible if at least one suitable bus function block exists in the application).

Bus Configuration Conflicts

Bus configuration: Duplicate name of frame gateway or frame capture

Two or more frame gateway and/or frame capture nodes of a bus configuration have identical names.

The names of frame gateway and frame capture nodes must be unique within a bus configuration.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy In the Conflicts Viewer, for example, specify unique names for the affected frame gateway and frame capture nodes.

Bus configuration: No valid application process assigned

A bus configuration is assigned to no or more than one application process.

A bus configuration must be assigned to one application process with at least one task.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy The conflict causes and remedies that apply to your case depend on the use scenario. For more information, refer to:

- Bus Manager in ConfigurationDesk: Resolving the Bus Configuration Conflict: No Valid Application Process Assigned (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving the Bus Configuration Conflict: No Valid Application Process Assigned (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Bus configuration: Unresolved communication matrix elements assigned

Elements of a bus configuration use communication matrix elements that are not available in the active ConfigurationDesk application.

This conflict occurs if a communication matrix was deleted without deleting the related bus configuration elements.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting bus configuration element in the executable application, delete it from the bus configuration. In the Buses Browser, for example, select the communication matrix that is marked as unresolved and choose Delete Completely from the context menu.

- If you need the conflicting bus configuration element in the executable application, add the previously deleted communication matrix or a communication matrix with exactly the same content to your ConfigurationDesk application.

Tip

- When you select the communication matrix node in the **Buses Browser**, the **Properties Browser** displays the name and the file path of the deleted matrix.
- If you do not add the previously deleted communication matrix but one with identical content to the ConfigurationDesk application, the names of the two communication matrices can differ.

CAN Cluster Conflicts

CAN cluster: Invalid data phase baud rate (communication matrix)

A communication matrix does not define a data phase baud rate for a CAN FD cluster, or a defined data phase baud rate exceeds the valid range.

For each CAN FD cluster, the communication matrix must define a data phase baud rate in the range 0 … $[1.6 \cdot 10^7]$ (bit/s).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify a valid value for the **Data phase baud rate** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN cluster: Invalid speed (baud rate) defined (communication matrix)

A communication matrix does not define a baud rate for a CAN cluster, or a defined baud rate exceeds the valid range.

For each CAN cluster, the communication matrix must define a baud rate in the range 0 … [Long.max] (bit/s).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify a valid value for the **Baud rate** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN cluster: Too many channels connected (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned CAN cluster, the communication matrix specifies more than one channel.

For each CAN cluster, the communication matrix must not specify more than one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN cluster: Too many channels connected (communication matrix)

A communication matrix specifies more than one channel for a CAN cluster.

For each CAN cluster, the communication matrix must not specify more than one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN Communication Controller Conflicts

CAN communication controller: Too many channels connected (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned CAN communication controller, the communication matrix specifies more than one channel.

For each CAN communication controller, the communication matrix must not specify more than one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration. In the Conflicts Viewer, set the Is assigned property to False.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN communication controller: Too many channels connected (communication matrix)

A communication matrix specifies more than one channel for a CAN communication controller.

For each CAN communication controller, the communication matrix must not specify more than one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN Frame Conflicts

CAN frame: Too many PDU-to-frame mappings connected (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned CAN frame, the communication matrix specifies more than one PDU-to-frame mapping.

A CAN frame must not contain more than one PDU-to-frame mapping.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN frame: Too many PDU-to-frame mappings connected (communication matrix)

A communication matrix specifies more than one PDU-to-frame mapping for one CAN frame.

A CAN frame must not contain more than one PDU-to-frame mapping.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN Frame Triggering Conflicts

CAN frame triggering: Duplicate CAN identifier on CAN channel (bus configuration)

Conflicting communication matrix elements are assigned to a bus configuration. For two or more assigned CAN frame triggerings, the communication matrix specifies the following settings:

- The CAN frame triggerings are mapped to the same CAN channel.
- For at least two of these CAN frame triggerings, identical settings are specified for the following properties:
  - Extended addressing
  - Frame identifier

The tuple [extended addressing; frame identifier] must be unique for each CAN frame triggering of one CAN channel and within the following range:

- Extended addressing = true:
  Extended frame identifier (29-bit) within the range 0 … 2^{29} -1
- Extended addressing = false:
  Standard frame identifier (11-bit) within the range 0 … 2^{11} -1

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify unique and valid values for the **Extended addressing** and/or **Identifier** properties, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN frame triggering: Duplicate CAN identifier on CAN channel (communication matrix)

A communication matrix specifies the following settings for two or more CAN frame triggerings:

- The CAN frame triggerings are mapped to the same CAN channel.
- For at least two of these CAN frame triggerings, identical settings are specified for the following properties:
  - Extended addressing
  - Frame identifier

The tuple [extended addressing; frame identifier] must be unique for each CAN frame triggering of one CAN channel and within the following range:

- Extended addressing = true:

  Extended frame identifier (29-bit) within the range 0 … 2^{29} -1
- Extended addressing = false:

  Standard frame identifier (11-bit) within the range 0 … 2^{11} -1

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify unique and valid values for the **Extended addressing** and/or **Identifier** properties, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN frame triggering: Invalid frame type (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned CAN frame triggering, the communication matrix references a LIN frame.

Each CAN frame triggering must reference a CAN frame.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN frame triggering: Invalid frame type (communication matrix)

A communication matrix references a LIN frame for a CAN frame triggering.

Each CAN frame triggering must reference a CAN frame.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN frame triggering: Invalid property values (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned CAN frame triggering, the communication matrix specifies the following settings:

At least one of the following properties is unspecified or exceeds the valid range:

- Extended addressing
- Frame identifier
- CAN FD frame support
- Bit rate switch

For a CAN frame triggering, the communication matrix must specify the tuple [extended addressing; frame identifier; CAN FD frame support; bit rate switch] within the valid range.

The tuple [extended addressing; frame identifier] must be within the following range:

- Extended addressing = true:
  Extended frame identifier (29-bit) within the range 0 … 2^{29} -1
- Extended addressing = false:
  Standard frame identifier (11-bit) within the range 0 … 2^{11} -1

Valid values for the CAN FD frame support and the bit rate switch are *true* and *false*.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify valid values for the **Extended addressing**, **Identifier**, **Bit rate switch**, and/or **CAN FD frame support** properties, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN frame triggering: Invalid property values (communication matrix)

A communication matrix specifies the following settings for a CAN frame triggering:

At least one of the following properties is unspecified or exceeds the valid range:

- Extended addressing
- Frame identifier
- CAN FD frame support
- Bit rate switch

For CAN frame triggering, the communication matrix must specify the tuple [extended addressing; frame identifier; CAN FD frame support; bit rate switch] within the valid range.

The tuple [extended addressing; frame identifier] must be within the following range:

- Extended addressing = true:

  Extended frame identifier (29-bit) within the range 0 … 2^{29} -1
- Extended addressing = false:

  Standard frame identifier (11-bit) within the range 0 … 2^{11} -1

Valid values for the CAN FD frame support and the bit rate switch are *true* and *false*.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify valid values for the **Extended addressing**, **Identifier**, **Bit rate switch**, and/or **CAN FD frame support** properties, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN frame triggering: Mismatching payload length of referenced frame (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned CAN frame triggering, the communication matrix specifies the following settings: The communication matrix assigns a CAN frame to the CAN frame triggering and the frame's payload length does not match the frame triggering requirements.

To match the requirements of a CAN frame triggering, the payload length of an assigned CAN frame must be within the following range:

- CAN frame triggering according to CAN 2.0: Payload length within the range 0 … 8 byte
- CAN frame triggering according to CAN FD: Payload length within the range 0 … 64 byte

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify valid values for the **Length** and/or **CAN FD frame support** properties, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

CAN frame triggering: Mismatching payload length of referenced frame (communication matrix)

A communication matrix assigns a CAN frame to a CAN frame triggering and the frame's payload length does not match the frame triggering requirements.

To match the requirements of a CAN frame triggering, the payload length of an assigned CAN frame must be within the following range:

- CAN frame triggering according to CAN 2.0: Payload length within the range 0 … 8 byte
- CAN frame triggering according to CAN FD: Payload length within the range 0 … 64 byte

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify valid values for the **Length** and/or **CAN FD frame support** properties, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Channel Conflicts

Channel: Invalid number of LIN masters (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned LIN channel, the communication matrix specifies more than one LIN master. At least two of the LIN masters are assigned to the bus configuration.

For each LIN channel that is assigned to a bus configuration, only one LIN master can be assigned.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Channel: Invalid number of LIN masters (communication matrix)

A communication matrix specifies no or more than one LIN master for a LIN channel.

For each LIN channel, the communication matrix must specify exactly one LIN master.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Channel: No name defined (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned channel, the communication matrix does not define a name.

For each channel, the communication matrix must define a name.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Channel: No name defined (communication matrix)

A communication matrix does not define a name for a channel.

For each channel, the communication matrix must define a name.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Cluster Conflicts

Cluster: Duplicate name (bus configuration)

Conflicting communication matrix elements are assigned to a bus configuration. For two or more assigned communication clusters, the communication matrix defines identical names.

For each communication cluster, the communication matrix must define a unique name.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Cluster: Duplicate name (communication matrix)

A communication matrix defines identical names for two or more communication clusters.

For each communication cluster, the communication matrix must define a unique name.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Cluster: No name defined (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned communication cluster, the communication matrix does not define a name.

For each communication cluster, the communication matrix must define a unique name.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Cluster: No name defined (communication matrix)

A communication matrix does not define a name for a communication cluster.

For each communication cluster, the communication matrix must define a unique name.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Coding Conflicts

Coding: Unsupported data types (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned ISignal, the communication matrix specifies at least one of the following settings:

- For the coded ISignal value, no data type is specified.
- For the coded and/or physical ISignal value, an unsupported data type is specified.
- For the physical ISignal value, no data type is specified. In addition, a specified computation method is not supported.
- The ISignal is an array signal but the specified data type of the coded and/or physical ISignal value is not supported for array signals.

The communication matrix must specify the coding of ISignals as follows:

- For the coded ISignal value, the communication matrix must specify a data type that the Bus Manager supports.
- If the communication matrix specifies a data type for the physical ISignal value, the specified data type must be supported by the Bus Manager as well.
- If the communication matrix does not specify a data type for the physical ISignal value, it must either specify no computation method or a computation method that the Bus Manager supports. In these cases, the Bus Manager derives the physical data type from the coded data type and the computation method.
- If an ISignal is an array signal, the data type of the coded and physical ISignal value must be UInt8. Other data types are not supported by the Bus Manager.

For an overview of the supported data types, refer to:

- Bus Manager in ConfigurationDesk: Supported PDU Types and Signal Data Types (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Supported PDU Types and Signal Data Types (Bus Manager (Stand-Alone) Implementation Guide 🕮)

For information on supported computation methods, refer to:

- Bus Manager in ConfigurationDesk: Signal Conversion by the Bus Manager (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Signal Conversion by the Bus Manager (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Select valid values for the coded and/or physical **Base data type** properties, or change the signal type via the **Category** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

    Note

    Depending on the cause of the conflict, this remedy might not resolve the conflict. For example, if the conflicting base data type is not specified at all or the Bus Manager cannot derive it, the related **Base data type** property is not available in ConfigurationDesk.

  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Coding: Unsupported data types (communication matrix)

A communication matrix specifies at least one of the following settings for an ISignal:

- For the coded ISignal value, no data type is specified.
- For the coded and/or physical ISignal value, an unsupported data type is specified.
- For the physical ISignal value, no data type is specified. In addition, a specified computation method is not supported.

- The ISignal is an array signal but the specified data type of the coded and/or physical ISignal value is not supported for array signals.

The communication matrix must specify the coding of ISignals as follows:

- For the coded ISignal value, the communication matrix must specify a data type that the Bus Manager supports.
- If the communication matrix specifies a data type for the physical ISignal value, the specified data type must be supported by the Bus Manager as well.
- If the communication matrix does not specify a data type for the physical ISignal value, it must either specify no computation method or a computation method that the Bus Manager supports. In these cases, the Bus Manager derives the physical data type from the coded data type and the computation method.
- If an ISignal is an array signal, the data type of the coded and physical ISignal value must be UInt8. Other data types are not supported by the Bus Manager.

For an overview of the supported data types, refer to:

- Bus Manager in ConfigurationDesk: Supported PDU Types and Signal Data Types (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Supported PDU Types and Signal Data Types (Bus Manager (Stand-Alone) Implementation Guide 🕮)

For information on supported computation methods, refer to:

- Bus Manager in ConfigurationDesk: Signal Conversion by the Bus Manager (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Signal Conversion by the Bus Manager (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify valid values for the coded and/or physical **Base data type** properties, or change the signal type via the **Category** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

Note

Depending on the cause of the conflict, this remedy might not resolve the conflict. For example, if the conflicting base data type is not specified at all or the Bus Manager cannot derive it, the related **Base data type** property is not available in ConfigurationDesk.

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Coding: Unsupported or invalid linear scaling (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For a linear computation method of an assigned ISignal, the communication matrix specifies an unsupported or invalid computation scale.

To transmit or receive an ISignal via a bus, the ISignal must be encoded or decoded according to a computation method. If the communication matrix specifies a computation method for an ISignal, it must be valid and supported by the Bus Manager.

For more information, refer to:

- Bus Manager in ConfigurationDesk: Signal Conversion by the Bus Manager (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Signal Conversion by the Bus Manager (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify valid values for the numerator and/or denominator arrays: Select the related ISignal, for example, in the **Bus Configurations** table and specify valid values for the **Numerators** and **Denominators** properties in the **Properties Browser**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

Tip

The Conflicts Viewer displays the coding and the related computation method but not the ISignal itself. To access the related ISignal, right-click the coding and select Select Related Elements in Bus Configurations Table from the context menu.

For information on the valid values, refer to:

Bus Manager in ConfigurationDesk: Configurable Settings of ISignals (ConfigurationDesk Bus Manager Implementation Guide 🕮)

Bus Manager (stand-alone): Configurable Settings of ISignals (Bus Manager (Stand-Alone) Implementation Guide 🕮)

- Resolve the conflict by correcting the original communication matrix. Refer to:

  Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

  Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Coding: Unsupported or invalid linear scaling (communication matrix)

A communication matrix specifies an unsupported or invalid computation scale for a linear computation method of an ISignal.

To transmit or receive an ISignal via a bus, the ISignal must be encoded or decoded according to a computation method. If the communication matrix specifies a computation method for an ISignal, it must be valid and supported by the Bus Manager.

For more information, refer to:

- Bus Manager in ConfigurationDesk: Signal Conversion by the Bus Manager (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Signal Conversion by the Bus Manager (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify valid values for the numerator and/or denominator arrays: Select the related ISignal, for example, in the Buses Browser and specify valid values for the Numerators and Denominators properties in the Properties Browser. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

Tip

The **Conflicts Viewer** displays the coding and the related computation method but not the ISignal itself. If the names of the coding or computation method match the ISignal name, you can use the displayed names to search for the ISignal via the **Find** command, for example.

For information on the valid values, refer to:

- Bus Manager in ConfigurationDesk: Configurable Settings of ISignals (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Configurable Settings of ISignals (Bus Manager (Stand-Alone) Implementation Guide 🕮)

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Communication Connector Conflicts

Communication connector: Invalid number of communication controllers (communication matrix)

A communication matrix specifies no or more than one communication controller for an ECU connector.

For each ECU connector, the communication matrix must specify exactly one communication controller.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Communication Matrix Conflicts

Communication matrix: Duplicate name in bus configuration

Two or more different communication matrix nodes of a bus configuration have identical names.

The names of different communication matrix nodes in a bus configuration (e.g., all the communication matrix nodes below the **Bus access requests** node in the **Bus Configurations** table) must be unique within the bus configuration.

Tip

The name of a communication matrix node in a bus configuration can be user-defined. The name of a communication matrix itself cannot be changed in ConfigurationDesk. For details on the node name, refer to:

- Bus Manager in ConfigurationDesk: Basics on Bus Configurations (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Basics on Bus Configurations (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy In the **Conflicts Viewer**, for example, specify unique names for the affected communication matrix nodes.

Communication matrix: Failed consistency checks during import of communication matrix

A communication matrix is added to the active ConfigurationDesk application and this communication matrix contains inconsistent settings (e.g., a channel that references a coding instead of a frame triggering due to an incorrect ID).

Inconsistent settings can cause unintended behavior in ConfigurationDesk or at run time (e.g., due to incorrectly generated tasks). To prevent unintended behavior, the settings defined in a communication matrices must be consistent.

Tip

When you add the communication matrix to the active application, the **Message Viewer** provides a warning message including detailed information for each inconsistent setting.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Communication matrix: Failed schema check during import of communication matrix

A communication matrix that complies with AUTOSAR or FIBEX is added to the active ConfigurationDesk application. This communication matrix contains settings that are invalid according to the XML schema defined by the AUTOSAR or FIBEX standard, respectively.

Invalid settings can cause unintended behavior in ConfigurationDesk or at run time (e.g., due to incorrectly generated tasks). To prevent unintended behavior, the settings of AUTOSAR- or FIBEX-compliant communication matrices must be valid according to the AUTOSAR or FIBEX XML schema, respectively.

Tip

When you add the communication matrix to the active application, the Message Viewer provides a warning message including detailed information for each invalid setting.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Communication Package Conflicts

Communication package: Duplicate name

The active ConfigurationDesk application contains two or more communication packages with the same name.

The names of communication packages must be unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, enter unique names for the communication packages in the Name edit field.

Communication package: Existing closed loop of blocking model communications

The model implementations assigned to the same application process are connected via a closed loop with blocking model communication.

To enable ConfigurationDesk to optimize the configuration of application processes to which multiple model implementations are assigned, the following blocks of at least one model implementation must be assigned to a communication package with non-blocking communication:

All Data Inport blocks that are part of the closed loop of blocking communication.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy In the Model Communication Package table, assign the following blocks of at least one model implementation to a communication package with non-blocking communication:

All Data Inport blocks that are part of the closed loop of blocking communication.

Contained IPDU Conflicts

Contained IPDU: Duplicate header ID (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned IPDU, the communication matrix specifies the following settings:

- The IPDU is a contained IPDU that is included in a dynamic container IPDU (i.e., the header type of the container IPDU is `SHORT-HEADER` or `LONG-HEADER`).
- The IPDU's header ID is not unique for the header type in the applicable context.

For each contained IPDU, the communication matrix must specify a header ID that is unique for the header type (`SHORT-HEADER` or `LONG-HEADER`) in the applicable context, i.e.:

- If the contained IPDU is included in at least one container IPDU whose `RX-ACCEPT-CONTAINED-IPDU` attribute is set to `ACCEPT-ALL`, the header ID must be unique within the communication matrix.
- If the contained IPDU is included only in container IPDUs whose `RX-ACCEPT-CONTAINED-IPDU` attribute is set to `ACCEPT-CONFIGURED`, the header ID must be unique within each container IPDU.

If you do not resolve this conflict, default code is generated, i.e., even though the header ID is not unique in the applicable context, the related container IPDU can be transmitted and received at run time. However, it is unpredictable which IPDU is used as contained IPDU. This might result in unexpected or incorrect payload data of the container IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	-	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Remove the conflicting element from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- Correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Duplicate header ID (communication matrix)

A communication matrix specifies the following settings for an IPDU:

- The IPDU is a contained IPDU that is included in a dynamic container IPDU (i.e., the header type of the container IPDU is `SHORT-HEADER` or `LONG-HEADER`).
- The IPDU's header ID is not unique for the header type in the applicable context.

For each contained IPDU, the communication matrix must specify a header ID that is unique for the header type (`SHORT-HEADER` or `LONG-HEADER`) in the applicable context, i.e.:

- If the contained IPDU is included in at least one container IPDU whose `RX-ACCEPT-CONTAINED-IPDU` attribute is set to `ACCEPT-ALL`, the header ID must be unique within the communication matrix.
- If the contained IPDU is included only in container IPDUs whose `RX-ACCEPT-CONTAINED-IPDU` attribute is set to `ACCEPT-CONFIGURED`, the header ID must be unique within each container IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Invalid offset value (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned IPDU, the communication matrix specifies the following settings:

- The IPDU is a contained IPDU that is included in a static container IPDU (i.e., the header type of the container IPDU is `NO-HEADER`).
- For the contained IPDU, no offset value is specified or a specified offset value is invalid.

For a contained IPDU that is included in a static container IPDU, an offset value must be specified. The offset value must be within the range of the container IPDU and must not overlap with other offset values.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Invalid offset value (communication matrix)

A communication matrix specifies the following settings for an IPDU:

- The IPDU is a contained IPDU that is included in a static container IPDU (i.e., the header type of the container IPDU is `NO-HEADER`).
- For the contained IPDU, no offset value is specified or a specified offset value is invalid.

For a contained IPDU that is included in a static container IPDU, an offset value must be specified. The offset value must be within the range of the container IPDU and must not overlap with other offset values.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Invalid update bit position (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned IPDU, the communication matrix specifies the following settings:

- The IPDU is a contained IPDU that is included in a static container IPDU (i.e., the header type of the container IPDU is `NO-HEADER`).
- For the update bit position of the contained IPDU, one of the following settings is specified:
  - No update bit position is specified, and for the related static container IPDU no unused bit pattern is specified.
  - A specified update bit position is invalid.

For a contained IPDU that is included in a static container IPDU without a specified unused bit pattern, an update bit position must be specified. The update bit position must be within the range of the container IPDU and must not overlap with other update bit positions.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Invalid update bit position (communication matrix)

A communication matrix specifies the following settings for an IPDU:

- The IPDU is a contained IPDU that is included in a static container IPDU (i.e., the header type of the container IPDU is `NO-HEADER`).
- For the update bit position of the contained IPDU, one of the following settings is specified:
  - No update bit position is specified, and for the related static container IPDU no unused bit pattern is specified.
  - A specified update bit position is invalid.

For a contained IPDU that is included in a static container IPDU without a specified unused bit pattern, an update bit position must be specified. The update bit position must be within the range of the container IPDU and must not overlap with other update bit positions.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Mismatching collection semantics (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned IPDU, the communication matrix specifies the following settings:

- The IPDU is a contained IPDU (i.e., it is included in a container IPDU).
- The IPDU's collection semantics differs from other contained IPDUs of the same container IPDU.

Within one container IPDU, the collection semantics specified for all contained IPDUs must be identical (`LAST-IS-BEST` or `QUEUED`).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Mismatching collection semantics (communication matrix)

A communication matrix specifies the following settings for an IPDU:

- The IPDU is a contained IPDU (i.e., it is included in a container IPDU).
- The IPDU's collection semantics differs from other contained IPDUs of the same container IPDU.

Within one container IPDU, the collection semantics specified for all contained IPDUs must be identical (`LAST-IS-BEST` or `QUEUED`).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Mismatching header ID (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned IPDU, the communication matrix specifies the following settings:

- The IPDU is a contained IPDU that is included in a dynamic container IPDU (i.e., the header type of the container IPDU is `SHORT-HEADER` or `LONG-HEADER`).
- For the contained IPDU, the required header ID is not specified or a specified header ID is invalid.

A contained IPDU can have two header IDs, one for each header type (`SHORT-HEADER` or `LONG-HEADER`). The header type of the container IPDU determines which header ID of the contained IPDU is required. This header ID must be specified. Valid ranges for the header IDs are:

- Short header ID: 0 … 2^{24}-1
- Long header ID: 0 … 2^{32}-1

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Mismatching header ID (communication matrix)

A communication matrix specifies the following settings for an IPDU:

- The IPDU is a contained IPDU that is included in a dynamic container IPDU (i.e., the header type of the container IPDU is `SHORT-HEADER` or `LONG-HEADER`).
- For the contained IPDU, the required header ID is not specified or a specified header ID is invalid.

A contained IPDU can have two header IDs, one for each header type (`SHORT-HEADER` or `LONG-HEADER`). The header type of the container IPDU determines which header ID of the contained IPDU is required. This header ID must be specified. Valid ranges for the header IDs are:

- Short header ID: 0 … 2^{24}-1
- Long header ID: 0 … 2^{32}-1

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Mismatching IPDU length (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned IPDU, the communication matrix specifies the following settings: The IPDU is a contained IPDU (i.e., it is included in a container IPDU) but the IPDU's payload length is too long.

The maximum valid payload length of a contained IPDU depends on:

- The length and the header type of the container IPDU:
  - Container IPDU with `SHORT-HEADER` header type:
    $\texttt{contained\_IPDU\_payload}_{max} \leq \texttt{(container\_IPDU\_length - 32 bit)}$
  - Container IPDU with `LONG-HEADER` header type:
    $\texttt{contained\_IPDU\_payload}_{max} \leq \texttt{(container\_IPDU\_length - 64 bit)}$

  - Container IPDU with `NO-HEADER` header type:
    $contained\_IPDU\_payload_{max} \leq (container\_IPDU\_length)$
- The payload lengths of the other contained IPDUs that are included in the container IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    In the **Conflicts Viewer**, for example, extend the length of the container IPDU and/or reduce the length of the contained IPDUs via the related **Length** property.

    This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Mismatching IPDU length (communication matrix)

A communication matrix specifies the following settings for an IPDU: The IPDU is a contained IPDU (i.e., it is included in a container IPDU) but the IPDU's payload length is too long.

The maximum valid payload length of a contained IPDU depends on:

- The length and the header type of the container IPDU:
  - Container IPDU with `SHORT-HEADER` header type:
    $contained\_IPDU\_payload_{max} \leq (container\_IPDU\_length - 32\ bit)$
  - Container IPDU with `LONG-HEADER` header type:
    $contained\_IPDU\_payload_{max} \leq (container\_IPDU\_length - 64\ bit)$

  - Container IPDU with `NO-HEADER` header type:
    $contained\_IPDU\_payload_{max} \leq (container\_IPDU\_length)$
- The payload lengths of the other contained IPDUs that are included in the container IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  In the Conflicts Viewer, for example, extend the length of the container IPDU and/or reduce the length of the contained IPDUs via the related Length property.

  This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Multiple container IPDUs related (bus configuration)

Conflicting communication matrix elements are assigned to the Simulated ECUs part of a bus configuration. For assigned TX IPDUs, the communication matrix specifies the following settings:

- At least one assigned TX IPDU is a contained IPDU that is included in more than one container IPDU of the same ECU.
- At least two of the related container IPDUs of the same ECU are assigned as well.

For each ECU, a TX contained IPDU must be included in exactly one container IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting

element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Multiple container IPDUs related (communication matrix)

A communication matrix specifies the following settings for a TX IPDU: The TX IPDU is a contained IPDU that is included in more than one container IPDU of the same ECU.

For each ECU, a TX contained IPDU must be included in exactly one container IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: No timeout value (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned IPDU, the communication matrix specifies the following settings:

- The IPDU is a contained IPDU (i.e., it is included in a container IPDU).
- Neither for the contained IPDU nor for the container IPDU a timeout value is specified.

To transmit a container IPDU, at least one timeout value must be specified, either for the container IPDU or for all its contained IPDUs.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: No timeout value (communication matrix)

A communication matrix specifies the following settings for an IPDU:

- The IPDU is a contained IPDU (i.e., it is included in a container IPDU).
- Neither for the contained IPDU nor for the container IPDU a timeout value is specified.

To transmit a container IPDU, at least one timeout value must be specified, either for the container IPDU or for all its contained IPDUs.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Unsupported header byte order (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned IPDU, the communication matrix specifies the following settings:

- The IPDU is a contained IPDU that is included in a dynamic container IPDU (i.e., the header type of the container IPDU is `SHORT-HEADER` or `LONG-HEADER`).
- For the header of the contained IPDU, no or an unsupported byte order is specified.

For each contained IPDU, the byte order of the header must be specified. The byte order must be either `LITTLE_ENDIAN` or `BIG_ENDIAN`.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Unsupported header byte order (communication matrix)

A communication matrix specifies the following settings for an IPDU:

- The IPDU is a contained IPDU that is included in a dynamic container IPDU (i.e., the header type of the container IPDU is `SHORT-HEADER` or `LONG-HEADER`).
- For the header of the contained IPDU, no or an unsupported byte order is specified.

For each contained IPDU, the byte order of the header must be specified. The byte order must be either `LITTLE_ENDIAN` or `BIG_ENDIAN`.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Contained IPDU: Unsupported header ID 0 (communication matrix)

A communication matrix specifies the following settings for an IPDU:

- The IPDU is a contained IPDU that is included in a dynamic container IPDU (i.e., the header type of the container IPDU is `SHORT-HEADER` or `LONG-HEADER`).
- At least one of the contained IPDU's header IDs is 0.

A contained IPDU can have two header IDs, one for each header type (`SHORT-HEADER` or `LONG-HEADER`). None of the header IDs must be 0.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Container IPDU Conflicts

Container IPDU: Invalid header type (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned IPDU, the communication matrix specifies the following settings: The IPDU is a container IPDU (i.e., other IPDUs can be included in this IPDU) but the IPDU's header type is not specified or a specified header type is invalid.

For each container IPDU, the communication matrix must specify the header type. Valid header types are:

- `SHORT-HEADER`
- `LONG-HEADER`
- `NO-HEADER`

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting

element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Container IPDU: Invalid header type (communication matrix)

A communication matrix specifies the following settings for an IPDU: The IPDU is a container IPDU (i.e., other IPDUs can be included in this IPDU) but the IPDU's header type is not specified or a specified header type is invalid.

For each container IPDU, the communication matrix must specify the header type. Valid header types are:

- `SHORT-HEADER`
- `LONG-HEADER`
- `NO-HEADER`

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Container IPDU: Unsupported nested container (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned IPDU, the communication matrix specifies the following settings:

- The IPDU is a container IPDU (i.e., other IPDUs can be included in this IPDU).
- The IPDU contains an IPDU that is a container IPDU itself.

A container IPDU can contain only IPDUs that are no container IPDUs themselves.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Container IPDU: Unsupported nested container (communication matrix)

A communication matrix specifies the following settings for an IPDU:

- The IPDU is a container IPDU (i.e., other IPDUs can be included in this IPDU).
- The IPDU contains an IPDU that is a container IPDU itself.

A container IPDU can contain only IPDUs that are no container IPDUs themselves.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Device Block Conflicts

Device block: Unresolved elements

At least one device topology element is used in the signal chain, but missing in the device topology.

Such elements are unresolved and displayed with a warning symbol in the **External Device Browser**. For details, refer to Basics on Device Blocks (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy The Conflicts Viewer displays the missing device topology elements.

Choose one of the following alternatives:

- In the Conflicts Viewer, select the higher hierarchy element of the missing element in the Value column to resolve the conflict.
- In the External Device Browser, right-click the missing element(s) and select Make Resolved from the context menu.
- Replace the device topology by a version that contains the missing elements.
- Delete the element(s) from the signal chain of the active ConfigurationDesk application.

Device Conflicts

Device: Duplicate name

The active ConfigurationDesk application contains two or more device blocks with the same name.

Device block names must be unique in an application. Duplicate device names might cause problems when you import or export device data.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	✓

Remedy In the Conflicts Viewer, for example, enter unique names for the device blocks.

Device: Invalid pin assignment

A device pin assigned to a device port is part of another device.

For more information on device pin assignment, see Basics on Configuring External Devices (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy The Conflicts Viewer displays the affected device pin.

Choose one of the following alternatives:

- In the External Device Connectors table, move the device pin to the correct device block.
- In the Pins and External Wiring table, assign the device pin to a different device port.
- In the External Device Browser, adjust the device topology to match the device pin assignment.

Device: Invalid port assignment

The reference port(s) of a device port are located in a different device.

You must only assign reference ports that belong to the same device.

For more information on reference ports, see Basics on Configuring External Devices (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy The Conflicts Viewer displays the affected ports and their locations in the device topology (port group addresses).

Choose one of the following alternatives:

- In the External Device Browser, move all ports referencing each other to the same device.
- Remove the port references via the Properties Browser.

Device Pin Conflicts

Device pin: Duplicate name

The active ConfigurationDesk application contains two or more device pins with the same name.

Device pin names must be unique in a connector or device. Otherwise, exported device or pin data might become ambiguous.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	✓

Remedy In the Conflicts Viewer, enter unique names for the device pins.

Device Port Conflicts

Device port: Duplicate name

The active ConfigurationDesk application contains two or more device ports with the same name in a port group or a device.

Device port names must be unique in a port group or a device. Otherwise, exported device data might become ambiguous.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	✓

Remedy In the Conflicts Viewer, for example, enter unique names for the device ports.

Device Port Mapping Conflicts

Device port mapping: Mismatching port type

There is an unrecommended mapping between device ports or signal ports in the currently active ConfigurationDesk application.

Certain mappings between device and signal ports are not recommended, such as mapping a bidirectional device port and a signal reference port.

For more information, see Device Port Mapping (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- In a working view, adjust the device port mapping.
- In the Conflicts Viewer, for example, change the port type of the device port.

ECU Conflicts

ECU: Duplicate name (bus configuration)

Conflicting communication matrix elements are assigned to a bus configuration. For two or more assigned ECUs, the communication matrix defines identical names.

For each ECU, the communication matrix must define a unique name.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ECU: Duplicate name (communication matrix)

A communication matrix defines identical names for two or more ECUs.

For each ECU, the communication matrix must define a unique name.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Event Conflicts

Event: Delay value of delayed event exceeds limits of assigned source task period

The Delay value of the delayed event exceeds the period limits of the timer event that triggers the source task.

The Delay value must be smaller than two times the period of the timer event that triggers the source task.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, specify a Delay value in the valid range.

Event: Invalid source task for delayed event

An invalid source task is specified for the delayed event.

A source task must be a task other than the task to be delayed.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	–	–	–

Remedy In the Conflicts Viewer, for example, select a task other than the task to be delayed as the source task from the Value list.

Event: No source task assigned to delayed event

No source task is specified for the delayed event.

A valid source task must be specified for the delayed event.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy In the Conflicts Viewer, for example, select a valid source task from the Value list.

Event: Multiple assignments to different tasks

There is an event assigned to different tasks in an application process.

An event must be assigned to only one task in an application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, select exactly one task that the event must be assigned to in the Value column (next to the Assigned to field).

Event: Multiple timer events are configured to be sent via the same Gigalink channel

Two or more timer events in the executable application are configured to be sent via Gigalink, and the same combination of Gigalink Number and Channel Number is used for these timer events.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, configure different combinations of Gigalink number and Channel number for the timer events.

Executable Application Conflicts

Executable application: Duplicate application process names

The executable application contains two or more application processes with the same name.

Application process names must be unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, enter unique names in the Name edit field for all application processes.

Executable application: Duplicate DAQ raster name for nonsynchronous tasks

Two or more tasks are assigned to different application processes with one of the following modeling constellations:

- Case 1: The tasks are triggered by timer events with different periods, and have the same DAQ raster name specified.
- Case 2: The tasks are not triggered by timer events, and have the same DAQ raster name specified.
- Case 3: At least one task is triggered by a timer event, and at least one task is not triggered by a timer event. The tasks have the same DAQ raster name specified.

If two application processes contain tasks with the same DAQ raster name, the experiment tool interprets this as meaning that these tasks are triggered at the same times, and that signals from the two application processes can be recorded together via the same DAQ raster. However, if the rasters have the same name but different trigger times, this can have side-effects on the measurement.

Effects (cases 1 and 3)

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Effects (case 2)

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy In the Conflicts Viewer, for example, enter unique DAQ raster names for the tasks in the DAQ raster name edit field.

Executable application: Duplicate XCP service port

At least two application processes with an assigned V-ECU implementation have the same XCP service port specified.

XCP ports must be unique for application processes with assigned V-ECU implementations.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, specify unique XCP ports for the application processes in the XCP service port edit field.

Executable application: Mismatching DAQ raster names for tasks with same timer event periods

There are two or more periodic tasks assigned to different application processes. The timer events triggering the tasks have the same period. The DAQ raster names specified for the tasks are different (not empty).

If two application processes contain tasks triggered at the same times, it is useful to enter the same DAQ raster name for the tasks. In the experiment tool, the signals from the two application processes can then be recorded together via the same DAQ raster. A warning is issued during the build process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy In the Conflicts Viewer, for example, enter identical names in the DAQ raster name edit field for the tasks.

Executable application: No processing unit application defined

There is no processing unit application defined in the executable application.

An executable application must have at least one processing unit application.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Processing Resource Assignment table, create a processing unit application via the New - Processing Unit Application or the New - Multiple Processing Unit Application command.

Executable application: Too many processing unit applications defined (max. is 32)

There are too many processing unit applications defined in the executable application.

The maximum allowed number of processing unit applications is 32.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Delete excess processing unit applications from the executable application.

Executable application: Unsupported hardware topology. MicroAutoBox III hardware is not supported in multi-processing-unit applications.

A MicroAutoBox III processor board is assigned to a processing unit application that is contained in a multi-processing-unit application.

Using MicroAutoBox III is not supported in multi-processing-unit applications.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	–	–	–

Remedy In the Conflicts Viewer, for example, assign a processing unit that is supported in multi-processing-unit applications to the relevant processing unit application.

External Cable Harness Conflicts

External cable harness: Mismatching pin set assignment

The pin(s) assigned to a device port are not equivalent to the wiring information in the external cable harness of the active ConfigurationDesk application.

For more information, refer to Calculating an External Cable Harness (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, adjust the pin set assignment to match the external cable harness.

- Recalculate the wiring information. Note that if you want to continue using an existing physical cable harness, you must not recalculate the wiring information. For instructions, refer to How to (Re)Calculate the External Cable Harness (ConfigurationDesk Real-Time Implementation Guide 🕮).

External cable harness: Missing wiring connections

There are device port mapping lines which are not represented in the calculated external cable harness of the currently active ConfigurationDesk application.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy The Conflicts Viewer displays the mapped device and/or signal ports.

Choose one of the following alternatives:

- In a working view, remove the mapping lines by using the context menu command Delete from Application.
- Recalculate the wiring information. Note that if you want to continue using an existing physical cable harness, you must not recalculate the wiring information. For instructions, refer to How to (Re)Calculate the External Cable Harness (ConfigurationDesk Real-Time Implementation Guide 🕮).

External cable harness: Too many connected ports (compared to device mapping)

The external cable harness contains a connection between pins of device ports and/or signal ports that is not represented in the logical signal chain (i.e., the device port mapping lines are missing).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy The Conflicts Viewer displays the device and signal ports that need to be mapped.

Choose one of the following alternatives:

- Complete the mapping in a working view.
- Recalculate the wiring information. Note that if you want to continue using an existing physical cable harness, you must not recalculate the wiring information. For instructions, refer to How to (Re)Calculate the External Cable Harness (ConfigurationDesk Real-Time Implementation Guide 🕮).

Feature Conflicts

Feature: Duplicate feature switch value

One or more bus manipulation features are added to an ISignal. For the feature switch of the ISignal, two or more identical values are available.

All the values that are available for the feature switch of an ISignal must be unambiguous.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting features nor the ISignal in the executable application, remove the ISignal from the bus configuration. In the Conflicts Viewer, for example, right-click the ISignal and choose Delete from Application from the context menu.
- If you need only one of the conflicting features in your executable application, disable all other conflicting features. In the Conflicts Viewer, for example, set the Enabled property of the related feature nodes to False.
- If you need all the conflicting features in your executable application, assign the ISignal to different bus configurations. Disable all but one of the conflicting features for the ISignal of this bus configuration. For each of the other bus configurations, add only one of the conflicting features to the ISignal.

Feature: Invalid feature switch value

One or more bus manipulation features are added to an ISignal. The specified value of the feature switch references a bus manipulation feature that was disabled for the ISignal.

The value that is specified for the feature switch of an ISignal must correspond to a bus manipulation feature that is added to the ISignal.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting feature nor the related bus configuration element in the executable application, remove the bus configuration element from the bus configuration. In the Conflicts Viewer, for example, right-click the bus configuration element and choose Delete from Application from the context menu.

- If you do not need the disabled bus manipulation feature for the affected ISignal, specify a valid value for the feature switch. In the Conflicts Viewer, for example, select an available value for the Feature switch property.
- If you need the disabled bus manipulation feature for the affected ISignal, add it to the ISignal. In the Conflicts Viewer, for example, right-click the affected ISignal and select Show - Show in Bus Manipulation Features Table from the context menu. In the Bus Manipulation Features table, select Enabled from the related bus manipulation feature's column.

Feature: Invalid property values for user code (Inspection)

The PDU User Code feature is added to a PDU. For the feature, at least one of the following settings is specified:

- No user code ID is specified.
- An ISignal is assigned to a user signal but according to the communication matrix this ISignal is not included in the PDU.

For the PDU User Code feature, the following conditions apply:

- A user code ID must be specified.
- ISignals can be assigned to user signals only if they are included in the related PDU according to the specifications of the communication matrix.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting feature nor the related bus configuration element in the executable application, remove the bus configuration element from the bus configuration. In the Conflicts Viewer, for example, right-click the bus configuration element and choose Delete from Application from the context menu.
- If you do not need the conflicting feature in the executable application, disable it for the bus configuration. In the Conflicts Viewer, set the feature's Enabled property to False to disable only this conflicting feature. Alternatively, set the bus configuration's Enabled property to False to disable all conflicting features in one step.
- If you need the conflicting feature in the executable application, specify valid values for the User code ID and User signal [<number>] properties, for example, in the Conflicts Viewer.

Feature: Invalid property values for user code (Manipulation)

The PDU User Code feature is added to a PDU. For the feature, at least one of the following settings is specified:

- No user code ID is specified.
- An ISignal is assigned to a user signal but according to the communication matrix this ISignal is not included in the PDU.

For the PDU User Code feature, the following conditions apply:

- A user code ID must be specified.
- ISignals can be assigned to user signals only if they are included in the related PDU according to the specifications of the communication matrix.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting feature nor the related bus configuration element in the executable application, remove the bus configuration element from the bus configuration. In the Conflicts Viewer, for example, right-click the bus configuration element and choose Delete from Application from the context menu.
- If you do not need the conflicting feature in the executable application, disable it for the bus configuration. In the Conflicts Viewer, set the feature's Enabled property to False to disable only this conflicting feature. Alternatively, set the bus configuration's Enabled property to False to disable all conflicting features in one step.
- If you need the conflicting feature in the executable application, specify valid values for the User code ID and User signal [<number>] properties, for example, in the Conflicts Viewer.

Feature: Invalid property values for user code (Simulated ECUs)

The PDU User Code feature is added to a PDU. For the feature, at least one of the following settings is specified:

- No user code ID is specified.
- An ISignal is assigned to a user signal but according to the communication matrix this ISignal is not included in the PDU.

For the PDU User Code feature, the following conditions apply:

- A user code ID must be specified.
- ISignals can be assigned to user signals only if they are included in the related PDU according to the specifications of the communication matrix.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting feature nor the related bus configuration element in the executable application, remove the bus configuration element from the bus configuration. In the Conflicts Viewer, for example, right-click

the bus configuration element and choose Delete from Application from the context menu.

- If you do not need the conflicting feature in the executable application, disable it for the bus configuration. In the Conflicts Viewer, set the feature's Enabled property to False to disable only this conflicting feature. Alternatively, set the bus configuration's Enabled property to False to disable all conflicting features in one step.
- If you need the conflicting feature in the executable application, specify valid values for the User code ID and User signal [<number>] properties, for example, in the Conflicts Viewer.

Feature: Invalid property values of counter signal

The Counter Signal feature is added to an ISignal. At least one of the following conflicting counter settings is specified for the feature:

- `maximum value ≤ minimum value`
- `increment value > (maximum value - minimum value)`
- `initial value < minimum value`
- `initial value > maximum value`

The counter settings of the Counter Signal feature must be specified as follows:

- `maximum value > minimum value`
- `increment value ≤ (maximum value - minimum value)`
- `initial value ≥ minimum value`
- `initial value ≤ maximum value`

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting feature nor the related bus configuration element in the executable application, remove the bus configuration element from the bus configuration. In the Conflicts Viewer, for example, right-click the bus configuration element and choose Delete from Application from the context menu.
- If you do not need the conflicting feature in the executable application, disable it for the bus configuration. In the Conflicts Viewer, set the feature's Enabled property to False to disable only this conflicting feature. Alternatively, set the bus configuration's Enabled property to False to disable all conflicting features in one step.
- If you need the conflicting feature in the executable application, specify valid values for the Minimum value, Maximum value, Increment value, and/or Initial value properties of the counter signal, for example, in the Conflicts Viewer.

Feature: Mismatching counter signal range or unsupported physical data type

The Counter Signal feature is added to an ISignal and at least one of the following conflicting settings is specified:

- The specified physical base data type of the ISignal is not supported by the Counter Signal feature.
- The specified counter minimum value or counter maximum value exceeds the valid counter signal range.

To use the Counter Signal feature, the physical base data type of the ISignal must be an integer type. Additionally, the specified counter minimum and maximum values must be within the valid counter signal range. The valid counter signal range is determined by the ISignal's physical and coded base data types, and the ISignal's length.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting feature nor the related bus configuration element in the executable application, remove the bus configuration element from the bus configuration. In the Conflicts Viewer, for example, right-click the bus configuration element and choose Delete from Application from the context menu.
- If you do not need the conflicting feature in the executable application, disable it for the bus configuration. In the Conflicts Viewer, set the feature's Enabled property to False to disable only this conflicting feature. Alternatively, set the bus configuration's Enabled property to False to disable all conflicting features in one step.
- If you need the conflicting feature in the executable application, specify valid values for the conflicting settings. Depending on the conflict cause, the conflicting settings can be the following:
  - Physical base data type of the ISignal
  - Coded base data type of the ISignal
  - Length of the ISignal
  - Maximum counter value
  - Minimum counter value

  You can access the settings, for example, in the Conflicts Viewer and/or in the Properties Browser when you select the related ISignal.

  Changing the physical or coded base data type or the ISignal's length changes the communication matrix in the active ConfigurationDesk application and all ISignal instances in all bus configurations of the application. The originally imported communication matrix remains unchanged. Changing the maximum or minimum counter value changes only the counter signal in the related bus configuration. Other counter signals, bus configurations, and the communication matrix remain unchanged.

Feature: Mismatching port settings (length)

The Frame Access feature is added to a PDU. A value that is specified for the Length function port of the feature exceeds the value that is specified for the Maximum length property of the feature.

For the Length funtion port of the Frame Access feature, the values of the Initial value, Initial substitute value, Saturation minimum value, and Saturation maximum value properties must be equal to or smaller than the value of the Maximum length property of the feature.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting feature nor the related bus configuration element in the executable application, remove the bus configuration element from the bus configuration. In the Conflicts Viewer, for example, right-click the bus configuration element and choose Delete from Application from the context menu.
- If you do not need the conflicting feature in the executable application, disable it for the bus configuration. In the Conflicts Viewer, set the feature's Enabled property to False to disable only this conflicting feature. Alternatively, set the bus configuration's Enabled property to False to disable all conflicting features in one step.
- If you need the conflicting feature in the executable application, choose one of the following alternatives: In the Conflicts Viewer, for example, do the following:
  - Extend the maximum frame length via the Maximum length property of the Frame Access feature.
  - Specify valid values for the Initial value, Initial substitute value, Saturation minimum value, and Saturation maximum value properties of the Length function port.

Feature: Mismatching port settings (length)

The Frame Length feature is added to a PDU. A value that is specified for the Length function port of the feature exceeds the maximum valid length of the related frame.

For the Length function port of the Frame Length feature, the values of the Initial value, Initial substitute value, Saturation minimum value, and Saturation maximum value properties must be in the range `0 byte ...<maximum valid length>` of the related frame. The maximum valid length depends on whether the frame is a classic CAN frame or CAN FD frame:

- Classic CAN frame: `<maximum valid length> = 8 bytes`
- CAN FD frame: `<maximum valid length> = 64 bytes`

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting feature nor the related bus configuration element in the executable application, remove the bus configuration element from the bus configuration. In the Conflicts Viewer, for example, right-click the bus configuration element and choose Delete from Application from the context menu.
- If you do not need the conflicting feature in the executable application, disable it for the bus configuration. In the Conflicts Viewer, set the feature's Enabled property to False to disable only this conflicting feature. Alternatively, set the bus configuration's Enabled property to False to disable all conflicting features in one step.
- If you need the conflicting feature in the executable application, choose one of the following alternatives:
  - Specify valid values for the Initial value, Initial substitute value, Saturation minimum value, and Saturation maximum value properties of the Length function port, for example, in the Conflicts Viewer.
  - Enable CAN FD support for the frame. To do so, select the related PDU, for example, in the Conflicts Viewer. On the Bus Communication page of the Properties Browser, set the CAN FD frame support property of the frame to True.

    This changes the communication matrix in the active ConfigurationDesk application and all frame instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

Feature: Mismatching port settings (length)

The PDU Length feature is added to a PDU. A value that is specified for the Length function port of the feature exceeds the maximum valid PDU length.

For the Length funtion port of the PDU Length feature, the values of the Initial value, Initial substitute value, Saturation minimum value, and Saturation maximum value properties must be in the range `0 … <PDU length specified in the communication matrix>`.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting feature nor the related bus configuration element in the executable application, remove the bus configuration element from the bus configuration. In the **Conflicts Viewer**, for example, right-click the bus configuration element and choose **Delete from Application** from the context menu.
- If you do not need the conflicting feature in the executable application, disable it for the bus configuration. In the **Conflicts Viewer**, set the feature's **Enabled** property to **False** to disable only this conflicting feature. Alternatively, set the bus configuration's **Enabled** property to **False** to disable all conflicting features in one step.
- If you need the conflicting feature in the executable application, choose one of the following alternatives:
  - Specify valid values for the **Initial value**, **Initial substitute value**, **Saturation minimum value**, and **Saturation maximum value** properties of the Length function port, for example, in the **Conflicts Viewer**.
  - Extend the PDU length in the communication matrix. To do so, select the PDU, for example, in the **Conflicts Viewer**. On the **General** page of the **Properties Browser**, extend the PDU length via the **Length** property.

    This changes the communication matrix in the active ConfigurationDesk application and all PDU instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

Feature: Unsupported bus configuration feature enabled

A bus configuration feature is enabled which is no longer supported for the affected bus configuration element.

Settings of the bus configuration element have changed in the communication matrix, e.g., because the assigned communication matrix was replaced in the bus configuration. Due to these changes, the bus configuration feature is no longer supported for the bus configuration element.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting feature nor the related bus configuration element in the executable application, remove the bus configuration element from the bus configuration. In the **Conflicts Viewer**, for example, right-click the bus configuration element and choose **Delete from Application** from the context menu.
- Disable the conflicting feature for the bus configuration: In the **Conflicts Viewer**, set the feature's **Enabled** property to **False** to disable only this conflicting feature. Alternatively, set the bus configuration's **Enabled** property to **False** to disable all conflicting features in one step.

Feature: Unsupported combination of Counter Signal feature and ISignal Value feature

The Counter Signal feature and the ISignal Value feature are added to a TX ISignal.

When you add the Counter Signal feature to a TX ISignal, the ISignal Value feature cannot be used for the ISignal. If both features are added to the ISignal, the ISignal Value feature is ignored. At run time, only the Counter Signal feature applies to the TX ISignal.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting features nor the ISignal in the executable application, remove the ISignal from the bus configuration. In the Conflicts Viewer, for example, right-click the ISignal and choose Delete from Application from the context menu.
- If you do not need the Counter Signal feature in the executable application, disable it for the ISignal. In the Conflicts Viewer, for example, set the Enabled property of the Counter Signal node to False.
- If you need only the Counter Signal feature in the executable application, disable the ISignal Value feature: In the Conflicts Viewer, set the Enabled property of the ISignal Value feature to False to disable the feature for an individual ISignal. Alternatively, set the bus configuration's Enabled property to False to disable all conflicting ISignal Value features in one step.
- If you need all the conflicting features in your executable application, assign the ISignals to different bus configurations. Disable all but one of the conflicting features for this bus configuration. For each of the other bus configurations, add only one of the conflicting features to the ISignals.

Feature: Unsupported combination of Frame Access feature and other PDU or signal features

The Frame Access feature is added to a PDU while other bus simulation features are added to the PDU or to elements that are included in the PDU.

When you add the Frame Access feature to a PDU, all other bus simulation features must be disabled for the PDU and its included elements. Included elements are:

- Contained IPDUs if you add the feature to a container IPDU.
- ISignal PDUs that are configured as the static or dynamic part if you add the feature to a multiplexed IPDU.
- ISignals that are directly included in the PDU or in any of its included PDUs.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting features nor the PDU and its included elements in the executable application, remove the PDU from the bus configuration. In the Conflicts Viewer, for example, right-click the PDU and choose Delete from Application from the context menu.
- If you do not need the Frame Access feature in the executable application, disable it for the PDU. In the Conflicts Viewer, for example, set the Enabled property of the Frame Access node to False.
- If you need only the Frame Access feature in the executable application, disable all other conflicting bus simulation features: In the Conflicts Viewer, set the Enabled property of a conflicting feature to False to disable only this bus simulation feature. Alternatively, set the bus configuration's Enabled property to False to disable all conflicting bus simulation features in one step.
- If you need all the conflicting features in your executable application, assign the PDU and its included elements to different bus configurations. Disable all but one of the conflicting features for this bus configuration. For each of the other bus configurations, add only one of the conflicting features to the PDU and/or its included elements.

Feature: Unsupported combination of PDU Raw Data feature and signal features

The PDU Raw Data feature is added to a TX PDU while bus simulation features are added to ISignals that are included in the TX PDU. For the PDU Raw Data feature, the Access mode property is set to Write.

When you write raw data to a TX PDU by using the PDU Raw Data feature, no bus simulation features can be used for the ISignals that are included in the TX PDU. If bus simulation features are added to the ISignals, these features are ignored. At run time, only the PDU Raw Data feature applies to the TX PDU, i.e., raw data is written to the TX PDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting features nor the PDU and its included ISignals in the executable application, remove the PDU from the bus configuration. In the Conflicts Viewer, for example, right-click the PDU and choose Delete from Application from the context menu.

- If you do not need the PDU Raw Data feature in the executable application, disable it for the PDU. In the Conflicts Viewer, for example, set the Enabled property of the PDU Raw Data node to False.
- If you only want to read the raw data of the TX PDU, set the Access mode property of the PDU Raw Data feature to Read, for example, in the Conflicts Viewer.
- If you only want to write raw data to the TX PDU, disable all conflicting bus simulation features except the PDU Raw Data feature: In the Conflicts Viewer, set the Enabled property of a conflicting feature to False to disable only this bus simulation feature. Alternatively, set the bus configuration's Enabled property to False to disable all conflicting bus simulation features in one step.
- If you want to write raw data to the TX PDU and you need all the conflicting features in your executable application, assign the PDU and its included ISignals to different bus configurations. Disable all but one of the conflicting features for this bus configuration. For each of the other bus configurations, add only one of the conflicting features to the PDU and/or its included ISignals.

Feature: Unsupported physical base data type

The ISignal Offset Value feature is added to an ISignal but the specified physical base data type of the ISignal is not supported by the feature.

To use the ISignal Offset Value feature, the physical base data type of the ISignal must be an integer type.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you need neither the conflicting feature nor the related bus configuration element in the executable application, remove the bus configuration element from the bus configuration. In the Conflicts Viewer, for example, right-click the bus configuration element and choose Delete from Application from the context menu.
- If you do not need the conflicting feature in the executable application, disable it for the bus configuration. In the Conflicts Viewer, set the feature's Enabled property to False to disable only this conflicting feature. Alternatively, set the bus configuration's Enabled property to False to disable all conflicting features in one step.
- If you need the conflicting feature in the executable application, specify a valid physical base data type for the ISignal. In the Conflicts Viewer, for example, select an available value for the Physical data type property.

  This changes the communication matrix in the active ConfigurationDesk application and all ISignal instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

Frame Conflicts

Frame: Exceeding PDUs and/or update bits (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned frame, the communication matrix specifies PDU-to-frame mappings and due to these mappings, PDUs and/or update bits exceed the frame boundaries.

The communication matrix must specify PDU-to-frame mappings in such a way that PDUs and update bits are within the boundaries of the frame.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Extend the length of the frame and/or reduce the length of the conflicting PDUs: Specify suitable values for the related **Length** property, for example, in the **Conflicts Viewer**. If required, you can extend the maximum frame length by enabling CAN FD support. To do so, set **CAN FD frame support** to **True**, for example, in the **Conflicts Viewer**.

    This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

    Note

    Depending on the PDU-to-frame mappings, this remedy might not resolve the conflict (e.g., if two mappings specify identical start positions for their related PDUs).

  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Frame: Exceeding PDUs and/or update bits (communication matrix)

A communication matrix specifies PDU-to-frame mappings for a frame and due to these mappings, PDUs and/or update bits exceed the frame boundaries.

The communication matrix must specify PDU-to-frame mappings in such a way that PDUs and update bits are within the boundaries of the frame.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Extend the length of the frame and/or reduce the length of the conflicting PDUs: Specify suitable values for the related Length property, for example, in the Conflicts Viewer. If required, you can extend the maximum frame length by enabling CAN FD support. To do so, set CAN FD frame support to True, for example, in the Conflicts Viewer.

  This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

  Note

  Depending on the PDU-to-frame mappings, this remedy might not resolve the conflict (e.g., if two mappings specify identical start positions for their related PDUs).

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Frame: Invalid byte layout for PDU-to-frame mapping (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned frame, the communication matrix specifies at least one of the following settings:

- For a PDU-to-frame mapping of the frame, no or more than one byte layout is defined.
- The position of the start bit is unspecified.
- The endianness is unspecified or invalid.

For each PDU-to-frame mapping of a frame, the communication matrix must define exactly one byte layout. The start bit position and endianness must be specified. The start bit position must be ≥ 0. The endianness must be either big endian or little endian.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Frame: Invalid byte layout for PDU-to-frame mapping (communication matrix)

A communication matrix contains a frame with at least one of the following settings:

- For a PDU-to-frame mapping of the frame, no or more than one byte layout is defined.
- The position of the start bit is unspecified.
- The endianness is unspecified or invalid.

For each PDU-to-frame mapping of a frame, the communication matrix must define exactly one byte layout. The start bit position and endianness must be specified. The start bit position must be ≥ 0. The endianness must be either big endian or little endian.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Frame: No name defined (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned frame, the communication matrix does not define a name.

For each frame, the communication matrix must define a unique name.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Frame: No name defined (communication matrix)

A communication matrix does not define a name for a frame.

For each frame, the communication matrix must define a unique name.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Frame: No payload length defined (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned frame, the communication matrix does not define a payload length.

For each frame, the communication matrix must define a payload length.

Tip

This rule does not apply to sporadic and event-triggered LIN frames.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify a valid value for the **Length** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Frame: No payload length defined (communication matrix)

A communication matrix does not define a payload length for a frame.

For each frame, the communication matrix must define a payload length.

Tip

This rule does not apply to sporadic and event-triggered LIN frames.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify a valid value for the **Length** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Frame: Overlapping PDUs and/or update bits (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned frame, the communication matrix specifies PDU-to-frame mappings and due to these mappings, PDUs and/or update bits overlap each other within the frame.

The communication matrix must specify PDU-to-frame mappings in such a way that PDUs and update bits do not overlap each other within the frame.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Reduce the length of the conflicting PDUs: Specify suitable values for the related **Length** property, for example, in the **Conflicts Viewer**.

This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

Note

Depending on the PDU-to-frame mappings, this remedy might not resolve the conflict (e.g., if two mappings specify identical start positions for their related PDUs).

- Resolve the conflict by correcting the original communication matrix. Refer to:

  Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

  Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Frame: Overlapping PDUs and/or update bits (communication matrix)

A communication matrix specifies PDU-to-frame mappings for a frame and due to these mappings, PDUs and/or update bits overlap each other within the frame.

The communication matrix must specify PDU-to-frame mappings in such a way that PDUs and update bits do not overlap each other within the frame.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Reduce the length of the conflicting PDUs: Specify suitable values for the related **Length** property, for example, in the **Conflicts Viewer**.

  This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

  Note

  Depending on the PDU-to-frame mappings, this remedy might not resolve the conflict (e.g., if two mappings specify identical start positions for their related PDUs).

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Function Block Conflicts

Function block: Disabled block provider assigned

The function block references another function block to use one of its features, but the feature is disabled.

Example: The function block references a Multi-Channel PWM Out function block to use the block's generated trigger signal as a trigger source. However, the related I/O function trigger generation property is set to Disabled.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Enable the required feature at the referenced function block.
- Reference a function block where the required feature is set to enabled.

Function block: Duplicate angle values and/or supported angle range exceeded

The value specified at the Camshaft angles property is invalid, because at least one of the following cases occurs:

- Duplicate angle values are specified.
- The covered angle range exceeds the angle range of the assigned master APU provider.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify values for the Camshaft angle property that are unique and match the angle range of the assigned master APU provider.

Function block: Duplicate angle values and/or supported angle range exceeded

One or more properties (Initial value, Stop value, Initial substitute value) of the related function port contain invalid values caused by the following:

- One or more of the properties contain duplicate angle values.
- The covered angle range of one or more properties exceeds the angle range of the assigned Engine Control Setup function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify values at the affected properties of the related function port that are unique and match the angle range of the assigned master APU provider.

Function block: Duplicate CAN channel name

The active ConfigurationDesk application contains two or more CAN function blocks that specify identical CAN channel names.

CAN channel names must be unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify unique names for the Channel name properties of the CAN function blocks.

Function block: Duplicate ECU interface container code prefix

Two or more ECU Interface Configuration function blocks are available in the active ConfigurationDesk application. ECU interface container (EIC) files with identical code prefixes are imported to these function blocks.

The code prefix of an ECU interface container must be unique within the active ConfigurationDesk application, i.e.:

- Each ECU interface container must be available in the application only once.
- If several ECU interface containers are available in the application, their code prefixes must differ.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Import different ECU interface containers to the ECU Interface Configuration function blocks. For instructions, refer to How to Import ECU Interface Container (EIC) Files (ConfigurationDesk Real-Time Implementation Guide 🕮).
- For the affected ECU interface containers, correct the duplicate settings via the dSPACE ECU Interface Manager. Export new EIC files and update the affected ECU Interface Configuration function blocks with these EIC files. For more information on updating EIC files, refer to Basics on Updating ECU Interface Containers in ConfigurationDesk Applications (ConfigurationDesk Real-Time Implementation Guide 🕮).
- In the ConfigurationDesk application, delete all ECU Interface Configuration function blocks except for one.

Function block: Duplicate ECU interface container identification

Two or more ECU Interface Configuration function blocks are available in the active ConfigurationDesk application. ECU interface container (EIC) files with identical container identifications are imported to these function blocks.

The container identification of an ECU interface container must be unique within the active ConfigurationDesk application, i.e.:

- Each ECU interface container must be available in the application only once.
- If several ECU interface containers are available in the application, their container identifications must differ.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Import different ECU interface containers to the ECU Interface Configuration function blocks. For instructions, refer to How to Import ECU Interface Container (EIC) Files (ConfigurationDesk Real-Time Implementation Guide 🕮).
- For the affected ECU interface containers, correct the duplicate settings via the dSPACE ECU Interface Manager. Export new EIC files and update the affected ECU Interface Configuration function blocks with these EIC files. For more information on updating EIC files, refer to Basics on Updating ECU Interface Containers in ConfigurationDesk Applications (ConfigurationDesk Real-Time Implementation Guide 🕮).

- In the ConfigurationDesk application, delete all ECU Interface Configuration function blocks except for one.

Function block: Duplicate IP address

The local IP addresses of at least one virtual Ethernet controller is not unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts viewer, for example, specify unique local IP addresses for the virtual Ethernet controllers.

Function block: Duplicate LIN channel name

The active ConfigurationDesk application contains two or more LIN function blocks that specify identical LIN channel names.

LIN channel names must be unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify unique names for the Channel name properties of the LIN function blocks.

Function block: Duplicate MAC address

At least one of the following cases occurs:

- The custom MAC address of the virtual Ethernet controller is the same as the MAC address of the referenced physical Ethernet controller.
- Two or more virtual Ethernet controllers use the same custom MAC address and reference the same physical Ethernet controller.

If you use a MicroAutoBox III, it is recommended to specify different custom MAC addresses for Virtual Ethernet Setup function blocks that reference the same physical Ethernet controller.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy In the Conflicts viewer, for example, specify a different custom MAC address.

Function block: Duplicate name

The active ConfigurationDesk application contains two or more function blocks with the same name.

Function block names must be unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, enter unique names in the Name edit field for all function blocks.

Function block: Duplicate network address per physical Ethernet adapter

The local IP address of two or more Ethernet setup function blocks (Ethernet Setup or Virtual Ethernet Setup function blocks) have the same subnetwork address and use the same physical Ethernet controller.

To prevent unintended behavior, it is recommended that two Ethernet setup function blocks should not reference the same physical Ethernet controller if their local IP addresses have the same subnetwork address.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy In the Conflicts viewer, for example, specify a local IP address that is a member of a different subnetwork. The subnet mask/subnet prefix length indicates how many most significant bits of the local IP address are used to specify the subnetwork address.

Function block: Duplicate or invalid local port number of assigned Ethernet clients

Using a function block that is assigned to an Ethernet setup function block (Ethernet Setup or Virtual Ethernet Setup function block), the conflict occurs in the following cases:

- Case 1: At least two function blocks are assigned to the same Ethernet setup function block and use a local port with the same port number, but they differ neither in the used transport protocol nor in the data flow direction.

  To use the same port number, you must use different transport protocols or different data flow directions. For example, the TCP, UDP Transmit and UDP Receive function blocks can use the same local port number, because they differ in the used transport protocol or in the data flow direction.

- Case 2: The function block is set to the default local port. The default local port is a placeholder with no valid port number.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives depending on the case that causes the conflict:

- Case 1: Choose one of the following alternatives:
  - In the Conflicts Viewer, for example, set the local port number of an affected function block to an unused port in the range 1 ... 65535.
  - In the Properties Browser, for example, assign the function blocks to different Ethernet setup function blocks.
- Case 2: In the Conflicts Viewer, for example, specify a local port number in the range 1 ... 65535.

Function block: Duplicate settings of assigned ECU Interfaces

The function block is referenced by two or more ECU Interface Configuration function blocks that have identical settings for the accessors grouped by <functional position>.

The settings for the accessors grouped by <functional position> must be different for ECU Interface Configuration function blocks that reference the same Ethernet Setup function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, reassign the reference between the conflicting function block and the related ECU Interface Configuration function blocks so that each ECU Interface Configuration function block references a separate Ethernet Setup function block.
- Delete one of the related ECU Interface Configuration function blocks from the ConfigurationDesk application.

  Correct the duplicate settings via the dSPACE ECU Interface Manager. Export a new EIC file and import this file to a new ECU Interface Configuration function block. Then assign the required Ethernet Setup function block to it.

Function block: Duplicate VLAN ID

VLANs with the same VLAN ID are specified for the Ethernet switch.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify VLANs with different VLAN IDs for the Ethernet switch.

Function block: Exceeding measurement interval

The values specified at the Measurement interval and/or the Recursion factor properties exceed the maximum allowed value.

Both the values specified at the Measurement interval and the Recursion factor must be smaller than or equal to twice the number of teeth of the crankshaft wheel specified in the imported wavetable file.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify values for the Measurement interval and/or the Recursion factor properties that match the maximum allowed value.

Function block: Interfering reaction activation (application termination vs. board restart)

The Reboot board and Termination of application options of the Response activator property are both selected.

The system supports only one of these options at a time.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	–	–	–

Remedy In the Conflicts Viewer, for example, clear either the Reboot board setting or the Termination of application setting.

Function block: Invalid assigned cylinder

One or more properties (Initial value, Stop value, or Initial substitute value) of the related function port contain a value that exceeds the value of the Number of cylinders property of the assigned Engine Control Setup function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a value for the related properties that does not exceed the value of the Number of cylinders property of the assigned Engine Control Setup function block.

Function block: Invalid baud rate

The specified value for the Baud rate property exceeds the value range of the selected transceiver type and/or the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a Baud rate that matches the value range of the selected transceiver type and the assigned hardware resource.

  If the function block is mapped to a Configuration Port block that is created for an RTICANMM ControllerSetup block, you cannot specify the baud rate in the Conflicts Viewer. In this case, adjust the baud rate in the related Simulink model and analyze the model, including task information, in ConfigurationDesk.
- In the Properties Browser, select a Transceiver type and/or assign a hardware resource that supports the specified baud rate.

Function block: Invalid baud rate

The specified value for the Baud rate property exceeds the value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a Baud rate that matches the value range of the assigned hardware resource.
- In the Properties Browser, assign a hardware resource that supports the specified baud rate.

Function block: Invalid channel pair

The Lambda Pump and Lambda Nernst channel requests are assigned to channels with a different channel number.

Both channel requests must be assigned to channels with the same channel number.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reassign the related channels to make the channel numbers match.

Function block: Invalid channel response configuration

At least one value, that is specified in the Challenge response pair list is invalid.

The following rules apply:

- In the pair list, all values for the challenges must be different. Double values are not allowed.
- In the pair list, all values for the responses must be different. Double values are not allowed.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify challenge and/or response values that fulfill the rules above.

Function block: Invalid chip select configuration

One or more Chip Select signal ports are not available to address an SPI slave for data transmission.

The number of provided Chip Select signal ports is specified by the Number of chip selects property.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, choose one of the following alternatives:

- Increase the number of chip select signals via the Number of chip selects property.
- Select Chip Select signals ports according to the specified number of chip select signals via the Chip select configuration property.

Function block: Invalid clock pause range

The specified value for the Clock pause property is outside the valid range. The valid range is:

Clock pause = 1.5 / [Clock frequency] s ... 1.0 s

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a clock pause that matches the valid range and the value required to match the configuration of the connected encoder. Refer to the data sheet of the encoder.

Function block: Invalid counter range

The specified counter settings are not plausible. One of the following requirements are not fulfilled:

- The specified values for counter start, end, and increment must be inside the value range of the counter.

  The value range of the counter is defined by the counter width in the imported application-specific protocol.
- The direction of counting must match the start and end values of the counter.

  If the counter increments, the counter start value must be less than or equals the end value. If the counter decrement, the start value must be greater than or equals the end value.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Counter start, Counter end, and Counter increment properties according to the requirements described above.
- In the application-specific protocol, increase the counter width and import the protocol file again.

Function block: Invalid CRC calculation mode

The Serial message mode property setting is not compatible with the CRC calculation property setting.

The Enhanced Serial Message (4Bit ID, 16Bit Data) and Enhanced Serial Message (8Bit ID, 12Bit Data) modes do not support Automatic (Legacy Mode) CRC calculation mode.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives to make the properties match:

- In the Conflicts Viewer, change the CRC calculation property to Custom or Automatic.
- Change the Serial message mode property to Short Serial message or to Disabled.

Function block: Invalid CRC ranges

The specified value for the at least one of the following properties is outside the valid range: CRC sum bit width, CRC sum start bit position, CRC-checked data bit width, and CRC-checked data start bit position.

- Valid range for CRC sum bit width: 1 ... [Transmission length]
- Valid range for CRC sum start bit position: 1 ... 1 + [Transmission length] - [CRC sum bit width]
- Valid range for CRC-checked data bit width: 1 ... [Transmission length] - [CRC sum bit width] bits
- Valid range for CRC-checked data start bit position: 1 ... 1 + [Transmission length] - [CRC-checked data bit width]

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify values for the affected properties that match the valid ranges and the value required to match the configuration of the connected encoder. Refer to the data sheet of the encoder.
- In the Conflicts Viewer, for example, set the CRC configuration property to Disabled.
- In the Conflicts Viewer, for example, increase the value of the CRC sum bit width property with respect to the value required to match the configuration of the connected encoder. Refer to the data sheet of the encoder.

Function block: Invalid CRC values

The specified value for the CRC polynomial property and/or the CRC initial value property is outside the valid range.

- Valid range for the CRC polynomial property: $2^{[\text{CRC sum bit width}]} + 1 \ldots 2^{[\text{CRC sum bit width}] + 1} - 1$
- Valid range for the CRC initial value property: $0 \ldots 2^{[\text{CRC sum bit width}]} - 1$

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify values for the affected properties that match the valid ranges and the value required to match the configuration of the connected encoder. Refer to the data sheet of the encoder.
- In the Conflicts Viewer, for example, set the CRC configuration property to Disabled.
- In the Conflicts Viewer, for example, increase the value of the CRC sum bit width property with respect to the value required to match the configuration of the connected encoder. Refer to the data sheet of the encoder.

Function block: Invalid custom idle value

The specified Custom idle value exceeds the upper limit of the wavetable value.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reduce the specified value for the Custom idle value property with respect to the upper limit, displayed by the Number of values property.

Function block: Invalid cylinder

The value specified for the Cylinder property or at least one value specified for the Cylinder map property exceeds the number of cylinders defined in the Engine Simulation Setup function block (Number of cylinders property) that is referenced by this function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reduce the specified value for the Cylinder or the Cylinder map property with respect to the number of cylinders specified in the assigned Engine Simulation Setup function block.

Function block: Invalid data phase baud rate

The specified value for the Data phase baud rate property exceeds the value range of the selected transceiver type and/or the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a Data phase baud rate that matches the value range of the selected transceiver type and the assigned hardware rescource.

  If the function block is mapped to a Configuration Port block that is created for an RTICANMM ControllerSetup block, you cannot specify the baud rate in the Conflicts Viewer. In this case, adjust the baud rate in the related Simulink model and analyze the model, including task information, in ConfigurationDesk.
- In the Properties Browser, select a Transceiver type and/or assign a hardware resource that supports the specified data phase baud rate.

Function block: Invalid destination port number

The default destination port number is a placeholder and no valid port number.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a destination port number in the range 1 ... 65535.

Function block: Invalid digital filter

A digital filter that is assigned to a function block is missing in the imported digital filters list, for example, because it has been removed after the assignment.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- From the Conflicts Viewer, for example, reimport the digital filter via the Imported Digital Filters property.
- In the Conflicts Viewer, for example, assign a digital filter which already has been imported to the electrical interface of the function block via Assigned digital filter property.

Function block: Invalid encoder distance configuration

At least one of the following cases occurs:

- The specified value of the Encoder minimum position position is greater than or equals the specified value of the Encoder maximum position property.

  The value of the Encoder minimum position must be less than the value of the Encoder maximum position property.
- The specified value of the Index position property is out of range for valid encoder positions.

  The Encoder minimum position and the Encoder maximum position properties define the range for valid encoder positions.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify valid values for the Encoder minimum position, Encoder maximum position, and the Index position properties. Use the error description above to decide on the appropriate settings.

Function block: Invalid engine direction value

The Initial value property of the Engine Direction function port is set to Reverse but the reverse crank mode is disabled.

The Reverse setting for the initial value is supported only if the Reverse Crank property is set to Enabled.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, select Undefined or Forward at the Initial value property of the function port.
- If the assigned hardware resource supports reverse rotation, set the Reverse Crank property to Enabled in the Properties Browser.

Function block: Invalid event down-sampling

The value specified for the Event downsampling property is higher than the value specified for the Maximum message count property.

The event downsampling value must be equal or lower than the maximum message count value.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reduce the value for the Event downsampling property.

Function block: Invalid event period angle

The value of one or more Period angle properties does not match the angle range of the assigned master APU provider.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify values for the Period angle properties that match the angle range of the assigned master APU provider.

Function block: Invalid event window(s)

This conflict is generated in the following cases:

- The specified value for the Window end angle is greater than the value for the Window start angle.
- The difference between the Window start angle and the Windows end angle exceeds 720°.
- The Number of event windows is set to 2 and:
  - The angle range covered by both event windows exceeds 720°.

    or
  - The specified values for Window start angle and Window end angle cause the event windows to overlap.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, specify valid values for the Windows start angle property and/or Windows end angle property. Use the error description in the Value column to decide on the appropriate settings.

Function block: Invalid feature configuration

At least one of the following cases occurs:

- The assigned Ethernet controller does not provide the settings specified for Address resolution method.
- The assigned Ethernet controller does not provide the settings for the suppression of Internet group management protocol (IGMP) messages.
- A default gateway for an IPv6 network is enabled.

  Default gateways are supported only for IPv4 networks.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives in the Conflicts Viewer, for example:

- Set the Optimization and/or Address resolution method properties to settings that match the features of the assigned Ethernet adapter.
- Set the Internet protocol property to a version that matches the specified features.
- Set the IGMP suppression property to a setting that matches the requested settings.
- Disable the default gateway.

Function block: Invalid flag positions

The specified position value for the at least one of the following properties is outside the valid range: Error flag position, Inverted error flag position, Warning flag position, and Inverted warning flag position.

Valid range for all listed properties: 0 ... [Transmission length]

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a position value for the affected properties that match the valid range and the value required to match the configuration of the connected encoder. Refer to the data sheet of the encoder.

Function block: Invalid interface type

The specified setting for the Interface type property is not supported by the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, select a setting for the Interface type property that is supported by the assigned hardware resource.
- Assign a hardware resource that supports the required interface type. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Invalid IP address

The specified IP address is not supported. The conflict occurs in the following cases:

- Case 1: The function block uses the default IP address. The default IP address is a placeholder and no valid IP address for network communication.
- Case 2: The Ethernet Setup or Virtual Ethernet Setup function block uses one of the following IP addresses as the local IP address:
  - Multicast address (IPv4: 224.0.0.0 ... 239.255.255.255, IPv6: ff00::/8)
  - Loopback address (IPv4: 127.0.0.0 ... 127.255.255.255, IPv6: ::1)
  - IPv4: IP address with 0 or 240 ... 255 as the first octet (0.0.0.0 ... 0.255.255.255 or 240.0.0.0 ... 255.255.255.255)
  - IPv6: Link-local address (fe80::/10)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Specify a supported IP address. The following IP address ranges are recommended.

- IPv4: 10.0.0.1 ... 10.255.255.254, 172.16.0.1 ... 172.31.255.254, and 192.168.0.1 ... 192.168.255.254
- IPv6: fc00::1 ... fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff (fc00::/7)

Function block: Invalid low power mode

The specified setting for the Low-power mode property is not supported by the selected transceiver type.

Only the ISO 11898-2 High-Speed CAN and the ISO 11898-3 Fault-Tolerant CAN transceiver types support the low-power mode.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Low power mode property to Disabled.
- In the Conflicts Viewer, for example, select a Transceiver type that supports the low power mode.

Function block: Invalid MAC address

The specified MAC address is no valid MAC address for an Ethernet controller. The conflict occurs in the following cases:

- The function block uses the default MAC address. The default MAC address is a placeholder and no valid MAC address for an Ethernet controller.
- The specified MAC address is a multicast or broadcast address. A valid MAC address of an Ethernet controller must be a unicast address.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy In the Conflicts viewer, for example, specify a unicast MAC address that differs from the default MAC address. For more information on the types of

MAC addresses, refer to Configuring the Basic Functionality (Virtual Ethernet Setup) (ConfigurationDesk I/O Function Implementation Guide 🕮).

Function block: Invalid master APU provider assigned

A custom function block (such as the FPGA custom function block), is assigned to a master APU provider on the same hardware resource to which the function block is assigned to.

A master APU provider cannot be connected and used from a slave APU that is located on the same board, for example, the same DS2655 FPGA Base Board.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, assign a master APU provider from a different hardware resource to the slave APU request.

Function block: Invalid maximum trigger angle vector size

The assigned hardware resource does not support the specified maximum vector size for trigger angles. The hardware resource supports 8 trigger angles for the 360° angle range.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify the maximum vector size for trigger angles to 8 or less.
- Assign a hardware resource to the function block that supports the specified vector size for trigger angles. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Invalid maximum vector size

The value specified for the Maximum vector size at the function block is outside the possible value range.

The value range depends on the assigned hardware.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a value from inside the range for the Maximum vector size property.

Function block: Invalid model connection for unmappable port

A function port for which Model access is set to Disabled is mapped to a model port.

ConfigurationDesk does not support mappings of this type.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- In a working view, remove the invalid mapping.
- In the Properties Browser, for example, set the Model access property of the involved function port to Enabled.

Function block: Invalid multiturn resolution range

The specified value for the Multiturn bit width property and/or the Multiturn start bit position property is outside the valid range.

- Valid range for the Multiturn bit width property: 0 ... [Transmission length]
- Valid range for the Multiturn start bit position property: 1 ... 1 + [Transmission length] - [Multiturn bit width]

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify values for the affected properties that match the valid ranges and the value required to match the configuration of the connected encoder. Refer to the data sheet of the encoder.

Function block: Invalid number of edges

This conflict is generated in the following cases:

- The Initial value at a Count function port exceeds the Number of edges value at the function block.

- The Number of edges exceeds the maximum value supported by the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives or both to make the properties match:

- In the Conflicts Viewer, for example, decrease the Initial value property.
- In the Conflicts Viewer, for example, decrease the Number of edges property.

Function block: Invalid number of referencing engine setups

Several Engine Simulation Setup function blocks reference the same Angular Clock Setup function block.

Only a single referencing Engine Simulation Setup block is allowed.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, configure the Assigned master APU provider property so that only a single Engine Simulation Setup function block references the affected Angular Clock Setup function block.

Function block: Invalid number of steps

The specified value for the Number of steps property is outside the valid range.

Valid settings are:

- 0: The number of steps is derived from the value specified at the Single-turn bit width property.
- $2^{[\text{Single-turn bit width}]}/(2^{22}- 1) \ldots 2^{[\text{Single-turn bit width}]}$

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a value with respect to the valid range and the value required to match the value stored in the encoders memory. Refer to the data sheet of the encoder.

If the specified values for the Single-turn bit width and the Number of steps properties match to the data sheet of the encoder and the conflict still is displayed, the function block does not support this encoder type.

Function block: Invalid optimization

The set optimization mode is not supported by the assigned Ethernet controller.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Optimization property to a supported optimization mode.
- In the Properties Browser, for example, assign a different Ethernet controller to the function block.

Function block: Invalid phase shift angle values

The value specified at the Minimum phase shift angle property is equal to or greater than the value specified at the Maximum phase shift angle property.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a value at the Minimum phase shift angle property that is smaller than the value of the Maximum phase shift angle property.

Function block: Invalid port settings

The value specified for one or more of the Initial value, Stop value or Initial substitute value properties exceeds the saturation range (system saturation values or user-configured saturation values) of the function port.

For basics on using initial and stop values, refer to Specifying Initialization and Stop Behavior (ConfigurationDesk I/O Function Implementation Guide 🕮).

For further details on using substitute values, refer to Configuring Test Automation Support (ConfigurationDesk I/O Function Implementation Guide 🕮).

For basics on user saturation, refer to Specifying User Saturation (ConfigurationDesk I/O Function Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Initial value, Stop value or Initial substitute value property within the possible value range.
- If you have specified your own min/max. values for the saturation range, reduce the Saturation minimum value or increase the Saturation maximum value.

Function block: Invalid port settings (active wavetable)

The settings of at least one property of the Active Wavetable function port (for example, Initial value or Stop value) exceeds the number of wavetable files imported to the function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a value for the related properties that matches the value range of the Imported wavetable files property.

Function block: Invalid port settings (Encoder Speed / Encoder Position)

The settings of at least one property of the Encoder Speed function port or the Encoder Position function port (for example, Initial value or Stop value), is outside the possible value range.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, change the specified value for the related properties with respect to the possible value range.

Function block: Invalid port settings (Position (Linear)/Update Position (Linear))

The values specified for one or more of the Initial value or Initial substitute value properties of the related function port are out of range for valid encoder positions.

The Encoder minimum position and the Encoder maximum position properties define the range for valid encoder positions.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Initial value or Initial substitute value property within the possible value range.
- In the Properties Browser, adapt the range for valid encoder positions with the Encoder minimum position and the Encoder maximum position properties.

Function block: Invalid protocol settings

The specified value for at least one property that defines a particular protocol setting conflicts with the setting of another property. At least one of the following cases occurs:

- A section in the protocol overlaps another section.

  The protocol contains the following sections that must not overlap: Multiturn section bit range, Single-turn section bit range, Sensor-specific diagnostic section bit range, and CRC sum section bit range. Each section covers the following bit range: [Start Bit Position] ... [Start Bit Position] + [Bit Width] - 1.
- The position of a flag collides with the position of another flag or with one of the above listed sections.

  The following positions must not collide with each other and with the above listed sections: Error flag bit position, Inverted error flag bit position, Warning flag bit position, and Inverted warning flag bit position.
- The Multiturn section bit range does not precede the Single-turn section bit range.

  If the Code type property is set to Grey code, the multiturn data must precede the single-turn data. Therefore, the value for the Single-turn start bit position must be higher than the value for the Multiturn start bit position.
- The CRC-checked data does not precede the CRC checksum.

  The CRC-checked data always must precede the CRC checksum. Therefore, the value for the CRC sum start bit position must be higher than the value for the CRC-checked data start bit position.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify values for the affected properties that match the valid ranges and the value required to match the configuration of the connected encoder. Refer to the data sheet of the encoder.

Function block: Invalid rated current

The function block sets the trigger level of the electronic fuse to an unsupported level.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Set the trigger level of the electronic fuse to a higher level via the Rated current (factor) property.
- Disable the control of the electronic fuse via the Configurable fuse property.

Function block: Invalid resource provider assigned

The Configuration port of a FlexRay function block is mapped to a Configuration Port block that was created for a FLEXRAYCONFIG UPDATE block of a Simulink model. One or more channel sets are assigned to the FlexRay function block that do not meet the requirements provided by the Configuration Port block.

The FLEXRAYCONFIG UPDATE block provides requirements regarding the simulation platform (i.e., SCALEXIO or MicroAutoBox III) and the channel type. All channel sets that are assigned to a FlexRay function block must meet these requirements.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, assign channel sets to the FlexRay function block that meet the requirements.
- Map the Configuration Port block to another FlexRay function block.

- Change the hardware requirements for the FLEXRAYCONFIG UPDATE block by specifying a different hardware configuration in the FlexRay Configuration Tool.

Function block: Invalid rotary resolution

The specified value for the Rotary resolution property exceeds the maximum value of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Specify a resolution that matches the value range of the assigned hardware resource.
- Assign a hardware resource that supports the specified resolution. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Invalid sample trigger configuration

The Sample trigger source property is set to Time (sequence-aligned), Trigger function, or to Angle and at least one of the following cases occurs:

- The Digital filter property is set to Enabled.
- The Trigger shift property is not set to 0. This means trigger shifting is enabled.
- The Sequence trigger source property is set to Threshold or Digital Threshold.

Only the Time (free-running) sample trigger source supports digital filtering and the trigger shift functionality.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify valid values for the Sample trigger source, Sequence trigger source, Digital filter and the Trigger shift properties. Use the error description above to decide on the appropriate settings.

Function block: Invalid saturation range

The specified value for the Saturation minimum value property exceeds the value for the Saturation maximum value property.

For basics on user saturation, refer to Specifying User Saturation (ConfigurationDesk I/O Function Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reduce the Saturation minimum value or increase the Saturation maximum value.

Function block: Invalid sensor-specific diagnostic range

The specified value for the Sensor-specific diagnostic bit width property and/or the Sensor-specific diagnostic start bit position property is outside the valid range.

- Valid range for the Sensor-specific diagnostic bit width property: 0 ... [Transmission length]
- Valid range for the Sensor-specific diagnostic start bit position property: 1 ... 1 + [Transmission length] - [Sensor-specific diagnostic bit width]

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify values for the affected properties that match the valid ranges and the value required to match the configuration of the connected encoder. Refer to the data sheet of the encoder.

Function block: Invalid sequence length of 0 for sequence trigger source = 'Sequence end'

The Sequence trigger source property is set to Sequence end and the Sequence length property is set to 0 (= infinite number of samples). In this case, the function block ignores the zero value and internally uses the specified value of the Maximum sequence length property instead.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Sequence trigger source property that supports sequence capture with an infinite sequence length, for example, Model task or Threshold.

- In the Conflicts Viewer, for example, specify a nonzero value for the Sequence length property (= infinite number of samples).

Function block: Invalid settings for multicast/broadcast

The function block receives broadcast messages although the reception of broadcast messages is disabled.

If you enable the reception of multicast messages, the reception of broadcast messages is automatically enabled.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy In the Conflicts Viewer, for example, choose one of the following alternatives:

- Set the Receive from broadcast address property to Enabled.

  To make sure that only unicast and multicast messages are taken into account, filter out potential broadcast messages in the behavior model, for example, by reading only messages with known source IP addresses and ports.
- If you do not need to receive multicast messages, set the Receive from multicast address property to Disabled.

Function block: Invalid signal amplitude

The value specified for the Signal amplitude property is outside the valid range.

The maximum value depends on the setting of the Measurement range property (High or Low), the value specified for the Signal offset property, and the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose at least one of the following alternatives:

- In the Conflicts Viewer, for example, reduce the specified value for the Signal amplitude property with respect to the specified measurement range, the specified signal amplitude, the assigned hardware resource, and the value required to match the signal amplitude of the connected encoder. Refer to the data sheet of the encoder.
- In the Conflicts Viewer, for example, change the setting of the Measurement range property from Low to High.
- In the Conflicts Viewer, for example, specify a value for the Signal offset property that matches the specified signal amplitude.

- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Invalid signal edge positions

The distance between an angle position for the rising and/or falling edge of the signals is too small. The distance between edge positions must be at least 1°.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify angle positions at the Signal edge position vector property that respect the minimum distance of at least 1° between edge positions. You have to consider both rising and falling edges.

Function block: Invalid signal offset

The value specified for the Signal offset property is outside the valid range.

The valid value range depends on the setting of the Measurement range property (High or Low), the value specified for the Signal amplitude property, and the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose at least one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Signal offset property with respect to the specified measurement range, the specified signal amplitude, the assigned hardware resource, and the value required to match the signal offset of the connected encoder. Refer to the data sheet of the encoder.
- In the Conflicts Viewer, for example, change the setting of the Measurement range property.
- In the Conflicts Viewer, for example, specify a value for the Signal amplitude property that matches the specified signal offset.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Invalid single-turn resolution range

The specified value for the Single-turn bit width property and/or the Single-turn start bit position property is outside the valid range.

- Valid range for the Single-turn bit width property: 0 ... [Transmission length]
- Valid range for the Single-turn start bit position property: 1 ... 1 + [Transmission length] - [Single-turn bit width]

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify values for the affected properties that match the valid ranges and the value required to match the configuration of the connected encoder. Refer to the data sheet of the encoder.

Function block: Invalid size of the capture values and/or the timestamps (size exceeds the recommended maximum)

The product of the Sequence length and Maximum sequence length property settings exceeds 655350. The product must be equal to or less than 655350.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reduce the values for the Sequence length property and/or the Maximum sequence length property.

Function block: Invalid termination

The setting of the Termination property does not match the selected transceiver type and/or the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, change the setting of the Termination property to match the selected transceiver type and assigned hardware resource.

- In the Properties Browser, select a Transceiver type and/or assign a hardware resource that supports the specified termination.

Function block: Invalid threshold values

The value specified at the Lower threshold property is equal to or greater than the value specified at the Upper threshold property.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a value at the Lower threshold property that is smaller than the value of the Upper threshold property.

Function block: Invalid time source

The assigned hardware resource does not provide a local time source.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Time Source property to Global.
- Assign a hardware resource that provides a local time source. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Invalid time vector entry

A value specified at the Time Vector function port is not supported by the assigned hardware resource.

The value range for specific properties at the Time Vector function port, for example, the Saturation minimum value property (if user saturation is used), changes depending on the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the affected property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Invalid time synchronization domain number

The specified time domain does not match the requirements of the selected PTP type. The IEEE 802.AS protocol supports only a single time domain with the number 0.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, set the time domain number of the affected function block to 0.

Function block: Invalid timeout value

The specified timeout value is shorter than the synchronization pulse of the SENT messages or too long to be measured.

The value range is calculated as follows:

- The minimum value depends on the assigned real-time hardware:
  - SCALEXIO:

    T_{min} = Sync pulse length · (1 - Tick period tolerance/100)
  - MicroAutoBox III:

    T_{min} = Sync pulse length · (1 + Tick period tolerance/100)

  Sync pulse length = (Low pulse ticks + Synchronization pulse high ticks) · Nominal tick period
- T_{max} = 32767 · Nominal tick period · (1 - Tick period tolerance/100)

  The absolute maximum value depends on the assigned real-time hardware.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, set the Timeout, Synchronization pulse high ticks, Low pulse ticks, Nominal tick period, and Tick period tolerance properties according to the requirements on the timeout value range described above.

Function block: Invalid timeout limit (compared to challenge period)

The value specified for the Timeout limit property is equal or greater than the value specified for the Challenge period property.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a value for the Timeout limit property that is less than the value specified for the Challenge period property.

Function block: Invalid tooth period ratio

The value specified at the Tooth period ratio property exceeds the value range of the property.

The value range depends on the crankshaft wheel and is specified in the imported wavetable file.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a value for the Tooth period ratio property that matches the value range specified in the imported wavetable file.

Function block: Invalid traffic shaping configuration

The sum of the reserved bandwidths of traffic class A and traffic class B exceeds the maximum data rate of the Ethernet switch port. The sum of the reserved bandwidths must not exceed 1 Gbit/s.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, decrease the reserved bandwidths for the traffic classes.

Function block: Invalid trigger angle vector size

A property value specified at the Trigger Angle Vector Size function port (for example, Initial value) exceeds the Maximum trigger angle vector size specified at the function block.

The Maximum trigger angle vector size is used only when the Function trigger property is set to Angle based at the function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, change the specified value for the related properties with respect to the defined maximum value.

Function block: Invalid trigger condition

The setting of the Trigger condition property is not supported by the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, select a setting for the Trigger condition property that is supported by the assigned hardware resource.
- Assign a hardware resource that supports the selected trigger condition. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Invalid trigger configuration

At least one of the following cases occurs:

- The Sequence length property is set to 0 (= infinite) and the Trigger shift property is not set to 0.
- The Sequence length property is set to 0 (= infinite) and the Trigger mode property is set to Ignore.

  In this case the function block ignores the setting and internally uses the Immediate setting instead.
- The Trigger shift property is set to < 0 and the Trigger mode property is set to Immediate.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy In the Conflicts Viewer, for example, specify valid values for the Sequence length and the Trigger shift or Trigger mode properties. Use the error description above to decide on the appropriate settings.

Function block: Invalid trigger function provider assigned (not on same board)

The channel set assigned to the function block and the channel set assigned to the referencing trigger function provider (for example, the Trigger In function block) are located on different I/O boards.

The hardware resources assigned to the function block and the referencing trigger function provider must be located on the same I/O board.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Assign hardware resources from the same I/O board to the affected function blocks. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).
- Reassign the trigger function provider (for example, a Trigger In function block), so that all assigned hardware resources are located on the same I/O board.

Function block: Invalid trigger period angle

The value of the Trigger period angle property does not match the angle range of the assigned master APU provider.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a value at the Trigger period angle property that matches the angle range of the assigned master APU provider.

Function block: Invalid trigger source feature configuration

At least one of the following cases apply:

- The settings for the Digital filter property and the Trigger source property are not compatible.

  You can use the digital filter only in combination with the Time (free-running) trigger source.
- Digital filtering is not supported by the assigned hardware resource.
- The specified setting for the Trigger source property is not supported by the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, change the setting of one property to resolve the unsupported value setting or to match the features of the assigned hardware resource.
- Assign a hardware resource which provides the specified feature (digital filtering or the specified setting for the Trigger source property). For instructions on assigning hardware resources via the Properties Browser, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Invalid turn bit width range

The sum of the values specified for the Single-turn bit width and the Multi-turn bit width properties equals 0 (applies only for EnDat Master function blocks) or exceeds the maximum value (applies for all function block types).

A maximum value of 48 bits is supported by the function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify values for the Single-turn bit width and the Multi-turn bit width properties with respect to the supported values and the values required to match the values stored in the encoders memory. Refer to the data sheet of the encoder.

Function block: Invalid values (event generation vs. number of event window(s))

The settings for the Event generation property and the Number of event windows property are not compatible.

If Event generation is enabled, the Number of event windows must not be set to 0.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, choose one of the following alternatives to make the properties match:

- Set the Event generation property to Disabled.
- Adjust the Number of event windows property.

Function block: Invalid values (event generation vs. trigger source)

The settings for the Event generation property and the Trigger source property are not compatible. Event generation is supported only when the Trigger source property is set to Function trigger.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, choose one of the following alternatives to make the properties match:

- Set the Event generation property to Disabled.
- Set the Trigger source property to Function trigger.

Function block: Invalid values (index position vs. linear resolution)

The values of the Index position and Linear resolution properties do not match.

The value range for the Index position property depends on the value of the Linear resolution property.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, increase the value range by configuring the Linear resolution property or decrease the Index position value to be inside the given range.

Function block: Invalid Vbat range

The value specified at the Vbat lower limit property is equal to or greater than the value specified at the Vbat upper limit property.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	✓

Remedy In the Conflicts Viewer, for example, specify a value at the Vbat lower limit property that is smaller than the value of the Vbat upper limit property.

Function block: Invalid vector size

A property value specified at the Vector Size function port (for example, Initial value) exceeds the setting of the Maximum vector size property specified at the function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, change the specified value for the related properties with respect to the defined maximum value.

Function block: Mismatching angle sequences

The captured angle sequences overlap, because the following two cases occur in combination:

- The sequence trigger source and the sample trigger source are both set to Angle.
- The difference between two angle sequence trigger positions is below the minimum value.

To avoid overlaps, the angle difference between two angle sequence trigger positions must not fall below the following value:

Minimum angle position difference = (Sequence length - 1) * Angle sample distance + Minimum angle sample distance

With the minimum angle sample distance = $2 * (360° / 2^{15}) \approx 0.022°$

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Increase the angle differences at which the capturing of a sequence is triggered via the Angle sequence triggers property.
- Specify an angle position difference (via the Sequence length and Angle sample distance properties) that is higher than the required minimum value (see formula above).
- Specify a setting other than Angle for at least one trigger source (sequence trigger source or sample trigger source).

Function block: Mismatching deadline violation settings

A Deadline Violation event of the FlexRay function block is assigned to a task, and the DLV of synchronization task and/or DLV of application or communication task properties are set to Terminate.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, change the deadline violation property settings to Warn.
- In the Executable Application table, for example, delete the assignment of the Deadline Violation event from the task.

Function block: Mismatching event generation and pitch counter configuration setting

The settings for the Event generation property and the Pitch counter configuration property does not match.

If Event generation is enabled, the Pitch counter configuration property must also be enabled.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Event generation property to Disabled.
- In the Properties Browser, set the Pitch counter configuration property to Enabled.

For more information on providing I/O events, refer to Configuring the Basic Functionality (Wheelspeed Out) (ConfigurationDesk I/O Function Implementation Guide 🕮).

Function block: Mismatching feedthrough and termination

In a CTR<n> function group of a FlexRay function block, a Feedthrough Chx value does not match the related Termination Chx value.

Feedthrough is set to True for a channel that is set to Terminated.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, change the value for Feedthrough Chx to False or the value for Termination Chx to Not Terminated.

Function block: Mismatching feedthrough mode and transceiver type

The selected feedthrough mode is not compatible with the selected transceiver type.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives to make the properties match:

- In the Conflicts Viewer, for example, change the Feedthrough mode property.
- In the Conflicts Viewer, for example, change the Transceiver type property.

Function block: Mismatching IGMP suppression setting

A function block requests the suppression of Internet group management protocol (IGMP) messages, but the referenced Ethernet Setup function block enables the sending of IGMP messages or vice versa.

Note: The conflict causes ConfigurationDesk to generate default code during the build process. For the default code, ConfigurationDesk uses the following setting:

- If all referencing function blocks request the same setting, ConfigurationDesk uses the requested setting.
- If the referencing function blocks request different settings, ConfigurationDesk uses the setting of the Ethernet Setup function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy In the Conflicts Viewer, for example, set the IGMP suppression property to a setting that matches the requested settings.

Function block: Mismatching internet protocol version

The specified Internet protocol version does not match to the version that you specified for the assigned Ethernet setup function block (Ethernet Setup or Virtual Ethernet Setup function block).

All members of an Ethernet network must use the same Internet Protocol version, either IPv4 or IPv6.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, change the specified value for the Internet protocol property to the commonly used protocol version.

Function block: Mismatching load descriptions

There is a mismatch between the (string-based) load description of a signal port and the (string-based) load description of the assigned hardware channel.

The load description is used for load compare checks, if the Compare mode property is set to Load description is compared by system. For details, refer to Specifying Load Settings (ConfigurationDesk I/O Function Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Choose one of the following alternatives to make the properties match:

- Change the Compare mode property to Load description is compared by user. In this case the load compare check by the system is disabled.
- In the Conflicts Viewer, enter string-based descriptions for the Load description property (at the corresponding signal port) and for the Load description property for the assigned hardware channel that are identical.

Function block: Mismatching load rejection settings

The specified settings for the Load rejection properties do not match for the corresponding signal. You can, for example, enforce load rejection at different positions in the signal path. For basics, refer to Basics on Load Rejection (ConfigurationDesk Real-Time Implementation Guide 🕮).

The conflict occurs in the following cases:

- Load rejection is not enforced by a signal port (of a function block), but by the assigned hardware.
- Load rejection is not enforced by a signal port (of a function block), but by the external load's device port mapped to the load signal port (of a function block).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy In the Conflicts Viewer, for example, change the settings for the Load rejection properties for one signal so that they are identically at all positions in the signal path.

Function block: Mismatching maximum current

This conflict applies to the Injection/Ignition Current In function block if you use extended signal analysis.

The number of channel requests for the Digital Interface does not equal the number of channel requests for the Analog interface, because you have specified mismatching values at the Required current property for both interfaces.

A single hardware channel is able to provide a certain voltage and current range depending on the channel type. ConfigurationDesk uses the specified Required current value to determine the number of hardware channels needed. For further details on channel multiplication refer to Specifying Current and Voltage Values for Channel Multiplication (ConfigurationDesk I/O Function Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify matching values for the Required current property of the Digital Interface and the Analog Interface of the function block.

Function block: Mismatching measurement range configuration

The setting of the Measurement range property of at least one Sine Encoder In function block differs from the setting of the Measurement range property of another Sine Encoder In function block.

All Sine Encoder In function blocks which use the Flexible In/Out 1 channel type from the same I/O board should have the same setting for the Measurement range property. The assigned I/O board supports only a common setting. In case of different settings, ConfigurationDesk uses the high measurement range as a common default for all affected Sine Encoder In function blocks.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, change the setting of the Measurement range property of the affected function blocks, so that all function blocks assigned to the same I/O board have the same setting (High or Low).
- Assign a different channel type from the related I/O board or a hardware resources from another I/O board to the function blocks, which then are configured to a different measurement range.

Function block: Mismatching network address

The specified IP address is not a member of the subnetwork you specified for the used Ethernet controller.

For example: You specified the subnet mask 255.255.255.0 and the local IP address 192.168.1.10 to define the subnetwork. Then, you specify the default gateway 192.168.2.2. In this case, the default gateway and the local IP address are not members of the same subnetwork. Therefore, IP packets cannot be sent to the default gateway.

The subnet mask of the Ethernet controller indicates how many most significant bits of the IP address are used to specify the subnetwork address. The rest of the IP address is the range to indicate members of the same subnetwork. In the example, valid IP addresses for the default gateway are in the range of 192.168.1.0 ... 192.168.1.255, because the resulting subnetwork address of the Ethernet controller is 192.168.1.x .

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify IP addresses that use the same subnetwork address.
- In the Properties Browser, specify a subnet mask for the used Ethernet controller that results in a subnetwork address that matches to all IP addresses.

Use the error description above to decide on the appropriate setting.

Function block: Mismatching optimization

A function block requests a certain optimization mode of the physical Ethernet controller, but the referenced Ethernet Setup function block specifies a different mode.

Note: The conflict causes ConfigurationDesk to generate default code during the build process. For the default code, ConfigurationDesk uses the following setting:

- If all referencing function blocks request the same optimization mode, ConfigurationDesk uses the requested optimization mode.
- If the referencing function blocks request different optimization modes, ConfigurationDesk uses the setting of the Ethernet Setup function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy In the Conflicts Viewer, for example, set the Optimization property to a setting that matches the requested optimization type.

Function block: Mismatching partial networking mode and transceiver type

The selected partial networking mode is not compatible with the selected transceiver type.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, change the Partial networking mode property.
- In the Conflicts Viewer, for example, change the Transceiver type property.

Function block: Mismatching property values regarding CAN bus access request

A CAN function block is referenced in one or more ECU Interface Configuration function blocks. At least one of the following settings differs for

the CAN function block and the EIC files that are imported to the ECU Interface Configuration function blocks:

- Baud rate
- Sample point
- CAN FD support
- Data phase baud rate
- Data phase sample point

If an EIC file specifies one or more of these settings, it might be useful to adapt the related properties of the CAN function block accordingly. Whether adapting the related properties is useful depends on your use scenario.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify valid values for the Baud rate, Data phase baud rate, Sample point, Data phase sample point, and/or CAN FD mode properties.
- If the CAN function block is affected by multiple EIC files that specify different settings, change the Referencing Configuration in the related ECU Interface Configuration function blocks:

  Reference this CAN function block in ECU Interface Configuration function blocks whose imported EIC files specify identical settings. Reference other CAN function blocks in the ECU Interface Configuration function blocks whose imported EIC files specify different settings.

Function block: Mismatching property values regarding CAN bus access request

A CAN function block is assigned to one or more bus access requests of one or more bus configurations. The assignment contains at least one of the following settings:

- Case 1: The baud rate and/or data phase baud rate of the CAN function block and an assigned bus access request differ.

  The baud rate and the data phase baud rate of a CAN function block and all its assigned bus access requests must match.
- Case 2: A bus access request requires CAN FD support but the CAN FD support is disabled for the CAN function block.

  If a bus access request requires CAN FD support, the CAN FD support must be enabled for the CAN function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives, depending on the case that causes the conflict:

- Case 1: Depending on the CAN function block assignments, resolve the mismatching baud rates as follows:
  - The CAN function block is assigned to bus access requests with identical baud rates: In the Conflicts Viewer, for example, adapt the conflicting Baud rate and/or Data phase baud rate property of the CAN function block to the related baud rate value of the bus access requests.
  - The CAN function block is assigned to bus access requests with different baud rates: Change the conflicting baud rates of the bus access requests or change the bus access assignment.

    To change the conflicting baud rate property of a bus access request, select the related communication cluster, for example, in the Bus Access Requests table. Then, specify a valid value for the Baud rate and/or Data phase baud rate property in the Properties Browser.

    To change the bus access assignment, select the related bus access requests in the Bus Access Requests table and change the assigned bus access via the Bus Access Assignment command, for example. Assign this function block to bus access requests with identical baud rates. Use other CAN function blocks to specify the bus access of the bus access requests with the different baud rates.
- Case 2: In the Conflicts Viewer, for example, enable the CAN FD mode:
  - Select Enabled (Non-ISO) if you work with the non-ISO CAN FD protocol. This is the original CAN FD protocol developed by Bosch.
  - Select Enabled (ISO) if you work with the ISO CAN FD protocol as specified in ISO 11898-1.

Function block: Mismatching property values regarding CAN Configuration Port

The Configuration port of a CAN function block is mapped to one or more Configuration Port blocks. For at least one of the following properties, a Configuration Port block provides different settings than specified in the CAN function block:

- Baud rate
- Data phase baud rate
- CAN FD mode

The baud rates and the CAN FD mode that are specified in the CAN function block must match the settings that are provided by Configuration Port blocks.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- If one or more Configuration Port blocks with identical settings are mapped to the CAN function block, adapt the function block settings:

  In the Conflicts Viewer, for example, specify valid values for the Baud rate, Data phase baud rate, and/or CAN FD mode properties.
- If multiple Configuration Port blocks with different settings are mapped to the CAN function block, change the mapping:

  Map the Configuration Port blocks with identical settings to this CAN function block. Map the Configuration Port blocks with different settings to other CAN function blocks.

Function block: Mismatching property values regarding LIN bus access request

A LIN function block is assigned to one or more bus access requests of one or more bus configurations. The assignment contains at least one of the following settings:

- Case 1: The baud rates of the LIN function block and an assigned bus access request differ.

  The baud rate of a LIN function block and all its assigned bus access requests must match.
- Case 2: The sum of the LIN slaves and LIN masters that are assigned to the LIN function block exceeds 64.

  The maximum number of LIN slaves of one physical LIN bus is limited. Therefore, one LIN function block can specify the bus access for up to 64 LIN slaves. Because each LIN master provides one LIN slave task, the number of assigned LIN masters affects the number of assigned LIN slaves as well.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives, depending on the case that causes the conflict:

- Case 1: Depending on the LIN function block assignments, resolve the mismatching baud rates as follows:
  - The LIN function block is assigned to bus access requests with identical baud rates: In the Conflicts Viewer, for example, adapt the Baud rate of the LIN function block to the baud rate of the bus access requests.

- The LIN function block is assigned to bus access requests with different baud rates: Change the conflicting baud rates of the bus access requests or change the bus access assignment.

  To change the baud rate of a bus access request, select the related communication cluster, for example, in the Bus Access Requests table. Then, specify a valid value for the Baud rate property in the Properties Browser.

  To change the bus access assignment, select the related bus access requests in the Bus Access Requests table and change the assigned bus access via the Bus Access Assignment command, for example. Assign this function block to bus access requests with identical baud rates. Use other LIN function blocks to specify the bus access of the bus access requests with the different baud rates.

- Case 2: Depending on the LIN function block assignments, reduce the number of LIN slaves and/or LIN masters as follows:
  - The LIN function block is assigned to one bus access request: In the Bus Configurations table, remove as many LIN slaves and/or LIN masters as necessary from the related bus configuration. Assign the removed LIN slaves and LIN masters to other bus configurations and use other LIN function blocks to specify their bus access.
  - The LIN function block is assigned to multiple bus access requests: In the Bus Access Requests table, select the related bus access requests and change the bus access assignment of as many bus access requests as necessary, for example, via the Bus Access Assignment command.

Function block: Mismatching property values regarding LIN Configuration Port

One or more Configuration ports of bus simulation containers or V-ECU implementations are mapped to a LIN function block. The baud rates of the Configuration ports and the LIN function block differ.

The baud rates of all the Configuration ports that are mapped to one LIN function block and the baud rates of the LIN function block itself must match.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- If either one Configuration port or multiple Configuration ports with identical baud rates are mapped to the LIN function block, adapt the function block's baud rate:

  In the Conflicts Viewer, for example, specify a valid value for the Baud rate property.

- If multiple Configuration ports with different baud rates are mapped to the LIN function block, change the mapping of the Configuration ports:

Map the Configuration ports with identical baud rates to this LIN function block. Map the Configuration ports with different baud rates to other LIN function blocks.

Function block: Mismatching sector signal pair

The signals specified for at least one signal pair (for example, A+ and A-) are problematic. The output levels of the related signal pairs are active at the same time. This might cause a short circuit in unprotected half bridges.

Signal settings for a signal pair that do not lead to a conflict are:

- Inactive and all other signals
- PWM and Inverted PWM
- Complement PWM and Complement inverted PWM

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Signal pair check property to Disabled. Ensure, that disabling the signal pair check will not result in damage to the connected half bridges..
- Specify only the signals for a signal pair that leads not to a problematic combination. Refer to Configuring Sector and Stationary Signals (Block-Commutated PWM Out) (ConfigurationDesk I/O Function Implementation Guide 🕮).

Function block: Mismatching values (pump cell resistance vs load description)

The descriptions of the Pump cell resistor property and the Load description property (at the corresponding signal port), and/or description of the assigned hardware channel do not match.

The load description is used for load compare checks, if the property Compare mode is set to Load description is compared by system. For details, refer to Specifying Load Settings (ConfigurationDesk I/O Function Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy In the Conflicts Viewer, for example, enter descriptions for the Load description property (at the corresponding signal port and for the assigned hardware channel, the latter via Platform Manager) that are identical to the Pump cell resistor value property value.

Function block: Mismatching wavetable sizes

This conflict is generated in the following cases:

- The angle range of the imported wavetables does not match the angle range of the referenced master APU provider.

  The function block supports two wavetable sizes:

  - Wavetables with 65536 entries that support master APUs with an angle range of 720° for one engine cycle.

    If the referenced master APU has an angle range of 360°, only the first 32768 entries are output.

  - Wavetables with 32768 entries that support master APUs with an angle range of 360° for one engine cycle.

    If the referenced master APU has an angle range of 720°, the wavetable is output twice.

  For more information, refer to Execution of angular-coded wavetables (ConfigurationDesk I/O Function Implementation Guide 🕮).

- Not all wavetables that are referenced by the same function block have the same wavetable size.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy Choose one of the following alternatives:

- Via the Imported wavetable files property, import wavetable files that support the angle range of the referenced master APU provider.
- Via the Assigned master APU provider property, assign a master APU provider that supports the wavetable size of the imported wavetable files.
- Change the wavetables so that they all have the same number of wavetable values.

Function block: Missing channel assignment

For an electrical interface unit with an assigned angle unit channel set or an assigned Ethernet adapter channel set, at least one channel request is not assigned.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, assign an angle unit or an Ethernet adapter (depending on the function block) to each channel request of the function block. For instructions on assigning them via the Properties

Browser, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Missing channel assignment

For an electrical interface unit with an assigned channel set, at least one channel request is not assigned.

ConfigurationDesk determines the types and number of channels required for a function block according to the assigned channel set, the function block features, the block configuration and the required physical ranges. The required channels are called *channel requests*.

Hardware resource (channels of channel sets) are required for most function blocks to perform the I/O functionality. For more basics, refer to Basics on Hardware Resource Assignment (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy In the Conflicts Viewer, for example, assign a channel to each channel request of the function block. For instructions on assigning them via the Properties Browser, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Missing channel set assignment

At least one electrical interface unit of the function block is not assigned to a channel set.

Hardware resource (channels of channel sets) are required for most function blocks to perform the I/O functionality. For more basics, refer to Basics on Hardware Resource Assignment (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, assign a channel set to the function block. For instructions on assigning them via the Properties Browser, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Missing or invalid digital filter

At least one of the following conditions is met:

- No digital filter file is imported via the Import digital filters property.
- No digital filter file is assigned at one or more of the Assigned digital filter A, B, C properties.
- A digital filter file that is assigned to a digital filter is missing in the list of imported digital filter files, for example, because it was removed after the assignment.

The following conditions must be met:

- At least one digital filter file must be imported.
- A digital filter file must be assigned to each of the three digital filters. All assigned digital filter files must be in the list of the imported digital filter files.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, import at least one digital filter file via the Import digital filters property and assign a digital filter file to each digital filter at the Assigned digital filter A, B, C properties.

Function block: Missing or supernumerous block connections (by constraint)

The number of connections to or from a custom function block to other function blocks (for example, the number of connections from APU provider to slave APUs) does not match a MultiplicityConstraint that was defined for the custom function block type.

For more information on custom function blocks, refer to ConfigurationDesk Custom I/O Function Implementation Guide 🕮.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Connect the correct number of function blocks to satisfy the MultiplicityConstraint.

Function block: Missing user file(s)

The source code file (`<Custom_Function_Block_Name>.cpp`) and/or header file (`<Custom_Function_Block_Name>.h`) of a custom function block used in your ConfigurationDesk application are/is missing.

Custom functions (such as the UART function) consist of an XML file, a source code file, a header file, and a type definition header file. The source code and

header files of a custom function must be located in the same custom functions directory as the XML file. A missing type definition header file (`<Custom_Function_Block_Name>_TypeDef.h`) does not cause the conflict since it would be automatically generated during the build process if it was missing.

The custom function directory is either the project-specific **`CustomFunctions`** directory or the global custom functions directory.

- For details on specifying the global custom functions directory, refer to Configuration Page (ConfigurationDesk User Interface Reference 🕮).
- For more information on custom functions in general, refer to ConfigurationDesk Custom I/O Function Implementation Guide 🕮.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, the names of the missing files are displayed in the Value column. Add the missing custom function files to the directory where the XML file is located (project-specific or global custom functions directory). Afterwards, use the Reload Custom Function Definitions command in the Function Browser.

Function block: Missing VLAN membership type

The protocol-based VLAN configuration is enabled, but no VLAN is specified for the membership tagged or untagged of the Ethernet switch port.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a VLAN for the membership tagged or untagged.

Function block: Multiple assignments to different application processes

The assignment of a function block to an application process is determined by two factors:

- The mapping to model port blocks of model implementations that are assigned to application processes.
- The assignment of I/O events to tasks that are assigned to application processes.

This conflict is generated only if these two factors together produce an assignment of a function block to more than one application process.

Note

If you use **Bus Configuration** function blocks in the ConfigurationDesk application: There are additional factors that determine the assignment to an application process. However, if a **Bus Configuration** function block is assigned to more than one application process, the 'Bus Configuration: No valid application process assigned' conflict is also generated. Resolve this conflict to assign the **Bus Configuration** function block to only one application process. Refer to Resolving the Bus Configuration Conflict: No Valid Application Process Assigned (ConfigurationDesk Bus Manager Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Ensure that a function block is assigned to only one application process according to the factors described above.

Function block: Multiple processing units connected

This conflict is generated in the following cases:

- Case 1: The function block provides two or more electrical interface units. These units are assigned to hardware resources (e.g., channels on different I/O boards) which are directly or indirectly connected to different SCALEXIO Processing Units.
- Case 2: The function block is mapped to a behavior model that is executed in a processing unit application that is assigned to SCALEXIO Processing Unit A. However, the electrical interface unit of the function block is assigned to hardware resources that are directly or indirectly connected to SCALEXIO Processing Unit B.
- Case 3: This function block A is referenced by function block B. Function block B also references function block C. The hardware resources that are assigned to function block A and C are directly or indirectly connected to different SCALEXIO Processing Units.
- Case 4: This function block A is referenced by function block B. Function block B is mapped to a behavior model. The behavior model is executed in a processing unit application that is assigned to a SCALEXIO Processing Unit. The hardware resources that are assigned to function block A are directly or indirectly connected to a different SCALEXIO Processing Unit.

A function block must have a direct or indirect hardware connection to only one SCALEXIO Processing Unit.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Choose one of the following alternatives depending on your hardware constellation:

- Assign a hardware resource with the following characteristics to the function block: The hardware resource must be directly or indirectly connected to the same SCALEXIO Processing Unit to which the function block is connected, e.g., by means of a processing unit application. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).
- Specify a connection to a SCALEXIO Processing Unit via the **Uplink** or **Downlink** properties to get a connection to the same SCALEXIO Processing Unit to which the function block is connected, e.g., by means of a hardware resource assignment. These hardware properties are available for IOCNET Routers and IOCNET Link Boards. For information on accessing the properties, refer to How to Access Hardware Properties (ConfigurationDesk Real-Time Implementation Guide 🕮). For basics on the IOCNET connections in a SCALEXIO system, refer to Network Concept (SCALEXIO Hardware Installation and Configuration 🕮).
- Only for case 2 and 4: Assign the same SCALEXIO Processing Unit to the processing unit application that is also connected to the function block by means of a hardware resource assignment. For instructions, refer to How to Assign Processing Units to Processing Unit Applications (ConfigurationDesk Real-Time Implementation Guide 🕮).
- Only for case 3 and 4: Delete the reference to this function block in the referencing function block.

Function block: Multiple triggering

The function block is triggered by more than one function module but is not multitriggerable. For more information on multiple function triggers, refer to Basics on Function Triggers (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy Make sure that the function is triggered by only one function module. For details, refer to How to Avoid Multiple Function Triggers (ConfigurationDesk Real-Time Implementation Guide 🕮).

Note: The assignment of a data port to its associated function module can be determined only during the model code generation and not during the model

analysis. If you move data port blocks in the behavior model from one subsystem to another, this action might trigger this conflict even if the reason for this conflict was eliminated and the model was re-analyzed in ConfigurationDesk. Generating the model code (either by starting the build process or by generating it explicitly) updates this information and resets the conflict.

Function block: No assignment to application process

A function block is implicitly assigned to an application process via:

- Its mapping lines to model port blocks of model implementations that are assigned to application processes.
- The assignment of its I/O events to tasks that are assigned to application processes.
- The hardware assignment: A function block must have a direct or indirect hardware connection to SCALEXIO processing hardware.

To generate code, each function block must be assigned to an application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Assign the function block to an application process. Consider the factors described above.

Function block: No central FIU hardware available in hardware topology

The hardware topology in your active ConfigurationDesk application does not contain central FIU hardware which supports the failure simulation feature.

To use failure simulation with a SCALEXIO system, it has to be equipped, for example, with a DS2680 I/O Unit or a DS2642 FIU & Power Switch Board. These boards provide a central failure insertion unit (FIU). The central FIU switches the signal from the ECU pin to simulate the failure.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Failure simulation property to Not required.
- Extend the hardware topology by adding the required central FIU hardware (a DS2680 I/O Unit or a DS2642 FIU & Power Switch Board). For instructions, refer to How to Extend Existing Hardware Topologies (ConfigurationDesk Real-Time Implementation Guide 🕮).

- Replace the hardware topology by scanning a registered SCALEXIO hardware system, which contains the required central FIU hardware (a DS2680 I/O Unit or a DS2642 FIU & Power Switch Board). For instructions, refer to How to Import a Hardware Topology (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: No EIC file imported

An ECU Interface Configuration function block is available in the active ConfigurationDesk application and no ECU interface container (EIC) file is imported to the function block.

Each ECU Interface Configuration function block that is available in a ConfigurationDesk application requires one ECU interface container (EIC) file.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Import an EIC file to the ECU Interface Configuration function block. For instructions, refer to How to Import ECU Interface Container (EIC) Files (ConfigurationDesk Real-Time Implementation Guide 🕮).
- Delete the ECU Interface Configuration function block from the ConfigurationDesk application.

Function block: No engine simulation setup assigned

The function block does not reference an Engine Simulation Setup function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy In the Conflicts Viewer, for example, select an instantiated Engine Simulation Setup function block via the Assigned simulation setup property.

Function block: No feature provider assigned

The function block does not reference the required feature provider, for example:

- A Current In or a Voltage In function block does not reference a master APU provider (such as an Angular Clock Setup function block).
- An ECU Interface Configuration function block does not reference an Ethernet Setup function block.
- A function block that use a trigger function as trigger source does not does not reference a trigger function provider.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, assign the required feature provider. Choose one of the following alternatives depending on the conflicting function block type:

- Assign a master APU provider (such as an Angular Clock Setup function block) to the conflicting Current In or Voltage In function block.
- Assign an Ethernet Setup function block to the conflicting ECU Interface Configuration function block.
- Assign a trigger function provider (such as a Trigger In function block) to the conflicting function block.

Function block: No or invalid master APU provider assigned

Using a function block requiring a master APU provider, the conflict occurs in one of the following cases:

- Case 1: An Engine Simulation Setup function block does not reference a master APU provider (such as an Angular Clock Setup function block).
- Case 2: There is a mismatch between the processing unit assigned to the master APU provider and the processing unit, that is indirectly connected to the Engine Simulation Setup function block, for example, via its model port mapping. For details, refer to Creating Multi-Processing-Unit Applications With ConfigurationDesk (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives depending on the case that causes the conflict:

- Case 1: In the Conflicts Viewer, for example, assign a master APU provider to the conflicting Engine Simulation Setup via the Assigned master APU provider property.
- Case 2: Resolve the mismatch by one of the following alternatives:
  - Assign a master APU providing function block that is assigned to the same processing unit as the Engine Simulation Setup function block itself.
  - Change the indirect connection between the Engine Simulation Setup function block and the processing unit, for example, via model port mapping.
  - Change the hardware resource assignment and (if existing) the model port mapping of the master APU provider.

- Assign the behavior model to an application process whose processing unit application is assigned to a matching processing unit.
- Assign the application process (containing a behavior model) to another processing unit application to match the processing unit assignment.

Function block: No timestamp provided

The assigned hardware resource does not support timestamp capturing.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Capture timestamp property to Disabled.
- Assign a hardware resource to the function block that supports timestamp capturing. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: No wavetable imported

No wavetable file specifying the crankshaft wheel is imported.

The crankshaft wheel in use must be specified via an imported wavetable file.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, import a wavetable file with crankshaft specifications via the Imported wavetable file property.

Function block: No wavetable imported

The function block must reference one or more wavetable files.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy Specify the Imported wavetable files property value by importing one or more wavetable files, which is possible via the Import Wavetables dialog. You can open the dialog from the Conflicts Viewer, for example.

Function block: Too many bits per SPI cycle

The required buffer size for the related SPI Cycle function exceeds the maximum allowed value (2048 bits).

The required buffer size is specified by the Number of words and Number of bits per word property values and can be calculated as follows:

Buffer size = floor ((Number of bits per word + 15) / 16) · 16 · Number of words.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, decrease the values of the Number of words and/or the Number of bits per word properties so that the required buffer size is not larger than 2048 bits (refer to the previous formula).

Function block: Too many instances of function type 'FuSa System Monitoring'

More than one FuSa System Monitoring function block is assigned indirectly to the same processor board via an application process.

Only one FuSa System Monitoring function block is supported for each processor board

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	✓

Remedy In the Conflicts Viewer, for example, delete all but one FuSa System Monitoring function block that are indirectly assigned to the same processor board (for example, MicroAutoBox III) from the ConfigurationDesk application.

Function block: Too many LIN masters assigned

A LIN function block is assigned to multiple LIN bus access requests. Two or more of these LIN bus access requests contain a LIN master.

Each physical LIN bus must have exactly one LIN master. If one LIN function block is assigned to multiple bus access requests with two or more LIN masters, exactly one of the LIN masters must be active at run time.

Note

If you do not resolve this conflict before you build a real-time application, you must manually ensure that exactly one LIN master is active at run time.

Tip

If the bus access requests that contain the LIN masters result from different bus configurations, you can add the Bus Configuration Enable feature to the affected bus configurations. If you do this, you can deactivate all the bus configurations except for one at run time to have exactly one LIN master active at a time. In this case, you do not have to resolve this conflict. For more information, refer to Enabling and Disabling Bus Configurations (ConfigurationDesk Bus Manager Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy Perform the following steps:

1. In the Conflicts Viewer, determine the bus access requests whose LIN master property is set to True.
2. Select these bus access requests in the Bus Access Requests table and change the bus access assignment except for one, for example, via the Bus Access Assignment command.

Function block: Too many time synchronization domains assigned

Function blocks that use different time domains are assigned to the same physical Ethernet controller.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the affected function blocks to the same time domain.
- In the Properties Browser, for example, assign different physical Ethernet controllers to the affected function blocks.

Function block: Too many wavetables assigned

The specified number of assigned wavetables exceeds the maximum value that is supported by the assigned hardware resource.

The maximum number of wavetables per function block is limited by the board's wavetable memory as follows:

- DS2680 I/O Unit: Max. 24 wavetable references
- DS2621 Signal Generation Board: Max. 12 wavetable references

- DS6101 Multi-I/O Board: Max. 12 wavetable references
- DS6202 Digital I/O Board: Max. 4 wavetable references
- DS6241 D/A Board: Max. 12 wavetable references

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Decrease the total number of assigned wavetables with respect to the limit of the assigned hardware resource via the **Imported wavetable files** property.
- Assign a hardware resource that supports the specified number of assigned wavetables. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Unfavorable angle sample distance

Depending on the rotating speed of the simulated or measured device (e.g., crankshaft), samples may not be captured, because the following two cases occur in combination:

- The sample trigger source is set to **Angle**.
- The distance between two angle samples is lower than a critical distance (see formula below).

To ensure that samples are captured at all APU speeds, the distance between two angle samples must not fall below the following value:

Critical distance = Minimum sample period * Maximum APU speed.

The minimum sample period depends on the hardware resource assigned to the function block. The maximum APU speed depends on the assigned APU provider. For hardware-specific values, refer to dSPACE Help.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy Choose one of the following alternatives:

- In the **Conflicts Viewer**, for example, specify an angle sample distance (via the **Angle sample distance** property) that is higher than the critical distance (see formula above).
- Reduce the maximum APU speed of the master APU provider via the **Maximum speed** property.
- Specify a setting other than **Angle** for the sample trigger source.

Function block: Unfavorable angle sequence triggers

Depending on the rotating speed of the simulated or measured device (e.g., crankshaft), sequences may not be captured, because the following two cases occur in combination:

- The sequence trigger source to Angle.
- The distance between two angle sequence trigger positions is lower than a critical distance (see formula below).

To ensure that sequences are captured at all APU speeds, the distance between two angle sequence trigger positions must not fall below the following value:

Critical distance = Minimum sample period * Maximum APU speed.

The minimum sample period depends on the hardware resource assigned to the function block. The maximum APU speed depends on the assigned APU provider. For hardware-specific values, refer to dSPACE Help.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Specify a distance between two angle sequence trigger positions (via the Angle sequence trigger property) that is higher than the critical distance (see formula above).
- Reduce the maximum APU speed of the master APU provider via the Maximum speed property.
- Specify a setting other than Angle for the sequence trigger source.

Function block: Unfavorable pulse configuration

The maximum pulse length of a nibble pulse might be longer than or equal to the minimum pulse length of the synchronization pulse.

In the Conflicts Viewer, the Maximal nibble length property and the Minimal sync pulse length property show the calculated pulse lengths.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy In the Conflicts Viewer, for example, check the properties for the SENT timing and set values that ensure a sufficiently long sync pulse length. For basics on the timing of SENT pulses, refer to Basics on the SENT Protocol (ConfigurationDesk I/O Function Implementation Guide 🕮).

Function block: Unresolved channel request

A hardware resource has been assigned, but is no longer required by the function block (= obsolete).

Example: A function block first requires five channels, but needs only two channels after reconfiguration. Three channels are obsolete but still assigned to the function block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy In the Conflicts Viewer, delete the assignment to resolve the conflict. You can use obsolete channels without explicitly deleting the assignment. In this case, obsolete channel requests and their assignments are removed automatically after reassigning the channel.

Function block: Unresolved elements

A port of the function block is obsolete. The port is mapped but due to configuration changes to the function block the port is no longer required.

The mapping is not deleted immediately, because this simplifies the remapping of the corresponding model port block or device block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Remove the mapping line from the obsolete port. This port then is deleted.

Function block: Unresolved function block type

You imported a ConfigurationDesk application which contains custom function blocks whose accompanying plug-in is not installed on your host PC.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Provide the missing plug-in. After this, the function block is resolved and you can use the block without restrictions.

Custom functions must be copied to a global or project-specific custom functions directory (refer to Configuration Page (ConfigurationDesk User Interface Reference 🕮)) or installed via dSPACE add-on.

Function block: Unresolved FuSa configuration conflicts

At least one function block that is related to Functional Safety (FuSa) has a conflicting configuration.

The build process is aborted if only one FuSa function block contains a conflicting configuration.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Resolve all configuration conflicts of the affected FuSa function blocks.

Function block: Unsupported activation by FRU relay

The Activation by FRU relay property is set to Allowed for a signal port even though you have configured current enhancement for that signal.

Activation by FRU relay is not supported in case of current enhancement. You can allow Activation by FRU relay, for example, to support the simulation of multiple failures in the experiment software.

- For more information on Activation by FRU relay and simulation of multiple failures, refer to Specifying Failure Simulation Settings (ConfigurationDesk Real-Time Implementation Guide 🕮).
- For more information on current enhancement, refer to see Specifying Current and Voltage Values for Channel Multiplication (ConfigurationDesk I/O Function Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy In the Conflicts Viewer, for example, set the Activation by FRU relay property to Not allowed.

Function block: Unsupported angle sequence triggers

No sequence is captured, because the following two cases occur in combination:

- The sequence trigger source is set to Angle.
- The distance between two angle sequence trigger positions is lower than the minimum angle sample distance.

To capture sequences, the distance between two angle sequence trigger positions must not fall below the following value:

Minimum angle sample distance = 2 * 720° / 2^{16}

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a setting other than Angle for the sequence trigger source.
- Specify a distance between two angle sequence trigger positions (via the Angle sequence trigger property) that is higher than the minimum angle sample distance (see formula above).

Function block: Unsupported ECU interface container for MicroAutoBox III

For the ECU Interface Configuration function block, the following applies:

- An ECU interface container (EIC file) of version 3.0.0 or earlier is imported.
- An Ethernet Setup function block is referenced to which a hardware resource provided by the MicroAutoBox III is assigned.

Hardware provided by the MicroAutoBox III supports ECU interface containers as of version 4.0.0. For more information, refer to Compatibility of ConfigurationDesk 6.7 (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Import an ECU interface container that is supported by the MicroAutoBox III to the ECU Interface Configuration function block. For instructions, refer to How to Import ECU Interface Container (EIC) Files (ConfigurationDesk Real-Time Implementation Guide 🕮).
- Assign a hardware resource that supports the imported ECU interface container to the referenced Ethernet Setup function block. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).
- Select an Ethernet Setup function block to which a suitable hardware resource is assigned as the Referencing Configuration in the ECU Interface Configuration function block.

Function block: Unsupported event generation for capture trigger source = 'Model-based'

The settings for the Event generation property and the Sequence trigger source property are not compatible.

Event generation is not supported for a Model task trigger source.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy In the Conflicts Viewer, for example, choose one of the following alternatives to make the properties match:

- Set the Event generation property to Disabled.
- Change the Sequence trigger source property to Sequence end, Threshold, Angle, Digital threshold or Trigger function.

Function block: Unsupported function trigger

The Trigger source property of the function block is set to I/O function, but the assigned hardware resource does not support any function block that can work as a trigger function provider.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Set the Trigger source property to another trigger source, for example, to Edge.
- Assign a hardware resource that supports an I/O function as trigger source. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Unsupported maximum message count

The specified value for the Maximum message count property exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Maximum message count property that matches the maximum value supported by the assigned hardware resource.
- Assign a hardware resource that supports the required number of nibbles. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Unsupported measurement point

This conflict is generated in the following cases:

- The specified settings for the Measurement point property and the Signal type property do not match.

  Note

  Current signals can only be measured on the load side and not on the ECU side.

- The specified setting for the Measurement point property is not supported by the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, change the setting for the Measurement point property or the Signal type property to match the setting of the other.
- In the Conflicts Viewer, for example, select a setting for the Measurement point property that is supported by the assigned hardware resource.
- Assign a hardware resource that supports the specified setting. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Unsupported number of angle sequence triggers

The specified value for the Number of angle sequence triggers property exceeds the valid range.

The number of angle sequence triggers must be within the range 1 … 72.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify a value for the Number of angle sequence triggers property in the valid range.

Function block: Unsupported number of signals

The specified value for the Number of signals property exceeds the possible value range of the assigned hardware resource.

The function block synchronously generates up to 32 PWM signals. You specify the number of non-inverted signals with the Number of signals property. If you use inverted signals, the inverted PWM signals are automatically added.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, choose one of the following alternatives:

- Specify a value for the Number of signals property that matches the max. number of signals: 32 signals if no inverted signals are generated, otherwise 16 signals.
- Set the Inverted phase generation property to Disabled.

Function block: Unsupported port settings (CRC, CRC Modify, CRC Value)

The setting of at least one property of the CRC, CRC Modify or CRC Value function ports (for example, Initial value) is not supported or outside the possible value range.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reduce the specified value for the related properties with respect to the possible value range.

Function block: Unsupported port settings (id, data)

The setting of at least one property of the Serial_Message - ID or Serial Message - Data function port (for example, Initial value) is outside the possible value range.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reduce the specified value for the related properties with respect to the possible value range.

Function block: Unsupported port settings (noise amplitude)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that match the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Unsupported reverse crankshaft signal

The reverse crank mode is enabled at the Reverse Crank property but the assigned hardware resource does not support the reverse crank mode.

The assigned hardware resource must support evaluating reverse crankshaft rotation. This is required for using active crankshaft sensors.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Reverse Crank property to Disabled.
- Assign a hardware resource that supports the reverse crank mode. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Unsupported sample period

The value specified for the Sample period property is outside the valid range with respect to the specified sequence length.

If you specify a value of 1 for the Sequence length property, the value for the Sample period property must be at least 2 µs.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy In the Conflicts Viewer, for example, choose one of the following alternatives:

- Specify a value for the Sequence length property that is at least 2.
- Specify a value for the Sample period property that is at least 2 µs.

Function block: Unsupported signal gains

One of the three values specified at the Signal gains property is too high with respect to the two others.

Each maximum possible signal gain value depends on the other two gain values.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, decrease the highest signal gain value so that the related *normalized signal gain* value becomes equal to or lower than the suggested upper limit for the normalized gains. The following equation lets you calculate an approximate possible maximum value, for example, for an excessive signal $gain_a$ value:

Max. signal $gain_a \leq (\text{Signal } gain_b + \text{Signal } gain_c) \cdot 2 - \varepsilon$ (with $\varepsilon \approx 0.01$)

Function block: Unsupported signal offset (current)

The assigned hardware resource does not support the specified signal offset value for the provided signal.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, set the Signal offset property to a value within the allowed range.

Function block: Unsupported signal offset (voltage)

The assigned hardware resource does not support the specified signal offset value for the provided signal.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, set the Signal offset property to a value within the allowed range.

Function block: Unsupported signal offsets

One or more of the values specified at the Signal offsets property are outside the value range specified by the setting of the Measurement range property.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, perform one of the following actions:

- Specify signal offset values that match the range specified by the setting of the Measurement range property.
- Change the setting of the Measurement range property so that the resulting range covers the specified offset values.

Function block: Unsupported signal type

The setting for the Signal type property is not supported by the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a setting for the Signal type property that is supported by the assigned hardware resource.
- Assign a hardware resource that provides the specified setting. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Unsupported sequence duration

The product of Sequence length and Sample period property settings exceeds the maximum sequence duration. The maximum sequence duration must be equal to or less than 1 s.

This conflict can only occur if Sequence trigger source property is set to Model task and the Sample trigger source property is set to Time (sequence-aligned).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, choose one of the following alternatives:

- Reduce the values for the Sequence length property and/or the Sample period property.
- Change the trigger source settings (Sequence trigger source and Sample trigger source), to another supported combination. Refer to Specifying Trigger Sources (Voltage Signal Capture - ADC Type 4) (ConfigurationDesk I/O Function Implementation Guide 🕮).

Function block: Unsupported trigger configuration

The settings specified for the Sequence trigger source property and/or for the Sample trigger source property are not supported by the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, change the trigger source to a setting that is supported by the assigned hardware resource.
- Assign a hardware resource that supports the specified settings. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Function block: Unsupported trigger configuration

The configuration of the trigger sources is not supported by the function block. The conflict occurs in the following cases:

- Case 1: The function block uses the Model task trigger source for both trigger sources (Sequence trigger source and Sample trigger source).
- Case 2: The Model task trigger source is used as Sequence trigger source in combination with enabled event generation.
- Case 3: A trigger function is used for both trigger sources (Sequence trigger source and Sample trigger source) and both trigger sources are assigned to the same trigger function provider.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, choose one of the following alternatives depending on the case that causes the conflict:

- Case 1: Specify a different trigger source than Model task for one of the trigger source properties (Sequence trigger source or Sample trigger source).
- Case 2: Change the Sequence trigger source property to Sequence end or Trigger function or set the Event generation property to Disabled.
- Case 3: Assign a different trigger source provider to one of the two trigger sources (Sequence trigger source or Sample trigger source).

Global Time Domain Conflicts

Global time domain: Duplicate domain ID on channel (bus configuration)

Conflicting communication matrix elements are assigned to a bus configuration. For two or more assigned global time domains of one channel, the communication matrix specifies identical domain IDs.

The communication matrix must specify a unique domain ID for each global time domain of one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Global time domain: Duplicate domain ID on channel (communication matrix)

A communication matrix specifies identical domain IDs for two or more global time domains of one channel.

The communication matrix must specify a unique domain ID for each global time domain of one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Global time domain: Invalid combination of checksum validation setting and data ID lists resulting in default code (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned global time domain, the communication matrix specifies the following settings:

- The CRC validation property of a time slave is set to CRC optional, i.e., if time synchronization is CRC-secured, time synchronization will be validated.
- The Synchronization message data IDs and/or Follow-up message data IDs properties are not specified, or a specified value is invalid.

If time synchronization is CRC-secured and validated, the communication matrix must specify a 16-bit vector for the Synchronization message data IDs and Follow-up message data IDs properties. Each vector element must be in the range 0 … 255.

If you do not resolve this conflict, default code is generated, i.e., the run-time behavior depends on whether time synchronization is CRC-secured:

- Time synchronization is not CRC-secured: No data IDs are required. Therefore, the conflicting communication matrix settings do not affect the processing of the received time synchronization information and the time base of the time slave can be synchronized.
- Time synchronization is CRC-secured: The run-time behavior depends on whether partial validation is enabled for the time slave:
  - Partial validation disabled: The received time synchronization information is not processed and the time base of the time slave is not synchronized.

If the **GTS Validation** feature is added to the affected RX global time domain, the related bits of the Result function port are set to 1.

- Partial validation enabled: The conflicting communication matrix settings are ignored. The received time synchronization information is processed and the time base of the time slave can be synchronized.

For more information on partial validation and the **GTS Validation** feature, 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 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Remove the conflicting element from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- Correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Global time domain: Invalid combination of checksum validation setting and data ID lists resulting in default code (communication matrix)

A communication matrix specifies the following settings for a global time domain:

- The **CRC validation** property of a time slave is set to **CRC optional**, i.e., if time synchronization is CRC-secured, time synchronization will be validated.
- The **Synchronization message data IDs** and/or **Follow-up message data IDs** properties are not specified, or a specified value is invalid.

If time synchronization is CRC-secured and validated, the communication matrix must specify a 16-bit vector for the **Synchronization message data IDs** and **Follow-up message data IDs** properties. Each vector element must be in the range 0 … 255.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Global time domain: Invalid combination of checksum validation setting and data ID lists resulting in no code (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned global time domain, the communication matrix specifies the following settings:

- Time synchronization is CRC-secured and validated, i.e:
  - The CRC-secured property of a time master is set to CRC supported.
  - The CRC validation property of a time slave is set to CRC validated.
- The Synchronization message data IDs and/or Follow-up message data IDs properties are not specified, or a specified value is invalid.

If time synchronization is CRC-secured and validated, the communication matrix must specify a 16-bit vector for the Synchronization message data IDs and Follow-up message data IDs properties. Each vector element must be in the range 0 … 255.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Global time domain: Invalid combination of checksum validation setting and data ID lists resulting in no code (communication matrix)

A communication matrix specifies the following settings for a global time domain:

- Time synchronization is CRC-secured and validated, i.e:
  - The CRC-secured property of a time master is set to CRC supported.
  - The CRC validation property of a time slave is set to CRC validated.
- The Synchronization message data IDs and/or Follow-up message data IDs properties are not specified, or a specified value is invalid.

If time synchronization is CRC-secured and validated, the communication matrix must specify a 16-bit vector for the Synchronization message data IDs and Follow-up message data IDs properties. Each vector element must be in the range 0 … 255.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Global time domain: Invalid property values (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned global time domain, the communication matrix specifies at least one of the following settings:

- The domain ID is not specified or a specified domain ID is invalid.
- The length of the related frame and/or general-purpose PDU is not specified or a specified length is invalid.

For each global time domain, the communication matrix must specify a domain ID, and a length for the related frame and general-purpose PDU. The specified values must be in the following range:

- Domain ID: 0 … 15
- Frame/PDU length: 8 bytes or 16 bytes

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify a valid value for the **Length** property of the frame and/or PDU, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

    Note

    This resolves the conflict only if the communication matrix specifies a valid domain ID.

  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Global time domain: Invalid property values (communication matrix)

A communication matrix specifies at least one of the following settings for a global time domain:

- The domain ID is not specified or a specified domain ID is invalid.
- The length of the related frame and/or general-purpose PDU is not specified or a specified length is invalid.

For each global time domain, the communication matrix must specify a domain ID, and a length for the related frame and general-purpose PDU. The specified values must be in the following range:

- Domain ID: 0 … 15
- Frame/PDU length: 8 bytes or 16 bytes

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify a valid value for the Length property of the frame and/or PDU, for example, in the Conflicts Viewer. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

  Note

  This resolves the conflict only if the communication matrix specifies a valid domain ID.

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Hardware Resource Assignment Conflicts

HW resource assignment: Invalid channel assignment

The assigned channel is not valid due to channel dependencies.

For example, the channel's location must be related to the location of another channel (for example, channel x must be located next to channel y).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reassign the related channels to match the required dependency, such as the channel locations.

HW resource assignment: Invalid channel assignment (by constraint(s))

The assigned channel is not valid due to channel dependencies.

For example, the channel's location must be related to the location of another channel (for example, channel x must be located next to channel y).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reassign the related channels to match the required dependency, such as the channel locations.

HW resource assignment: Invalid HW assigned to Gigalink function

The required hardware settings (port and/or channel specification) for a Gigalink function block are not done completely.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, specify the Gigalink number (Gigalink port) and the Channel number to be used for signal transmission.

Note that, internally, SCALEXIO provides up to four Gigalink ports, but only one of them (default: port number four) is available on the SCALEXIO backplane. For further information, refer to Configuring the Basic Functionality (Gigalink) (ConfigurationDesk I/O Function Implementation Guide 🕮).

HW resource assignment: Invalid platform assigned for Gigalink function

A Gigalink function block is assigned to a processing unit application to which a MicroAutoBox III processor board is assigned.

MicroAutoBox III processor boards do not support Gigalink communication.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Assign a hardware resource that is not associated with a MicroAutoBox III to the relevant processing unit application.
- Delete the Gigalink function block from the ConfigurationDesk application.

HW resource assignment: Invalid resolution value

The specified value for the resolution is outside the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, change the values to match the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Invalid settings for Maximum message count and Nibbles per message

The settings of the Maximum message count and Nibbles per message properties result in a number of nibble pulses that exceeds the maximum number of pulses that is supported by the assigned hardware resource.

The expected number of pulses must not exceed the maximum supported number of pulses and is calculated as follows:

- If the Pause pulse mode property is set to Enabled: Maximum expected number of nibble pulses = (2 + Nibbles per message) x Maximum message count.
- If the Pause pulse mode property is set to Disabled: Maximum expected number of nibble pulses = (1 + Nibbles per message) x Maximum message count.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reduce the settings of the Maximum message count property and/or the Nibbles per message property to drop below the maximum number of nibble pulses supported by the assigned hardware resource.

HW resource assignment: Invalid signal reference

The setting of the Signal reference property is not supported by the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a setting for the Signal reference property that is supported by the assigned hardware resource.
- Assign a hardware resource that supports the required signal reference. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Mismatching Ethernet port type

The Ethernet port type has been changed from automotive to standard Ethernet or vice versa. Therefore, the configured Ethernet port is not compatible to the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Properties Browser, assign an Ethernet switch that provide the PHYs for the configured Ethernet switch ports.

HW resource assignment: Missing channel set or channel assignment

This conflict is generated for custom function blocks in the following cases:

- At least one electrical interface unit of the function block is not assigned to a channel set.
- For an electrical interface unit with an assigned channel set, at least one channel request is not assigned.

ConfigurationDesk determines the types and number of channels required for a function block according to the assigned channel set, the function block features, the block configuration and the required physical ranges. The required channels are called *channel requests*. For more basics, refer to Basics on Hardware Resource Assignment (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy In the Conflicts Viewer, for example, assign a channel set to the function block and/or assign a channel to each channel request of the function block. For instructions on assigning them via the Properties Browser , refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Missing connection to a processing unit

The function block is assigned to a hardware resource which is not connected (directly or indirectly) to a SCALEXIO Processing Unit.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Choose one of the following alternatives depending on your hardware constellation:

- If your hardware topology contains not a processing unit: Extend the hardware topology by adding a SCALEXIO Processing Unit to the system. For instructions, refer to How to Extend Existing Hardware Topologies (ConfigurationDesk Real-Time Implementation Guide 🕮).
- Specify a connection to a SCALEXIO Processing Unit via the Uplink or Downlink properties. These hardware properties are available for IOCNET Routers and IOCNET Link Boards. For information on accessing them, refer to How to Access Hardware Properties (ConfigurationDesk Real-Time Implementation Guide 🕮). For basics on the IOCNET connections in a SCALEXIO system, refer to Network Concept (SCALEXIO Hardware Installation and Configuration 🕮).
- Replace the hardware topology by scanning a registered hardware system, which provides the required connections to a SCALEXIO Processing Unit. For instructions, refer to How to Import a Hardware Topology (ConfigurationDesk Real-Time Implementation Guide 🕮) .
- Assign a hardware resource that is connected directly or indirectly to a SCALEXIO Processing Unit. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Missing or invalid resource assignment for referenced feature provider

The function block references a feature provider, for example, an Angular Clock Setup function block, which has no or an invalid hardware resource assigned to its channel request.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, select a suitable feature provider that already has assigned hardware resources to its channel request.
- In the Conflicts Viewer, for example, assign a hardware resource to the referenced feature provider, for example a suitable angle unit to the Angular Clock Setup function block.

HW resource assignment: Missing piggyback transceiver

The hardware resource assigned to a CAN function block with Transceiver type = ISO 11992-1 Truck and Trailer CAN is not equipped with a piggyback module.

For more information on piggyback-related CAN function block settings, refer to CAN (ConfigurationDesk I/O Function Implementation Guide 🕮).

For more information on the required hardware, refer to DS2671 Bus Board (SCALEXIO Hardware Installation and Configuration 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Assign a hardware resource equipped with a piggyback module. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Multiple application processes connected

Two or more function blocks access channels of the same I/O module and are assigned to different application processes.

Some I/O modules of the MicroAutoBox III require all function blocks accessing their channels to be assigned to the same application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Assign the relevant function blocks to the same application process. A function block is implicitly assigned to an application process via:

- Its mapping lines to model port blocks of model implementations that are assigned to application processes.
- The assignment of its I/O events to tasks that are assigned to application processes.
- The hardware assignment: A function block must have a direct or indirect hardware connection to the processing hardware.

HW resource assignment: Multiple assignments to channel requests

A channel is assigned to more than one channel request in your ConfigurationDesk application.

If the channel is already assigned to the request of another function block, it's name is displayed in parentheses.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, assign a channel to the channel request that is not yet used. Unused channels are displayed in black letters.

If the assigned channel set does not provide unused channels, you can assign another channel set to the function block. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Multiple boards assigned

The channel sets assigned to a function block which contains several electrical interface units are located on different I/O boards.

For this specific function block, all hardware resources assigned must be located on the same I/O board.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Assign hardware resources from the same I/O board to all the electrical interface units of the function block. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Multiple functions assigned to the same Gigalink channel

A channel of the same Gigalink port (Gigalink number) is used for signal transmission by more than one Gigalink function block in your application.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, change the Channel number and/or Gigalink number properties to a combination that is not yet used.

Note that, internally, SCALEXIO provides up to four Gigalink ports, but only one of them (default: port number four) is available on the SCALEXIO backplane. For further information, refer to Configuring the Basic Functionality (Gigalink) (ConfigurationDesk I/O Function Implementation Guide 🕮).

HW resource assignment: No angle position provided

The assigned hardware resource does not support capturing of angle positions.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, perform one of the following steps depending on the related function block:
  - Set the Capture angle position property to Disabled.
  - At the Feature activator property, disable Capture angle positions.
- Assign a hardware resource to the function block that supports capturing of angle positions. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Too many connected master APUs

In a ConfigurationDesk application including angle-based signal processing, too many slave APU to master APU connections exist for a hardware resource.

Hardware resources with slave APUs (for example, the DS2621 Signal Generation Board with 1 slave APU or the DS2680 I/O Unit with 6 slave APUs) must not be connected to more master APUs than slave APUs are available.

The following example shows how connections between a single slave APU and two master APUs are established, causing the conflict:

- A channel of a DS2621 Signal Generation Board is assigned to a Crank/Cam Current Sink function block.
- A different channel from the same board is assigned to a different Crank/Cam Current Sink function block.
- Each Crank/Cam Current Sink function block is assigned to a different Engine Simulation Setup block via Assigned simulation setup property.
- Each Engine Simulation Setup function block is assigned to a different Angular Clock Setup function block via Assigned master APU provider property.
- Two angle units are assigned to the Angular Clock Setup function blocks via hardware resource assignment.

Now connections from one slave APU (on the DS2621) to two master APUs (angle units) have been established and the conflict is generated.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, assign different hardware resources to the involved function blocks.

HW resource assignment: Too many functions requesting same type of resource

In a ConfigurationDesk application, too many function blocks of the same type are assigned to the same I/O board.

Some I/O boards support several function block types only with a limited number of function blocks. This number is lower than the number of provided I/O channels. For example, the DS6202 Digital I/O Board allows the following maximum number of function blocks to be assigned to the I/O board:

- 6 Digital Incremental Encoder In function blocks
- 6 Digital Incremental Encoder Out function blocks
- 16 Digital Pulse Capture function blocks
- 16 Wavetable Digital Out function blocks
- 16 Angular Wavetable Digital Out function blocks
- 16 Waveform Digital Out function blocks
- 4 Digital Pulse Out function blocks
- 8 SENT In function blocks
- 8 SENT Out function blocks

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Assign hardware resources from a different I/O board to the involved function blocks.

HW resource assignment: Too many instances of function type

One of the following cases apply:

- Too many function blocks of the same type are assigned to the same processing unit application.

  For example, a maximum of 64 Non-Volatile Memory Access function blocks is supported for one processing unit application.
- The function block is implicitly assigned to a processing unit (via processing unit application) that is not supported by the related function block type.

For example, the Temperature System Monitoring function block type does not support the DS6001 Processor Board, or the SCALEXIO Processing Unit. This type only supports the DS1403 Processor Board, i.e., the processing unit of the MicroAutoBox III.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives depending on the case that causes the conflict:

- Reduce the number of function blocks that are assigned to the related processing unit application.
- Use the function block only in combination with a processing unit that is supported by the function block type.

HW resource assignment: Too many wavetables

The specified number of imported and default wavetables exceeds the maximum value supported by the assigned hardware resource.

The I/O board memory to store wavetables limits the total number of wavetables that can be referenced by the Crank/Cam and Angular Wavetable function blocks.

The total number of wavetables per I/O board is limited as follows:

- DS2680 I/O Unit: Max. 24 wavetables.
- DS2621 Signal Generation Board: Max. 12 wavetables.
- DS6101 Multi-I/O Board: Max. 12 wavetables.
- DS6202 Digital I/O Board: Max. 16 function blocks with up to 4 wavetables per function block.
- DS6241 D/A Board: Max. 12 wavetables.

A wavetable that is referenced N times by different Crank/Cam or Angular Wavetable function blocks is stored N times in the wavetable memory. Default wavetables for function blocks without imported wavetable files are stored N times in the wavetable memory.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Decrease the total number of wavetables with respect to the limit of the assigned hardware resource via the Imported wavetable files property.

- If you use several function blocks that reference wavetables in your application, assign channel sets from different I/O boards to some function blocks.
- Assign a hardware resource that supports the specified number of wavetables. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unresolved HW resource assigned

A function block is assigned to a hardware resource which is no longer available in the hardware topology.

The hardware resource was assigned while it was available in the hardware topology. However, the hardware topology of the application was changed and the current hardware topology does not contain the hardware resource (= missing or unresolved resource).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Add the missing hardware to the hardware topology.
- In the Conflicts Viewer, for example, assign a different available hardware resource. Note that assigning a different hardware resource may introduce wiring conflicts, so the external cable harness must be recalculated.

HW resource assignment: Unsupported analog lowpass filter feature

The assigned hardware resource does not provide an analog low-pass filter.

You can use the analog anti-aliasing (low-pass) filter, for example, to reduce signal noise.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Analog low-pass filter property to Disabled.
- Assign a hardware resource that provides an analog low-pass filter for input signals. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported CAN FD mode

The assigned hardware resource does not support the CAN FD mode.

The DS2671 Bus Board, for example, supports the CAN FD mode.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the CAN FD mode property to Disabled.
- Assign a hardware resource that supports the CAN FD mode. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported current measurement feature

Current measurement is not supported by the assigned hardware resource.

With the current measurement feature, you can measure currents via an additional function port during run time.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Current measurement property to Disabled.
- Assign a hardware resource that provides current measurement. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported current value(s)

The specified values for one or more of the current values at the electrical interface (for example, Data pulse current) are outside the possible value range of the assigned hardware resource.

The value range for specific properties changes depending on the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy The Conflicts Viewer shows the range of possible values in the Possible Values column.

Choose one of the following alternatives:

- In the Conflicts Viewer, for example, change the values to match the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported current values (current tooth and current gap)

The specified value for the Current tooth or the Current gap property exceeds the valid value range of the assigned hardware resource.

The properties let you specify the current that is generated when a crankshaft tooth or a camshaft marker is detected (wavetable value > 0.0).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Current tooth or Current gap property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported data width

The setting of the Data width property exceeds the maximum available data width for the non-volatile memory of the related processing hardware.

The maximum value for the data width of non-volatile memory of MicroAutoBox III, SCALEXIO LabBox, SCALEXIO AutoBox, and SCALEXIO Processing Units is 4,096.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reduce the setting of the Data width property to a value that is supported by the related processing hardware.

HW resource assignment: Unsupported deadband value

The specified value for the Deadband property exceeds the possible value range of the assigned hardware resource.

A deadband between the rising and falling edges of the generated PWM signal and its inverted PWM signal avoids shoot through currents on a connected half bridge.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Deadband property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported digital filter feature

The assigned hardware resource does not provide the use of digital filters.

You can use digital filters, for example, to reduce signal noise. smooth the input signal or to reduce signal noise, or to adapt the signal bandwidth to a specific model sample frequency in order to avoid aliasing problems.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Digital filter property to Disabled.

- Assign a hardware resource that provides the use of digital filters for input signals. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported edge count

The specified value for the Edge count property exceeds the possible value range of the assigned hardware resource. The property lets you specify the number of detected edges after which an event is generated.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Edge count property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that supports the required edge count value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported edge type

The assigned hardware resource supports only falling edge detection.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Edge type property to Falling.
- Assign a hardware resource that supports rising edge detection. Refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported entry in Data Vector

A value specified at the Data Vector function port is not supported by the assigned hardware resource.

The value range for specific properties at the Data Vector function port, for example, the Initial value property, changes depending on the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the affected property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported event delay

The specified value for the Event delay property exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Event delay property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported event downsampling

The specified value for the Event downsampling property exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Event downsampling property that matches the value range of the assigned hardware resource.

- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported event generation feature

Event generation is not supported by the assigned hardware resource.

With event generation you can generate I/O events that you can use in an executable application to trigger a task.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Event generation property to Disabled.
- Assign a hardware resource that provides event generation. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported failure simulation feature

The failure simulation feature is not supported by the assigned hardware resource.

If you use failure simulation for the signals in your function block, you can control the failure simulation with the experiment software.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Failure simulation property to Not required.
- Assign a hardware resource that provides failure simulation for its hardware channels. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported feedthrough

The feedthrough mode for the bus lines (CAN, FlexRay) is not supported by the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, disable the feedthrough mode by setting the Feedthrough mode property to False.
- Assign a hardware resource that provides the feedthrough mode. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported fuse feature

Monitoring electronic fuses is not supported by the assigned hardware resource.

The Configurable fuse property lets you enable the control of the electronic fuses. Then, you can monitor the state of the electronic fuses and reset them (after removing the failure) in your experiment software.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Configurable fuse property to Not required.
- Assign a hardware resource that provides electronic fuses to be monitored. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported galvanic isolation feature

Galvanic isolation is not supported by the assigned hardware resource.

You can identify a DS2680 I/O Unit which supports galvanic isolation by the Galvanic isolation property.

You can access it as follows: In the Hardware Resource Browser navigate the hardware tree of the SCALEXIO system to the Analog Out 2 item (via DS2680 I/O Unit - DS2680 I/O Module). If the Galvanic isolation property has the Galvanic isolated or the Switchable setting, the DS2680 supports this feature.

Note that if you create a hardware topology from scratch, ConfigurationDesk adds a new DS2680 I/O Unit to the topology that supports switchable galvanic isolation. Make sure that the galvanic isolation support of the DS2680 in your ConfigurationDesk application corresponds to the DS2680 in your SCALEXIO system, which is displayed in the Platform Manager. In the Platform Manager,

you can also use the relevant property of the Analog Out 2 channel type to check the board's support of galvanic isolation.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Electrical interface selection property to Internal load (non-isolated reference).
- Assign a hardware resource that provides galvanic isolation. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported generation mode

The specified setting for the Generation mode property is not supported by the assigned hardware resource.

Enabling Generation mode allows the simulation of a reverse crank sensor signal. For general information on the direction sensitive (reverse) crank sensor signal, refer to Basics on Reverse Crankshaft Signal Generation (ConfigurationDesk I/O Function Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Generation mode property to Disabled.
- Assign a hardware resource that supports the generation mode. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported idle value

The specified value for the Idle value property exceeds the possible value range of the assigned hardware resource.

The Idle value specifies the signal level when no waveform is generated.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Idle value property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported interface type

The specified setting for the Interface type property is not supported by the assigned hardware resource.

You have to specify the required input circuit or output circuit (= interface type) for the signal that is connected to the function block. This helps you to assign a suitable channel set of the dSPACE real-time hardware, because the associated channel types support different potentials. For example, the Flexible In 1 channel type is galvanically isolated to the ground potential of the dSPACE real-time hardware, so you can connect an individual arbitrary potential. For the Flexible In 2 or Digital In 1 channel types, the interface type setting has to be Ground-based, because the Reference signal port is hard-wired to the internal ground of the dSPACE real-time hardware. For basics, refer to Specifying the Circuit Type for Voltage Input Signals (ConfigurationDesk I/O Function Implementation Guide 🕮) or Specifying the Circuit Type for Analog Output Signals (ConfigurationDesk I/O Function Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, select a setting for the Interface type property that is supported by the assigned hardware resource.
- Assign a hardware resource that supports the required interface type. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported interface type or high reference potential

This conflict occurs in one of the following cases:

- Case 1: The setting of the Interface type property is not supported by the assigned hardware resource.

The interface type defines how the digital outputs are driven to get the required output signal (binary 0 or binary 1). This helps you assign a suitable channel set of the dSPACE real-time hardware, because the associated channel types support different interface types.

For example, the **Flexible Out 1** channel type provides channels that are galvanically isolated from the ground potential of the dSPACE real-time hardware. Each channel has its own electrical connection and is independent of the other channels. This allows the connection of a potential shifted signal. If the channel type is **Digital Out 2**, the reference signal is hard-wired to the internal ground of the dSPACE real-time hardware.

- Case 2: The setting of the **High reference potential** property is not supported by the assigned hardware resource.

  A high reference potential only is required, if you want to operate the digital outputs as high-side switch or in push-pull configuration. Specifying this property helps you assign a suitable channel set of the dSPACE real-time hardware, because the associated channel types support different high reference potentials.

  For example, if the channel type is **Digital Out 1**, the internal VBat of the dSPACE real-time hardware is used as high-reference potential. The **High Reference** signal port is not available,

For basics, refer to Configuring Standard Features of the Electrical Interface (ConfigurationDesk I/O Function Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives depending on the case that causes the conflict:

- Case 1: In the **Conflicts Viewer**, for example, specify an interface type for the **Interface type** property that is supported by the assigned hardware resource.
- Case 2: In the **Conflicts Viewer**, for example, specify a reference potential for the **High reference potential** property that is supported by the assigned hardware resource.
- Both cases: Assign a hardware resource that supports the required interface type or the required high reference potential. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported linkspeed

The assigned Ethernet controller does not support the specified link speed.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy In the Conflicts Viewer, for example, specify a link speed that is supported by the assigned Ethernet controller.

HW resource assignment: Unsupported maximum current

The specified value for the Required current property exceeds the possible value range of the assigned hardware resource.

The property lets you specify the required maximum current of the connected signal of the external device (e.g., ECU). ConfigurationDesk uses the specified value to calculate the number of hardware channels needed. With this channel multiplication feature, the max. current of a single hardware channel can be enhanced. For more details, refer to Specifying Current and Voltage Values for Channel Multiplication (ConfigurationDesk I/O Function Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Required current property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported maximum current for central FIU hardware

The hardware topology in your active ConfigurationDesk application contains a central FIU hardware which supports failure simulation but this hardware does not support the specified required current range.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Failure simulation property to Not required.
- In the Conflicts Viewer, for example, reduce the specified value for the Required current property with respect to the possible value range of the central FIU hardware. For example, the DS2680 I/O Unit provides only a maximum current of 48 A.
- Extend the hardware topology by adding a central FIU hardware which provides the required maximum current range. For example, the DS2642 FIU & Power Switch Board provides a maximum current of 80 A. For instructions, refer to How to Extend Existing Hardware Topologies (ConfigurationDesk Real-Time Implementation Guide 🕮).
- Replace the hardware topology by scanning a registered hardware system, which contains a central FIU hardware which provides the required maximum current range. For example, the DS2642 FIU & Power Switch Board provides a maximum current of 80 A. For instructions, refer to How to Import a Hardware Topology (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported maximum current for central FIU hardware if Short to VBAT

This conflict is caused if all the following conditions occur:

- The Failure simulation property of the electrical interface of the affected function block is set to Required.
- The Short to VBAT property is set to Allowed for at least one signal port of the function block.
- No Power Switch function block is instantiated in your active ConfigurationDesk application or you are using at least one Power Switch function block whose maximum ampacity is lower than the specified Required current value for the affected function block.

  The maximum ampacity for "Short to VBAT" failures at function blocks with channel multiplication depends on the channel type as follows:

  - Power Switch 1 (on DS2642): Number of channels used * 8 A
  - Power Switch 2 (on DS2680): 19.2 A

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Choose one of the following alternatives:

- Add a Power Switch function block to the signal chain and assign a suitable hardware resource to the function block.
- In the Conflicts Viewer, for example, set the Short to VBAT property to Not allowed for all signal ports of the affected function block.
- In the Conflicts Viewer, for example, set the Failure simulation property to Not required for the affected function block.

- In the **Conflicts Viewer**, for example, reduce the specified value for the **Required current** property with respect to your minimum required value range.
- Extend the hardware topology by adding a central FIU hardware which provides the required maximum current range. For example, the DS2642 FIU & Power Switch Board provides a maximum current of 80 A. For instructions, refer to How to Extend Existing Hardware Topologies (ConfigurationDesk Real-Time Implementation Guide 🕮).
- Replace the hardware topology by scanning a registered hardware system, which contains a central FIU hardware which provides the required maximum current range. For example, the DS2642 FIU & Power Switch Board provides a maximum current of 80 A. For instructions, refer to How to Import a Hardware Topology (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported maximum voltage

The specified value for the **Required voltage** property exceeds the possible value range of the assigned hardware resource.

The property lets you specify the required maximum voltage of the connected signal of the external device (e.g., ECU). ConfigurationDesk uses the specified value to calculate the number of hardware channels needed. With this channel multiplication feature, the max. voltage of a single hardware channel can be enhanced. For more details, refer to Specifying Current and Voltage Values for Channel Multiplication (ConfigurationDesk I/O Function Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the **Conflicts Viewer**, for example, specify a value for the **Required voltage** property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported measurement delay time(s)

The specified value(s) for the **Measure delay time** property/ies exceed(s) the possible value range of the assigned hardware resource.

The measurement delay time lets you define the delay time for starting measurement after a trigger occurs.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify value(s) for the Measure delay time property/ies that match(es) the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value(s). For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported minimum frequency

The specified value for the Minimum frequency property exceeds the possible value range of the assigned hardware resource.

For a PWM/PFM In function block, for example, the minimum frequency defines the minimum measurable frequency for zero frequency detection of the input signal.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Minimum frequency property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported minimum pulse duration

The specified value for the Minimum pulse duration property exceeds the possible value range of the assigned hardware resource.

This property lets you specify a filter that evaluates the pulse duration. Pulses that have a duration less than the specified value are ignored.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Minimum pulse duration property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported minimum PWM frequency

The specified value for the Minimum PWM frequency property exceeds the possible value range of the assigned hardware resource.

The minimum PWM frequency is used to detect the absence of the input signal and initiate further current measurements independently from a PWM voltage signal. Frequencies lower than specified are detected as 0 Hz.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Minimum PWM frequency property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported noise generation feature

The noise generation feature is enabled for the function block but not supported by the assigned hardware resource.

With the noise generation feature, you can disturb the output signal with additional noise.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Noise Generation property to Disabled.
- Assign a hardware resource that supports the noise generation feature. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported number of bits

The specified value for the Number of bits property exceeds the possible value range of the assigned hardware resource.

The number of bits equals the digital input lines representing the input signal. If the specified Number of bits property exceeds the number of hardware channels you can allocate, ConfigurationDesk creates the channels but does not connect them to the hardware.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Number of bits property that matches the max. number of channels (= digital input lines) provided by the assigned hardware resource.
- Assign a hardware resource that provides the required number of channels (= digital input lines). For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported number of nibbles

The settings of the Nibbles per message and Maximum message count properties result in a number of nibbles that is not supported by the message buffer of the assigned hardware resource.

The maximum number of nibbles that are supported by the message buffer is 84 nibbles. The number is calculated as follows:

- If the Pause pulse mode property is set to Disabled:

  Maximum number of nibbles = (Nibbles per message + 1) · Maximum message count

- If the Pause pulse mode property is set to Enabled:
  Maximum number of nibbles = (Nibbles per message + 2) · Maximum message count

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, reduce the setting of the Nibbles per message property or the Maximum message count property. Use the error description above to decide on the appropriate settings.

HW resource assignment: Unsupported number of nibbles

The specified value for the Nibbles per message property exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Nibbles per message property that matches the maximum value supported by the assigned hardware resource.
- Assign a hardware resource that supports the required number of nibbles. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported partial networking

The assigned hardware resource does not support partial networking.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Partial networking mode property to Disabled.
- Assign a hardware resource that supports partial networking. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (amplitude)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that match the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (current limit)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that match the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (current)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that match the value range of the assigned hardware resource.

- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (frequency)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that match the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (gap width)

The specified value of at least one property of the related function port exceeds the possible value range.

The possible value range is 0 pitches ... 255 pitches and can be reduced by the Pitch count property.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	✓	–	–	–	–

Remedy In the Conflicts Viewer, for example, choose one of the following alternatives:

- Specify a value for the related property/ies that matches the valid value range.
- Increase the value for the Pitch count property to extend the valid value range.

HW resource assignment: Unsupported port settings (hysteresis)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that match the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (noise amplitude)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that match the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (pulse width)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related property/ies that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (resistance value)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that match the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (speed)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that match the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (threshold)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that match the value range of the assigned hardware resource.

- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (velocity)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (voltage out)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that match the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (voltage)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported port settings (wavetable period)

The specified value of at least one property of the related function port exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the related properties that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported power wake-up feature

The assigned hardware resource does not support power wake-up.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Power wake-up property to Disabled.
- Assign a hardware resource that supports power wake-up. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported pulse delay

A value specified for at least one property of the Pulse Delay function port (for example, Initial value) exceeds the value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Initial value or Stop value property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that supports the required pulse delay value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported pulse width

The specified value for the Speed pulse width and/or Data pulse width property exceeds the possible value range of the assigned hardware resource.

Values for the speed pulse width and the data pulse width are given by the sensor specification of the simulated wheel speed sensor type.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Speed pulse width and/or Data pulse width property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that supports the required pulse width value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported resistance

The specified value for the Resistance property exceeds the possible value range of the assigned hardware resource.

This adjustable property lets you enter the end-to-end terminal resistance value of a potentiometer. It specifies the resistance between the Terminal 1 and the Terminal 2 signal port.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Resistance property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that supports the required resistance value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported sample period

The value specified for the Sample period property exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Sample period property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported standstill detection speed

The specified value for the Standstill detection speed property exceeds the possible value range of the assigned hardware resource.

The adjustable speed value lets you configure a value to detect a minimum wheel speed. If the input value of the Speed function port falls below the configured Standstill detection speed value, the function block changes to the standstill mode with a special output signal.

The Standstill detection speed value is given by the sensor specification of the simulated wheel speed sensor type.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Standstill detection speed property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that supports the required speed value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported threshold

The specified value for the Threshold property exceeds the possible value range of the assigned hardware resource.

The adjustable threshold lets you configure a level for the input signal to transform the input signal into binary values.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Threshold property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that supports the required threshold value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported threshold range value

The specified value for the Threshold or the Digital threshold property exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Threshold or the Digital threshold property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the required value ranges. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported time synchronization feature

The onboard Ethernet controllers of the following processing hardware do not support the precision time protocol (PTP):

- SCALEXIO Processing Unit with the Real-Time PC version 1.0 and 1.1
- DS6001 Processor Board

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Assign an Ethernet controller of an Ethernet board to the function block, such as a controller of the DS6331-PE or DS6335-CS.

HW resource assignment: Unsupported transceiver

The selected transceiver type is not supported by the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, set the Transceiver type property to a transceiver type the assigned hardware resource supports.
- Assign a hardware resource that supports the selected transceiver type. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported timeout value

The specified value for the Timeout property exceeds the value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Timeout property that matches the value range of the assigned hardware resource.

- Assign a hardware resource that provides the required timeout value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported trigger delay

The specified value for the Trigger delay property exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Trigger delay property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported trigger downsampling

The specified value for the Trigger downsampling property exceeds the possible value range of the assigned hardware resource.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Trigger downsampling property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that provides the specified value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported trigger threshold

The specified value for the Trigger Threshold property exceeds the possible value range of the assigned hardware resource.

The adjustable trigger threshold lets you configure the level of the triggering voltage input signal, to avoid, for example, the measurement being triggered by noise.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, specify a value for the Trigger Threshold property that matches the value range of the assigned hardware resource.
- Assign a hardware resource that supports the required threshold value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported tristate feature

The Tristate property is set to Enabled and one of the following conditions holds true:

- The digital outputs of the function block do not operate in Push-pull configuration.
- The assigned hardware resource does not provide the tristate feature.

The tristate feature lets you allow the digital outputs (operated in push-pull configuration) of a function block to be set into high impedance state.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- If you do not want to support the tristate feature at all:
  - In the Conflicts Viewer, for example, set the Tristate property to Disabled.
- If you want to support the tristate feature:
  - In the Conflicts Viewer, for example, set the Interface type property to Push-pull.
  - Assign a hardware resource that supports the tristate feature, for example, the DS2621, DS2680, or DS2690. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource assignment: Unsupported wavetable size

The specified number of wavetable values exceeds the possible value range of the assigned hardware resource.

A Wavetable function block can reference multiple wavetables. However, the total number of wavetable values (wavetable size * number of wavetables) is restricted to a hardware-specific limit. For basics, refer to Basics on Signal

Generation Using Time-Coded Wavetables (ConfigurationDesk I/O Function Implementation Guide 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- Decrease the total number of wavetable values with respect to the limit of the assigned hardware resource via the **Imported wavetable files** property.

  Currently available SCALEXIO hardware, for example, the DS2621 Signal Generation Board, provides a limit of up to 1024 wavetable values.
- Assign a hardware resource that supports the required wavetable value. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Hardware Resource Conflicts

HW resource: Duplicate name

A SCALEXIO rack in the hardware topology of the active ConfigurationDesk application contains two or more units, boards, processing units, or processor boards with the same name.

Unit, board, processing unit, and processor board names must be unique within a SCALEXIO rack.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the **Conflicts Viewer**, for example, enter unique names for the units, boards, processing units, or processor boards.

HW resource: Duplicate name

A SCALEXIO Processing Unit or a LabBox in the hardware topology of the active ConfigurationDesk application contains two or more Ethernet controller or switches with the same name.

Names of Ethernet controllers and Ethernet switches must be unique within a SCALEXIO Processing Unit or a LabBox.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, enter unique names for the Ethernet controllers and switches.

HW resource: Exceeding angular speed of assigned I/O boards

The slave APU of an assigned hardware resource does not support the specified maximum speed of the master APU provider. The functionality of function blocks that require angular functionalities is not guaranteed and might lead to unpredictable results if the supported speed of the virtual engine exceeds the speed of the slave APU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Reduce the Maximum speed property setting of the master APU provider to a value that is supported by all hardware resources and will not be exceeded by your real-time application. 168000 °/s is supported by all relevant hardware resources. Keep in mind that the Maximum speed property does not saturate the speed of the virtual engine.
- Assign a hardware resource to the function block that fulfill the angular speed requirement. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

HW resource: No mounted I/O module for I/O base board

An I/O module is required to provide the I/O interface for the FPGA base board.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose at least one of the following alternatives:

- In the Hardware Resource Browser, add the missing I/O module to the FPGA base board.
- Install an I/O module on the FPGA base board and replace the hardware topology.

HW resource: Too many events assigned

In a ConfigurationDesk application the event generation feature is enabled for too many function blocks which all are assigned to the same I/O board. The number of events exceeds the allowed maximum for the assigned hardware resource. For example, the DS6101 Multi I/O Board allows only max. 8 events per board.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, disable Event generation for that function blocks which do not require the event generation feature.
- Assign a hardware resource that supports the required number of events or reassign some function blocks to other I/O boards so you no longer exceed the maximum number of supported events. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).

Hardware Unit Slot Conflicts

HW unit slot: Multiple boards assigned

One slot of a slot I/O unit is assigned multiple times by different I/O boards.

A slot of a slot I/O unit can only be used once.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy In the Properties Browser, for example, reassign the boards, so that every slot is used only once in a slot I/O unit.

IPDU Conflicts

IPDU: Duplicate J1939 identifier on CAN channel (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For two or more assigned J1939-compliant PDUs, the communication matrix specifies the following settings:

- The IPDUs are mapped to the same CAN channel.
- For at least two of these IPDUs, identical settings are specified for the following properties:
  - Parameter group number
  - Source address
  - Destination address

The tuple [parameter group number; source address; destination address] must be unique for each J1939-compliant PDU of one CAN channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Duplicate J1939 identifier on CAN channel (communication matrix)

A communication matrix specifies the following settings for two or more J1939-compliant PDUs:

- The IPDUs are mapped to the same CAN channel.
- For at least two of these IPDUs, identical settings are specified for the following properties:
  - Parameter group number
  - Source address
  - Destination address

The tuple [parameter group number, source address, destination address] must be unique for each J1939-compliant PDU of one CAN channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Invalid end-to-end protection profile or description (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an ISignal group of an assigned IPDU, the communication matrix specifies an unsupported or invalid end-to-end protection profile.

For the end-to-end protection of ISignal groups contained in IPDUs, the communication matrix must specify a profile that the Bus Manager supports. The specified profile must be compliant to the AUTOSAR specifications.

For an overview of the end-to-end protection profiles the Bus Manager supports, refer to:

- Bus Manager in ConfigurationDesk: Aspects of Supported AUTOSAR Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Aspects of Supported AUTOSAR Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

If you do not resolve this conflict, default code is generated, i.e., no end-to-end protection is used for the affected ISignal group. Nevertheless, the ISignal group and its related IPDU can be transmitted and received at run time.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Remove the conflicting element from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- Correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Invalid end-to-end protection profile or description (communication matrix)

A communication matrix specifies an unsupported or invalid end-to-end protection profile for an ISignal group of an IPDU.

For the end-to-end protection of ISignal groups contained in IPDUs, the communication matrix must specify a profile that the Bus Manager supports. The specified profile must be compliant to the AUTOSAR specifications.

For an overview of the end-to-end protection profiles the Bus Manager supports, refer to:

- Bus Manager in ConfigurationDesk: Aspects of Supported AUTOSAR Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Aspects of Supported AUTOSAR Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Invalid J1939 payload length (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned J1939-compliant PDU, the communication matrix specifies an invalid length.

For each J1939-compliant PDU, the communication matrix must specify a payload length in the range 0 … 1785 bytes.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify a valid value for the **Length** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Invalid J1939 payload length (communication matrix)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned J1939-compliant PDU, the communication matrix specifies an invalid length.

For each J1939-compliant PDU, the communication matrix must specify a payload length in the range 0 … 1785 bytes.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify a valid value for the **Length** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Invalid length of secured IPDU (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned secured IPDU, the communication matrix specifies an invalid length.

For each secured IPDU, the communication matrix must specify a valid length. The valid length depends on whether the communication matrix configures the secured PDU as cryptographic PDU:

- Secured IPDU configured as cryptographic IPDU:

  ```
  valid length ≥ (secured PDU header length +
  freshness value TX length + authenticator TX length +
  message link length)
  ```

- Secured IPDU not configured as cryptographic IPDU:

  ```
  valid length ≥ (secured PDU header length +
  freshness value TX length + authenticator TX length +
  authentic IPDU payload length)
  ```

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify a valid value for the **Length** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Invalid length of secured IPDU (communication matrix)

A communication matrix specifies an invalid length for a secured IPDU.

For each secured IPDU, the communication matrix must specify a valid length. The valid length depends on whether the communication matrix configures the secured IPDU as cryptographic IPDU:

- Secured IPDU configured as cryptographic IPDU:

  ```
  valid length ≥ (secured PDU header length +
  freshness value TX length + authenticator TX length +
  message link length)
  ```

- Secured IPDU not configured as cryptographic IPDU:

  ```
  valid length ≥ (secured PDU header length +
  freshness value TX length + authenticator TX length +
  authentic IPDU payload length)
  ```

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify a valid value for the Length property, for example, in the Conflicts Viewer. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Invalid number of ISignal groups for end-to-end protected IPDU (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned IPDU, the communication matrix specifies the following settings:

- An end-to-end protection definition contains one or more references to the IPDU.
- To at least one of these references, the end-to-end protection definition maps no or more than one ISignal group.

Within an end-to-end protection definition, exactly one ISignal group must be mapped to one IPDU reference. If the end-to-end protection definition references an IPDU more than once, one ISignal group must be mapped to each of these references.

If you do not resolve this conflict, default code is generated, i.e., no end-to-end protection is used for the affected ISignal group. Nevertheless, the ISignal group and its related IPDU can be transmitted and received at run time.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Remove the conflicting element from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- Correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Invalid number of ISignal groups for end-to-end protected IPDU (communication matrix)

A communication matrix specifies the following settings for an IPDU:

- An end-to-end protection definition contains one or more references to the IPDU.
- To at least one of these references, the end-to-end protection definition maps no or more than one ISignal group.

Within an end-to-end protection definition, exactly one ISignal group must be mapped to one IPDU reference. If the end-to-end protection definition references an IPDU more than once, one ISignal group must be mapped to each of these references.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Invalid period for J1939 transport protocol communication (bus configuration)

A bus configuration is configured as follows:

- At least one J1939-compliant PDU with more than 8 data bytes is assigned to the bus configuration, i.e., a transport protocol is used for the J1939 communication.
- For the Timer Event of the Bus Configuration task, a period of more than 0.05 s is specified.

If transport protocols are used for J1939 communication, the Timer Event of the Bus Configuration task should have a period in the range 0 … 0.05 s.

If you do not resolve this conflict, default code is generated, i.e., because the period exceeds 0.05 s, not all of the timing requirements of the transport-protocol-based J1939 communication might be fulfilled at run time. Therefore, timeouts might occur.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Remove the conflicting PDUs from the bus configuration: In the Conflicts Viewer, set the Is assigned property of the bus configuration to False.
- Specify a valid value for the Period property of the Timer Event of the Bus Configuration task, for example, in the Conflicts Viewer.

IPDU: Missing authentic IPDU for secured IPDU (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned secured IPDU, the communication matrix specifies the following settings:

- The secured IPDU is configured as a cryptographic IPDU.
- No authentic IPDU is mapped to the secured IPDU.

To each cryptographic IPDU, the communication matrix must map one authentic IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting

element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Missing authentic IPDU for secured IPDU (communication matrix)

A communication matrix configures a secured IPDU as a cryptographic IPDU but does not map an authentic IPDU to the cryptographic IPDU.

To each cryptographic IPDU, the communication matrix must map one authentic IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Unsupported hierarchical multiplexed IPDUs (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For the dynamic and/or static part of an assigned multiplexed IPDU, the communication matrix specifies at least one other multiplexed IPDU.

For the dynamic and static part of a multiplexed IPDU, the communication matrix must specify only ISignal IPDUs.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting

element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

IPDU: Unsupported hierarchical multiplexed IPDUs (communication matrix)

For the dynamic and/or static part of a multiplexed IPDU, a communication matrix specifies at least one other multiplexed IPDU.

For the dynamic and static part of a multiplexed IPDU, the communication matrix must specify only ISignal IPDUs.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal Conflicts

ISignal: Duplicate ISignal or ISignal group name (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned PDU, the communication matrix defines two or more ISignals or ISignal groups with identical names.

For ISignals and ISignal groups of the same PDU, the communication matrix must define unique names.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal: Duplicate ISignal or ISignal group name (communication matrix)

A communication matrix defines two or more ISignals or ISignal groups with identical names for one PDU.

For ISignals and ISignal groups of the same PDU, the communication matrix must define unique names.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal: Invalid byte alignment (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned ISignal, the communication matrix specifies the following settings:

- The ISignal is an array signal.
- The length and/or start bit position of the ISignal is not byte-aligned.

According to AUTOSAR, array signals must be byte-aligned, i.e.:

- The signal length (in bits) must be 8 or a multiple of 8.
- The signal's start bit position (in bits) must be 0 or a multiple of 8.

If you do not resolve this conflict, default code is generated, i.e., even though the array signal is not byte-aligned, it can be transmitted and received at run time.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application. Choose one of the following alternatives:
  - Remove the conflicting element from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

    Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
  - Specify valid values for the **Length** and/or **Start bit position** properties, or change the signal type via the **Category** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal: Invalid byte alignment (communication matrix)

A communication matrix specifies the following settings for an ISignal:

- The ISignal is an array signal.
- The length and/or start bit position of the ISignal are not byte-aligned.

Array signals must be byte-aligned, i.e.:

- The signal length (in bit) must be 8 or multiples of 8.
- The signal's start bit position (in bit) must be 0 or multiples of 8.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

Specify valid values for the **Length**, **Start bit position**, and/or **Category** properties, or change the signal type via the **Category** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal: Invalid coding defined (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned ISignal, the communication matrix defines no or more than one coding.

For each ISignal, the communication matrix must define exactly one coding.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal: Invalid coding defined (communication matrix)

A communication matrix defines no or more than one coding for an ISignal.

For each ISignal, the communication matrix must define exactly one coding.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal: Invalid length (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned ISignal, the communication matrix does not define a length in bit, or a defined length exceeds the valid range.

For each ISignal, the communication matrix must define a length ≥ 1 bit. The maximum valid ISignal length depends on the coded base data type of the ISignal.

For more information, refer to:

- Bus Manager in ConfigurationDesk: Configurable Settings of ISignals (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Configurable Settings of ISignals (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

Specify a valid value for the Length property, for example, in the Conflicts Viewer. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

Tip

You can change the maximum valid ISignal length by specifying another coded base data type for the ISignal. To access the related ISignal property in the Properties Browser, right-click the conflicting ISignal-to-IPDU mapping in the Conflicts Viewer and select Select Related Elements in Bus Configurations Table from the context menu.

- Resolve the conflict by correcting the original communication matrix. Refer to:

  Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

  Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal: Invalid length (communication matrix)

A communication matrix does not define a length for an ISignal, or a defined length exceeds the valid range.

For each ISignal, the communication matrix must define a length ≥ 1 bit. The maximum valid ISignal length depends on the coded base data type of the ISignal.

For more information, refer to:

- Bus Manager in ConfigurationDesk: Configurable Settings of ISignals (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Configurable Settings of ISignals (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify a valid value for the Length property, for example, in the Conflicts Viewer. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

Tip

You can change the maximum valid ISignal length by specifying another coded base data type for the ISignal. To do so, select the ISignal in the **Buses Browser**, for example. Then, select another data type for the coded **Base data type** property in the **Properties Browser**.

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal: No name defined (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned ISignal, the communication matrix does not define a name.

For each ISignal, the communication matrix must define a name. The name must be unique for all the ISignals and ISignal groups of one IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal: No name defined (communication matrix)

A communication matrix does not define a name for an ISignal.

For each ISignal, the communication matrix must define a name. The name must be unique for all the ISignals and ISignal groups of one IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal Group Conflicts

ISignal group: Invalid end-to-end transformation profile or description (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned ISignal group or ISignal, the communication matrix specifies the following settings:

- The ISignal group or ISignal is protected by an end-to-end transformer.
- For the end-to-end transformer, one of the following settings is specified:
  - The specified **Profile name** references an end-to-end protection profile the Bus Manager does not support.
  - One or more transformer attributes are invalid for the specified end-to-end protection profile.

End-to-end transformers must use an end-to-end protection profile the Bus Manager supports. All the transformer attributes that are required by the used end-to-end protection profile must be specified within the valid range according to the AUTOSAR specifications.

For an overview of the end-to-end protection profiles the Bus Manager supports, refer to:

- Bus Manager in ConfigurationDesk: Aspects of Supported AUTOSAR Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Aspects of Supported AUTOSAR Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

If you do not resolve this conflict, default code is generated, i.e., no end-to-end protection is used for the affected ISignal group or ISignal. Nevertheless, the ISignal group or ISignal and its related IPDU can be transmitted and received at run time.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Remove the conflicting element from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- Correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal group: Invalid end-to-end transformation profile or description (communication matrix)

A communication matrix specifies the following settings for an ISignal group or ISignal:

- The ISignal group or ISignal is protected by an end-to-end transformer.
- For the end-to-end transformer, one of the following settings is specified:
  - The specified **Profile name** references an end-to-end protection profile the Bus Manager does not support.
  - One or more transformer attributes are invalid for the specified end-to-end protection profile.

End-to-end transformers must use an end-to-end protection profile the Bus Manager supports. All the transformer attributes that are required by the used end-to-end protection profile must be specified within the valid range according to the AUTOSAR specifications.

For an overview of the end-to-end protection profiles the Bus Manager support, refer to:

- Bus Manager in ConfigurationDesk: Aspects of Supported AUTOSAR Features (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Aspects of Supported AUTOSAR Features (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal group: Invalid number of ISignal IPDUs for end-to-end protected ISignal group (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned ISignal group, the communication matrix specifies the following settings:

- An end-to-end protection definition contains a reference to the ISignal group.
- The end-to-end protection definition maps no or more than one ISignal IPDU to the ISignal group reference.

Within an end-to-end protection definition, exactly one ISignal IPDU must be mapped to one ISignal group reference.

If you do not resolve this conflict, default code is generated, i.e., no end-to-end protection is used for the affected ISignal group. Nevertheless, the ISignal group and its related IPDU can be transmitted and received at run time.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Remove the conflicting element from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- Correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal group: Invalid number of ISignal IPDUs for end-to-end protected ISignal group (communication matrix)

A communication matrix specifies the following settings for an ISignal group:

- An end-to-end protection definition contains a reference to the ISignal group.
- The end-to-end protection definition maps no or more than one ISignal IPDU to the ISignal group reference.

Within an end-to-end protection definition, exactly one ISignal IPDU must be mapped to one ISignal group reference

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal group: Invalid transformation configuration (communication matrix)

A communication matrix specifies the following settings for an ISignal group or ISignal:

- The ISignal group or ISignal is protected by an end-to-end transformer.
- For the ISignal group or ISignal, at least one of the following references is missing:
  - Reference to a data transformation
  - Reference to a transformation description

ISignal groups and ISignals that are protected by an end-to-end transformer must reference a data transformation and a transformation description.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal group: Multiple end-to-end protected ISignal IPDUs related (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned ISignal group, the communication matrix specifies the following setting: The ISignal group is end-to-end protected but it is referenced by more than one end-to-end protection definition.

An end-to-end-protected ISignal group must be referenced by exactly one end-to-end protection definition within the end-to-end protection set of the communication matrix.

If you do not resolve this conflict, default code is generated, i.e., no end-to-end protection is used for the affected ISignal group. Nevertheless, the ISignal group and its related IPDU can be transmitted and received at run time.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Remove the conflicting element from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- Correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal group: Multiple end-to-end protected ISignal IPDUs related (communication matrix)

A communication matrix specifies the following setting for an ISignal group: The ISignal group is end-to-end protected but it is referenced by more than one end-to-end protection definition.

An end-to-end-protected ISignal group must be referenced by exactly one end-to-end protection definition within the end-to-end protection set of the communication matrix.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal-to-IPDU Mapping Conflicts

ISignal-to-IPDU mapping: Invalid endianness (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned ISignal-to-IPDU mapping, the communication matrix specifies at least one of the following settings:

- For the ISignal-to-IPDU mapping, no or more than one byte layout is defined.
- The position of the start bit is unspecified.
- The endianness is unspecified or invalid.

For each ISignal-to-IPDU mapping, the communication matrix must define exactly one byte layout. The start bit position and endianness must be specified. The start bit position must be ≥ 0. Depending on the ISignal type, the endianness must be either big endian, little endian, or opaque.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

Specify valid values for the **Start bit position** and/or **Endianness** properties of the ISignal-to-IPDU mapping, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

- Resolve the conflict by correcting the original communication matrix. Refer to:

  Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

  Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

ISignal-to-IPDU mapping: Invalid endianness (communication matrix)

A communication matrix contains at least one of the following settings:

- For an ISignal-to-IPDU mapping, no or more than one byte layout is defined.
- The position of the start bit is unspecified.
- The endianness is unspecified or invalid.

For each ISignal-to-IPDU mapping, the communication matrix must define exactly one byte layout. The start bit position and endianness must be specified. The start bit position must be ≥ 0. Depending on the ISignal type, the specified endianness must be either big endian, little endian, or opaque.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify valid values for the **Start bit position** and/or **Endianness** properties of the ISignal-to-IPDU mapping, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

J1939 Network Management Node Conflicts

J1939 network management node: Invalid J1939 name on CAN channel (bus configuration)

One or more J1939 network management nodes are assigned to a bus configuration. For the assigned J1939 network management nodes, the following applies:

- For one or more assigned J1939 network management nodes, the communication matrix specifies at least one of the following settings:
  - For an assigned J1939 network management node, the communication matrix specifies an invalid value for the **Arbitrary address capable**, **Industry group**, **Vehicle system instance**, **Vehicle system**, **Function**, **Function instance**, **ECU instance**, **Manufacturer code**, and/or **Identity number** property.
  - The communication matrix maps two or more of the assigned J1939 network management nodes to the same CAN channel and specifies identical settings for all of the properties above.
- The **J1939 Network Management Enable** feature is added to the affected communication controllers and the feature's **Enable** property is set to **1: Static address handling** or **2: Dynamic address handling**.

The tuple of the following properties must be unique for each J1939 network management node of one CAN channel and within the following ranges:

- Arbitrary address capable: `True` or `False`
- Industry group: 0 … 7
- Vehicle system instance: 0 … 15
- Vehicle system: 0 … 127
- Function: 0 … 255
- Function instance: 0 … 31
- ECU instance: 0 … 7
- Manufacturer code: 0 … 2047
- Identity number: 0 … 2097151

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Set the **Enable** property of the **J1939 Network Management Enable** feature to **0: Disabled**, for example, in the **Conflicts Viewer**. Refer to:

    Bus Manager in ConfigurationDesk: Enabling and Disabling J1939 Network Management (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Enabling and Disabling J1939 Network Management (Bus Manager (Stand-Alone) Implementation Guide 🕮)
  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

J1939 network management node: Invalid J1939 name on CAN channel (communication matrix)

A communication matrix specifies the following settings for a J1939 network management node: The tuple of the following properties is not unique on a CAN channel or at least one of the property values exceeds the valid range:

- Arbitrary address capable
- Industry group
- Vehicle system instance
- Vehicle system
- Function
- Function instance
- ECU instance
- Manufacturer code
- Identity number

The tuple of the following properties must be unique for each J1939 network management node of one CAN channel and within the following ranges:

- Arbitrary address capable: `True` or `False`
- Industry group: 0 … 7
- Vehicle system instance: 0 … 15
- Vehicle system: 0 … 127
- Function: 0 … 255
- Function instance: 0 … 31
- ECU instance: 0 … 7
- Manufacturer code: 0 … 2047
- Identity number: 0 … 2097151

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

J1939 Transport Protocol Node Conflicts

J1939 transport protocol node: Invalid transport protocol address (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned J1939 transport protocol node, the communication matrix specifies an invalid transport protocol address.

For each J1939 transport protocol node, the communication matrix must specify a transport protocol address in the range 0 … 253.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Enable dynamic address handling via the **J1939 Network Management Enable** feature: If the feature is already added, set the feature's **Enable** property to **2: Dynamic address handling**, for example, in the **Conflicts Viewer**. If the feature is not added yet, add the feature to the communication controller of the J1939 transport protocol node, for example, in the **Bus Configurations Table**.

Note

This resolves the conflict only if the transport protocol address of the J1939 transport protocol node is 254.

Tip

The **Conflicts Viewer** displays the J1939 transport protocol node and not the communication controller. To access the controller, right-click the transport protocol node and select **Select Related Elements in Bus Configurations Table** from the context menu.

Refer to:

Bus Manager in ConfigurationDesk: Enabling and Disabling J1939 Network Management (ConfigurationDesk Bus Manager Implementation Guide 🕮)

Bus Manager (stand-alone): Enabling and Disabling J1939 Network Management (Bus Manager (Stand-Alone) Implementation Guide 🕮)

- Resolve the conflict by correcting the original communication matrix. Refer to:

  Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

  Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

J1939 transport protocol node: Invalid transport protocol address (communication matrix)

A communication matrix specifies an invalid transport protocol address for a J1939 transport protocol node.

For each J1939 transport protocol node, the communication matrix must specify a transport protocol address in the range 0 … 253.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN Cluster Conflicts

LIN cluster: Invalid speed (baud rate) defined (communication matrix)

A communication matrix contains at least one of the following settings:

- For a LIN cluster, no baud rate is defined.
- A defined baud rate exceeds the valid range.

For each LIN cluster, the communication matrix must define a baud rate in the range 1 … 20 kbit/s.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify a valid value for the Baud rate property, for example, in the Conflicts Viewer. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN cluster: Too many channels connected (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned LIN cluster, the communication matrix specifies more than one channel.

For each LIN cluster, the communication matrix must not specify more than one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting

element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN cluster: Too many channels connected (communication matrix)

A communication matrix specifies more than one channel for a LIN cluster.

For each LIN cluster, the communication matrix must not specify more than one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN Communication Controller Conflicts

LIN communication controller: Invalid initial schedule table

For a LIN master, a bus configuration references an initial schedule table, but this schedule table does not exist in the assigned communication matrix.

A communication matrix can define multiple schedule tables for a LIN master. The **Initial value** of a LIN Schedule Index function port references the position of a schedule table in the communication matrix. This schedule table is used as the initial schedule table. The specified **Initial value** must be equal to or smaller than the number of schedule tables defined in the communication matrix. Alternatively, you can specify 0 as the **Initial value** to use no initial schedule table at all.

For details on schedule tables, refer to:

- Bus Manager in ConfigurationDesk: Working with LIN Schedule Tables (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Working with LIN Schedule Tables (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy In the Conflicts Viewer, for example, specify a valid Initial value for the LIN Schedule Index function port.

LIN communication controller: Too many channels connected (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned LIN communication controller, the communication matrix specifies more than one channel.

For each LIN communication controller, the communication matrix must not specify more than one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration. In the Conflicts Viewer, set the Is assigned property to False
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)s

LIN communication controller: Too many channels connected (communication matrix)

A communication matrix specifies more than one channel for a LIN communication controller.

For each LIN communication controller, the communication matrix must not specify more than one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN Frame Conflicts

LIN frame: Invalid payload length defined (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned LIN frame, the communication matrix defines a payload length that exceeds the valid range.

For a LIN frame, the communication matrix must define a payload length within the range 1 … 8 bytes.

> **Tip**
>
> The payload length of sporadic and event-triggered frames can exceed this range.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

Specify a valid value for the **Length** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

- Resolve the conflict by correcting the original communication matrix. Refer to:

  Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

  Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame: Invalid payload length defined (communication matrix)

A communication matrix defines a payload length for a LIN frame that exceeds the valid range.

For a LIN frame, the communication matrix must define a payload length within the range 1 … 8 bytes.

Tip

The payload length of sporadic and event-triggered frames can exceed this range.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify a valid value for the **Length** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame: Invalid sender defined (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned sporadic LIN frame or a related unconditional frame, the communication matrix specifies a LIN slave as the transmitting LIN bus node.

Sporadic LIN frames and their related unconditional frames can only be transmitted by the LIN master.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame: Invalid sender defined (communication matrix)

A communication matrix specifies a LIN slave as the transmitting LIN bus node for a sporadic LIN frame or a related unconditional frame.

Sporadic LIN frames and their related unconditional frames can only be transmitted by the LIN master.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame: No collision resolver defined for event-triggered frame (communication matrix)

A communication matrix does not define a collision resolver schedule table for a LIN frame.

For each LIN frame, the communication matrix must define a collision resolver schedule table.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame: Too many LIN frame triggerings connected (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. An assigned LIN frame is used as a substitute frame for a sporadic or event-triggered frame. For this assigned LIN frame, the communication matrix specifies more than one frame triggering for the channel that is also used by the sporadic or event-triggered frame.

For a LIN frame that is used as a substitute frame for a sporadic or event-triggered frame, the communication matrix must specify exactly one frame triggering for the channel that is also used by the sporadic or event-triggered frame.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame: Too many LIN frame triggerings connected (communication matrix)

A communication matrix specifies a LIN frame that is used as a substitute frame for a sporadic or event-triggered frame and has more than one frame triggering for the channel that is also used by the sporadic or event-triggered frame.

For a LIN frame that is used as a substitute frame for a sporadic or event-triggered frame, the communication matrix must specify exactly one frame triggering for the channel that is also used by the sporadic or event-triggered frame.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame: Too many PDU-to-frame mappings connected (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned unconditional LIN frame, the communication matrix specifies more than one PDU-to-frame mapping.

An unconditional LIN frame must not contain more than one PDU-to-frame mapping.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame: Too many PDU-to-frame mappings connected (communication matrix)

A communication matrix specifies more than one PDU-to-frame mapping for an unconditional LIN frame.

An unconditional LIN frame must not contain more than one PDU-to-frame mapping.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN Frame Triggering Conflicts

LIN frame triggering: Duplicate LIN identifier on LIN channel (bus configuration)

Conflicting communication matrix elements are assigned to a bus configuration. For two or more assigned LIN frame triggerings of one channel, the communication matrix specifies identical identifier values.

If the communication matrix specifies an identifier value for a LIN frame triggering, the value must be unique for all the frame triggerings of one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify unique and valid values for the **Identifier** properties, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame triggering: Duplicate LIN identifier on LIN channel (communication matrix)

A communication matrix specifies identical identifier values for two or more LIN frame triggerings of one channel.

If the communication matrix specifies an identifier value for a LIN frame triggering, the value must be unique for all the frame triggerings of one channel.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify unique and valid values for the **Identifier** properties, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame triggering: Invalid frame type (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned LIN frame triggering, the communication matrix references a CAN frame.

Each LIN frame triggering must reference a LIN frame.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame triggering: Invalid frame type (communication matrix)

A communication matrix references a CAN frame for a LIN frame triggering.

Each LIN frame triggering must reference a LIN frame.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame triggering: Invalid property values (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned LIN frame triggering of one channel, the communication matrix specifies an invalid frame identifier.

For a LIN frame triggering of an unconditional or event-triggered frame, the communication matrix must specify a frame identifier within the range 0 … 63.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify a valid value for the Identifier property, for example, in the Conflicts Viewer. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame triggering: Invalid property values (communication matrix)

A communication matrix specifies an invalid frame identifier for a LIN frame triggering.

For a LIN frame triggering of an unconditional or event-triggered frame, the communication matrix must specify a frame identifier within the range 0 … 63.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

Specify a valid value for the **Identifier** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN frame triggering: No frame triggering of related unconditional frame on the same channel (communication matrix)

A communication matrix specifies the following settings for a LIN frame and its frame triggerings:

- An unconditional LIN frame is referenced by an event-triggered or sporadic frame.
- None of the frame triggerings of the unconditional frame and the event-triggered or sporadic frame use the same channel.

An unconditional frame that is referenced by an event-triggered or sporadic frame must have at least one frame triggering on the same channel as the referencing frame.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN Schedule Table Conflicts

LIN schedule table: Invalid LIN schedule table entries (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For entries of an assigned LIN schedule table, the communication matrix specifies invalid settings for the delay, position in table, and/or the used channel.

For each entry, a LIN schedule table must at least specify the following settings:

- Delay
- Position in table

  The values of this setting must be unique for each entry of a schedule table.

- The channel used by the frame triggerings

  The frame triggerings of each entry must use the same channel as the schedule table itself.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

LIN schedule table: Invalid LIN schedule table entries (communication matrix)

For entries of a LIN schedule table, a communication matrix specifies invalid settings for the delay, position in table, and/or the used channel.

For each entry, a LIN schedule table must at least specify the following settings:

- Delay
- Position in table

  The values of this setting must be unique for each entry of a schedule table.
- The channel used by the frame triggerings

  The frame triggerings of each entry must use the same channel as the schedule table itself.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Model Conflicts

Model: Multiple assignments to different application processes

A model implementation is assigned to two or more application processes in the executable application.

The assignment of a model implementation to an application process must be unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, select only one application process to assign the model implementation to.

Model: No source code and no supported library available

The executable application contains an application process to which a Functional Mock-up Unit (FMU), a Simulink implementation container file (SIC file), or a bus simulation container file (BSC file) is assigned, that does not contain source code or a precompiled library in a supported version. You can work with such FMUs, SIC files, and BSC files as usual. This means that you can map model port blocks, configure tasks, etc. However, you cannot perform a build process for your executable application.

To perform a successful build process, FMUs, SIC files, or BSC files that are assigned to an application process must contain source code or precompiled libraries in a supported version.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Remove the FMU,SIC file, or BSC file that does not have source code or a precompiled library from the application process, and assign an FMU, SIC file, or BSC file with source code or a precompiled library to it.

Model: Unsupported combination of assigned processing unit and model(s)

Binary libraries contained in a model container or referenced by a Simulink model are incompatible with the operating system of the processing unit assigned to the relevant processing unit application.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy The Simulink model or model container must be replaced with a Simulink model or model container that uses binary libraries that are compatible with the registered platform.

Model: Unsupported platform assigned to V-ECU

The V-ECU implementation only contains platform packages that do not match the processing unit that is assigned to the related processing unit application, or that are not supported by ConfigurationDesk.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Depending on the cause of the conflict, choose one of the following alternatives:

- Remove the V-ECU implementation from the ConfigurationDesk application and replace it with one that contains platform packages that match the assigned processing unit.
- If the V-ECU implementation only contains platform packages that do not match the processing unit assigned to the related processing unit application, assign a processing unit to the relevant processing unit application that is supported by the V-ECU implementation.

Model Port Block Conflicts

Model port block: Duplicate ID

The model topology contains two or more model implementations with the same interface, which means, that two or more model port blocks have the same ID. At least two of these model implementations with the same interface are assigned to an application process. These model port blocks are marked with a specific icon in the Model Browser.

Only one of the model implementations that have the same interface must be assigned to an application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In the Processing Resource Assignment table, remove all but one assignment from model implementations with the same interface to application processes.

Model port block: No assignment to application process

This conflict is caused by all of the following conditions:

- The model port block is resolved.
- The model port block is a part of the signal chain.
- The model implementation providing the model port block is not assigned to an application process.

The model port block is highlighted in the working view. During code generation, the model port block is treated like an unresolved model port block (refer to Characteristics of Model Port Blocks (ConfigurationDesk Real-Time Implementation Guide 🕮)).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Choose one of the following alternatives:

- Assign the model implementation to an application process.
- Create a preconfigured application for the model implementation.
- Delete the related model port block from the configuration.

Model port block: Unresolved port

A model port is obsolete.

An obsolete model port can occur, for example, if a model port is connected to an I/O function port and the model port is deleted from the model port block in the Simulink model. If you then perform a model analysis, the port is unresolved, but it is still displayed in ConfigurationDesk as long as it is connected to any other port. The obsolete model port is highlighted in the working view.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Delete the connection between the obsolete model port and the port it is connected to.

Model Port Mapping Conflicts

Model port mapping: Invalid connection between data port blocks of same model

Two model port blocks of the same model are mapped in the currently active ConfigurationDesk application.

Mappings between model port blocks must only be used for model communication between different models.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In a working view, modify the model port mapping.

Model port mapping: Invalid mapping to RTIxxxMM

The Configuration port of a CAN or LIN function block is mapped to a Configuration Port block that is generated for an RTICANMM ControllerSetup or RTILINMM ControllerSetup block. Additionally, the CAN or LIN function block specifies the hardware access for a bus configuration or an ECU Interface Configuration function block.

RTICANMM ControllerSetup and RTILINMM ControllerSetup blocks require exclusive access to CAN and LIN function blocks, i.e., the function blocks must not specify the hardware access for any other blocks or model implementations.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Choose one of the following alternatives:

- In a working view, delete the mapping line.
- In the Conflicts Viewer, select another suitable CAN or LIN function block from the list of the Assigned feature provider property. If no suitable function block is available, instantiate a new CAN or LIN function block.

  For basic information on specifying the hardware access for bus configurations, refer to Specifying the Hardware Access (ConfigurationDesk Bus Manager Implementation Guide 🕮).

Model port mapping: Mismatching data width

The model port mapping contains two ports with mismatching values for the Data width property.

Model port mappings can either be between a function port and a model port or between two model ports. The width of two mapped ports does not necessarily need to be identical. If the port widths differ, the intersecting vector elements are used. The other elements are neglected, and ConfigurationDesk issues a warning for the signal chain. Note that only scalars and vectors are supported. Multidimensional signals are not supported.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Choose one of the following alternatives:

- Adjust the function block settings to make the data width of the function port match the data width of the model port (not possible if two mapped model ports are affected).
- Adjust the behavior model in Simulink and reanalyze it.
- In a working view, modify the model port mapping.

Model port mapping: Mismatching port type

A function port is mapped to another function port, or the Configuration port of a function block is mapped to a model port not belonging to a Configuration Port block.

Function ports must not be mapped to other function ports. Configuration ports of function blocks must be mapped to Configuration Port blocks.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In a working view, modify the model port mapping.

Model port mapping: Mismatching protocol type

The Configuration port of a CAN, FlexRay, or LIN function block is mapped to a Configuration Port block that was created for a different bus protocol. For example, the Configuration port of a CAN function block is mapped to a Configuration Port block that was created for a FLEXRAYCONFIG UPDATE block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Modify the model port mapping.

Model port mapping: Multiple data sources connected

The inport of a model port block is mapped to two or more function ports or outports of a model port block.

The inport of a model port block must be mapped to exactly one function port or outport of a model port block.

Note

Multiple data sources are allowed for other sources.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Modify the model port mapping.

Model port mapping: Multiple links

The Configuration port of a FlexRay, CAN, or LIN function block is mapped to Configuration Port blocks in a way that is not supported.

For FlexRay function blocks: The Configuration port must be mapped to exactly one Configuration Port block of a Simulink model.

For CAN and LIN function blocks: The Configuration port must be mapped in one of the following ways:

- To exactly one Configuration Port block of one Simulink model or one V-ECU implementation.
- To one or more Configuration Port blocks of one or more bus simulation containers.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Modify the mapping to Configuration Port blocks to match the valid mappings as described above.

Model port mapping: No suitable configuration port connected

The Configuration port of an instantiated FlexRay function block is not mapped to a suitable Configuration Port block.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy In a working view, map the Configuration port to a suitable Configuration Port block.

Model port mapping: Unsupported combination of used models for model communication

The ConfigurationDesk application contains a precompiled Simulink implementation container (SIC file) created with the Model Interface Package for Simulink, or an SIC file created with TargetLink 4.4 that provides structured data ports. The structured data ports participate in model communication with a bus simulation container (BSC file) created with the Bus Manager of dSPACE Release 2019-A or earlier.

For model communication, this combination of model implementations is not supported.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Make sure that you only use supported model implementation combinations for model communication.

Model port mapping: Unsupported connection of variable-size data port

The model port with a variable-size signal has an invalid mapping line.

The mapping of variable-size model ports must comply with the following rules:

- Variable-size inports of model port blocks are mapped to variable-size outports or fixed-size outports of a model port block.

- The data width of the model inports must be ≥ the data width of the model outports.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Delete the unsupported mapping line from the variable-size model port and create a mapping line that complies with the above rules.

Model port mapping: Unsupported mapping of RTIxxxMM configuration to MicroAutoBox bus I/O function

The ConfigurationDesk application contains a Configuration Port block that was created for an RTICANMM ControllerSetup or RTILINMM ControllerSetup block. The configuration port of the Configuration Port block is mapped to a CAN or LIN function block to which a hardware resource is assigned that is provided by the MicroAutoBox III.

The MicroAutoBox III does not support bus communication that was configured using the RTI CAN MultiMessage Blockset or RTI LIN MultiMessage Blockset, respectively.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Choose one of the following alternatives:

- Assign a suitable SCALEXIO-based hardware resource to the CAN or LIN function block. For instructions, refer to How To Assign Hardware Resources Manually via Properties Browser (ConfigurationDesk Real-Time Implementation Guide 🕮).
- Map the configuration port of the Configuration Port block to a CAN or LIN function block to which a SCALEXIO-based hardware resource is assigned.

Multiplexed IPDU Conflicts

Multiplexed IPDU: Duplicate selector field code (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned multiplexed IPDU, the communication matrix specifies two or more dynamic part alternatives with identical selector field codes.

For each dynamic part alternative of a multiplexed IPDU, the communication matrix must specify a unique selector field code.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Duplicate selector field code (communication matrix)

A communication matrix specifies identical selector field codes for two or more dynamic part alternatives of a multiplexed IPDU.

For each dynamic part alternative of a multiplexed IPDU, the communication matrix must specify a unique selector field code.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Exceeding selector field and/or data segment positions (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned multiplexed IPDU, the communication matrix specifies the positions of the selector field or the data segments in such a way that at least one of them exceeds the boundaries of the multiplexed IPDU.

The communication matrix must specify the positions of the selector field and the data segments within the boundaries of a multiplexed IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Extend the length of the multiplexed IPDU: Specify a suitable value for the **Length** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

    Note

    Depending on the specified selector field and/or data segment positions, this remedy might not resolve the conflict.

  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Exceeding selector field and/or data segment positions (communication matrix)

A communication matrix specifies the positions of the selector field or the data segments of a multiplexed IPDU in such a way that at least one of them exceeds the boundaries of the multiplexed IPDU.

The communication matrix must specify the positions of the selector field and the data segments within the boundaries of a multiplexed IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Extend the length of the multiplexed IPDU: Specify a suitable value for the **Length** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

  Note

  Depending on the specified selector field and/or data segment positions, this remedy might not resolve the conflict.

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Invalid selector field length (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned multiplexed IPDU, the communication matrix specifies at least one of the following settings:

- For the multiplexed IPDU, no selector field is specified.
- A specified selector field has an invalid bit length.

For each multiplexed IPDU, the communication matrix must specify a selector field with a bit length in the range 1 … 8 bits.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Invalid selector field length (communication matrix)

A communication matrix contains at least one of the following settings:

- For a multiplexed IPDU, no selector field is specified.
- A specified selector field has an invalid bit length.

For each multiplexed IPDU, the communication matrix must specify a selector field with a bit length in the range 1 … 8 bits.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Invalid selector field code (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned multiplexed IPDU, the communication matrix specifies at least one of the following settings:

- For a dynamic part alternative of the multiplexed IPDU, no selector field code is specified.
- A specified selector field code exceeds the bit length of the related selector field.

For each dynamic part alternative of a multiplexed IPDU, the communication matrix must specify a selector field code. The selector field code must be smaller than the bit length of the related selector field:

$$selector\_field\_code_{max} \leq 2^{bit\_length\_selector\_field} - 1$$

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Invalid selector field code (communication matrix)

A communication matrix contains at least one of the following settings:

- For a dynamic part alternative of a multiplexed IPDU, no selector field code is specified.
- A specified selector field code exceeds the bit length of the related selector field.

For each dynamic part alternative of a multiplexed IPDU, the communication matrix must specify a selector field code. The selector field code must be smaller than the bit length of the related selector field:

$$selector\_field\_code_{max} \leq 2^{bit\_length\_selector\_field} - 1$$

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Invalid endianness of aggregated elements (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For the selector field and/or the data segment positions of the static or dynamic part of an assigned multiplexed IPDU, the communication matrix specifies at least one of the following settings:

- At least one start bit position is unspecified.
- At least one endianness (byte order) is unspecified or invalid, or the specified byte orders differ.

For the selector field and the data segment positions of the static and dynamic part of a multiplexed IPDU, the start bit positions and endiannesses must be specified. The endianness must be the same for all the elements. It must be either big endian or little endian.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Invalid endianness of aggregated elements (communication matrix)

A communication matrix specifies at least one of the following settings for the selector field and/or the data segment positions of the static or dynamic part of a multiplexed IPDU:

- At least one start bit position is unspecified.
- At least one endianness (byte order) is unspecified or invalid, or the specified byte orders differ.

For the selector field and the data segment positions of the static and dynamic part of a multiplexed IPDU, the start bit positions and endiannesses must be specified. The endianness must be the same for all the elements. It must be either big endian or little endian.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Invalid PDU-to-frame mapping references (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned multiplexed IPDU, the communication matrix specifies at least one of the following settings:

- For the dynamic part of the multiplexed IPDU, no PDU-to-frame mapping is specified.
- A PDU-to-frame mapping that is specified as the dynamic part of the multiplexed IPDU is not related to an IPDU in the communication matrix.

At least one PDU-to-frame mapping must be specified as the dynamic part of a multiplexed IPDU and the PDU-to-frame mapping must be related to an IPDU in the communication matrix.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Invalid PDU-to-frame mapping references (communication matrix)

A communication matrix specifies at least one of the following settings for a multiplexed IPDU:

- For the dynamic part of the multiplexed IPDU, no PDU-to-frame mapping is specified.
- A PDU-to-frame mapping that is specified as the dynamic part of the multiplexed IPDU is not related to an IPDU in the communication matrix.

At least one PDU-to-frame mapping must be specified as the dynamic part of a multiplexed IPDU and the PDU-to-frame mapping must be related to an IPDU in the communication matrix.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Overlapping selector field and/or data segment positions (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned multiplexed IPDU, the communication matrix specifies at least one of the following settings:

- The data segment positions in the multiplexed IPDU overlap.
- The selector field and the data segment positions of the static part overlap.

The communication matrix must specify the selector field and the data segment positions in such a way that they do not overlap each other within the multiplexed IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Multiplexed IPDU: Overlapping selector field and/or data segment positions (communication matrix)

A communication matrix specifies at least one of the following settings for a multiplexed IPDU:

- The data segment positions in the multiplexed IPDU overlap.
- The selector field and the data segment positions of the static part overlap.

The communication matrix must specify the selector field and the data segment positions in such a way that they do not overlap each other within the multiplexed IPDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

PDU Conflicts

PDU: Duplicate name (bus configuration)

Conflicting communication matrix elements are assigned to a bus configuration. For two or more assigned PDUs of the same ECU and direction (TX or RX), the communication matrix specifies identical names.

For PDUs of the same ECU and direction, the communication matrix must specify unique names.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

PDU: Duplicate name (communication matrix)

A communication matrix specifies identical names for two or more PDUs of the same ECU and direction (TX or RX).

For PDUs of the same ECU and direction, the communication matrix must specify unique names.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

PDU: Exceeding ISignals and/or update bits (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned PDU, the communication matrix specifies ISignal-to-IPDU mappings and due to these mappings, ISignals and/or update bits exceed the PDU boundaries.

The communication matrix must specify ISignal-to-IPDU mappings in such a way that ISignals and update bits are within the boundaries of the PDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Extend the length of the PDU and/or adjust the conflicting settings of its included ISignals: Specify suitable values for the **Length** property of the PDU and/or the **Length**, **Start bit position**, and/or **Endianness** properties of the conflicting ISignals, for example, in the **Conflicts Viewer**.

    This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

    Note

    If you change the length of an ISignal, you might have to adapt the ISignal's coded **Base data type**. For more information, refer to:
    - Bus Manager in ConfigurationDesk: Configurable Settings of ISignals (ConfigurationDesk Bus Manager Implementation Guide 🕮)
    - Bus Manager (stand-alone): Configurable Settings of ISignals (Bus Manager (Stand-Alone) Implementation Guide 🕮)

    Tip

    To access the coded **Base data type** property of an ISignal in the **Properties Browser**, right-click the ISignal in the **Conflicts Viewer** and select **Select Related Elements in Bus Configurations Table** from the context menu.

  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

PDU: Exceeding ISignals and/or update bits (communication matrix)

A communication matrix specifies ISignal-to-IPDU mappings for a PDU and due to these mappings, ISignals and/or update bits exceed the IPDU boundaries.

The communication matrix must specify ISignal-to-IPDU mappings in such a way that ISignals and update bits are within the boundaries of the PDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Extend the length of the PDU and/or adjust the conflicting settings of its included ISignals: Specify suitable values for the **Length** property of the PDU and/or the **Length**, **Start bit position**, and/or **Endianness** properties of the conflicting ISignals, for example, in the **Conflicts Viewer**.

  This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

  Note

  If you change the length of an ISignal, you might have to adapt the ISignal's coded **Base data type**. For more information, refer to:
  - Bus Manager in ConfigurationDesk: Configurable Settings of ISignals (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Configurable Settings of ISignals (Bus Manager (Stand-Alone) Implementation Guide 🕮)

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

PDU: Invalid payload length defined (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned PDU, the communication matrix specifies an invalid payload length.

The communication matrix must specify a valid payload length for each PDU. The valid value range depends on the bus system and bus protocol as follows:

- Classic CAN and LIN: 0 … 8 bytes
- CAN FD: 0 … 64 bytes
- J1939: 0 … 1785 bytes

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify a valid value for the **Length** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

PDU: Invalid payload length defined (communication matrix)

A communication matrix specifies an invalid payload length for a PDU.

The communication matrix must specify a valid payload length for each PDU. The valid value range depends on the bus system and bus protocol as follows:

- Classic CAN and LIN: 0 … 8 bytes
- CAN FD: 0 … 64 bytes
- J1939: 0 … 1785 bytes

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify a valid value for the **Length** property, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

PDU: No cyclic timing for configured TX PDU (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned CAN TX PDU, the communication matrix does not specify a cyclic transmission.

For CAN TX PDUs, the Bus Manager supports only cyclic transmission as the transmission mode specified in the communication matrix. Other transmission modes that are specified in the communication matrix are not supported.

For more information, refer to:

- Bus Manager in ConfigurationDesk: Limitations for CAN Communication (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Limitations for CAN Communication (Bus Manager (Stand-Alone) Implementation Guide 🕮)

If you do not resolve this conflict, default code is generated, i.e., because the communication matrix does not specify a cyclic transmission for the affected CAN TX PDU, the PDU is not transmitted at run time.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application. Choose one of the following alternatives:
  - Remove the conflicting element from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

    Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
  - Add cyclic timing elements to the PDU: Select the related PDU, for example, in the **Bus Configurations** table. In the **Properties Browser**, right-click the PDU and select **Add Element to Communication Matrix - Cyclic Timing** from the context menu. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged. Refer to:

    Bus Manager in ConfigurationDesk: Adding Cyclic Timing Elements to Basic PDUs (ConfigurationDesk Bus Manager Implementation Guide 🕮)

Bus Manager (stand-alone): Adding Cyclic Timing Elements to Basic PDUs (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Tip

The **Conflicts Viewer** displays the PDU triggering and not the PDU itself. To access the related PDU, right-click the PDU triggering and select **Select Related Elements in Bus Configurations Table** from the context menu.

- Add the **PDU Cyclic Timing Control**, **PDU Trigger**, or **Frame Access** feature to the PDU: Select the related PDU, for example, in the **Bus Configrations** table and add a feature via the context menu. Refer to:

  Bus Manager in ConfigurationDesk: Controlling the Cyclic Timing of CAN PDUs (ConfigurationDesk Bus Manager Implementation Guide 🕮), Specifying User-Defined Triggers for Transmitting PDUs (ConfigurationDesk Bus Manager Implementation Guide 🕮), or Accessing CAN Frame Settings (ConfigurationDesk Bus Manager Implementation Guide 🕮), respectively.

  Bus Manager (stand-alone): Controlling the Cyclic Timing of CAN PDUs (Bus Manager (Stand-Alone) Implementation Guide 🕮), Specifying User-Defined Triggers for Transmitting PDUs (Bus Manager (Stand-Alone) Implementation Guide 🕮), or Accessing CAN Frame Settings (Bus Manager (Stand-Alone) Implementation Guide 🕮), respectively.
- If the communication matrix specifies a global time domain for the PDU, assign the global time domain element to the bus configuration. Refer to:

  Bus Manager in ConfigurationDesk: Implementing Global Time Synchronization in Executable Applications (ConfigurationDesk Bus Manager Implementation Guide 🕮)

  Bus Manager (stand-alone): Implementing Global Time Synchronization in Executable Applications (Bus Manager (Stand-Alone) Implementation Guide 🕮)
- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

PDU: Overlapping ISignals and/or update bits (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned PDU, the communication matrix specifies ISignal-to-IPDU mappings and due to these mappings, ISignals and/or update bits overlap each other within the PDU.

The communication matrix must specify ISignal-to-IPDU mappings in such a way that ISignals and update bits do not overlap each other within the PDU.

If you do not resolve this conflict, default code is generated, i.e., the run-time behavior depends on the PDU direction:

- TX PDU: It is undefined which signal value is written last to the PDU bits that are affected by the overlapping ISignals and/or update bits. Therefore, the run-time behavior is undefined when the affected PDU is transmitted on the bus.
- RX PDU: All the signals that are available in a PDU are extracted separately. Therefore, the values of the PDU bits that are affected by the overlapping ISignals and/or update bits apply to each of the affected ISignals and update bits.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓		✓	–	✓	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application. Choose one of the following alternatives:
  - Remove the conflicting element from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

    Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
  - Specify suitable values for the **Length**, **Start bit position**, and/or **Endianness** properties of the conflicting ISignals, for example, in the **Conflicts Viewer**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

    Note

    If you change the length of an ISignal, you might have to adapt the ISignal's coded **Base data type**. For more information, refer to:
    - Bus Manager in ConfigurationDesk: Configurable Settings of ISignals (ConfigurationDesk Bus Manager Implementation Guide 🕮)
    - Bus Manager (stand-alone): Configurable Settings of ISignals (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Tip

To access the coded **Base data type** property of an ISignal in the **Properties Browser**, right-click the ISignal in the **Conflicts Viewer** and select **Select Related Elements in Bus Configurations Table** from the context menu.

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

PDU: Overlapping ISignals and/or update bits (communication matrix)

A communication matrix specifies ISignal-to-IPDU mappings for a PDU and due to these mappings, ISignals and/or update bits overlap each other within the PDU.

The communication matrix must specify ISignal-to-IPDU mappings in such a way that ISignals and update bits do not overlap each other within the PDU.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify suitable values for the **Length**, **Start bit position**, and/or **Endianness** properties of the conflicting ISignals, for example, in the **Conflicts Viewer**.

  This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

  Note

  If you change the length of an ISignal, you might have to adapt the ISignal's coded **Base data type**. For more information, refer to:
  - Bus Manager in ConfigurationDesk: Configurable Settings of ISignals (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Configurable Settings of ISignals (Bus Manager (Stand-Alone) Implementation Guide 🕮)

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

PDU Triggering Conflicts

PDU triggering: Invalid cyclic timing (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For the cyclic timing of an assigned PDU, the communication matrix specifies at least one of the following settings:

- No time period (cycle time) is specified or a specified time period is invalid.
- A specified time offset (delay time) is invalid.

For each cyclic timing of a PDU, the communication matrix must specify a time period in in the range 0 … 3600 seconds. If a time period of 0 is specified, the cyclic timing of the PDU is disabled and the PDU cannot be transmitted according to this cyclic timing. If the communication matrix specifies a time offset, it must also be in the range 0 … 3600 seconds.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Specify valid values for the Time Period and/or Time Offset properties of the PDU, for example, in the Conflicts Viewer. If the Conflicts Viewer does not display configurable properties, no time period is specified. In this case, add a Time Period element to the PDU via the Properties Browser. Alternatively, add the PDU Cyclic Timing Control, PDU Trigger, or Frame Access feature to the PDU. If the communication matrix specifies a global time domain for the PDU, you can also assign the global time domain element to the bus configuration. For more information, refer to:

Bus Manager in ConfigurationDesk: Configurable Settings of PDUs (ConfigurationDesk Bus Manager Implementation Guide 🕮), Controlling the Cyclic Timing of CAN PDUs (ConfigurationDesk Bus Manager Implementation Guide 🕮), Specifying User-Defined Triggers for Transmitting PDUs (ConfigurationDesk Bus Manager Implementation Guide 🕮), Accessing CAN Frame Settings (ConfigurationDesk Bus Manager Implementation Guide 🕮), or Implementing Global Time Synchronization in Executable Applications (ConfigurationDesk Bus Manager Implementation Guide 🕮) respectively.

Bus Manager (stand-alone): Configurable Settings of PDUs (Bus Manager (Stand-Alone) Implementation Guide 🕮), Controlling the Cyclic Timing of CAN PDUs (Bus Manager (Stand-Alone) Implementation Guide 🕮), Specifying User-Defined Triggers for Transmitting PDUs (Bus Manager (Stand-Alone) Implementation Guide 🕮), Accessing CAN Frame Settings (Bus Manager (Stand-Alone) Implementation Guide 🕮), or Implementing Global Time Synchronization in Executable Applications (Bus Manager (Stand-Alone) Implementation Guide 🕮), respectively.

Specifying valid values for the **Time Period** or **Time Offset** properties or adding a **Time Period** element to the PDU changes the communication matrix in the active ConfigurationDesk application and all PDU instances in all bus configurations of the application. The originally imported communication matrix remains unchanged. Adding a feature to the PDU or assigning a global time domain element changes only the related bus configuration. Other bus configurations and the communication matrix are not affected.

Tip

The **Conflicts Viewer** displays the PDU triggering and not the PDU itself. To access the related PDU, right-click the PDU triggering and select **Select Related Elements in Bus Configurations Table** from the context menu.

- Resolve the conflict by correcting the original communication matrix. Refer to:

  Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

  Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

PDU triggering: Invalid cyclic timing (communication matrix)

A communication matrix specifies a cyclic timing for a PDU with at least one of the following settings:

- No time period (cycle time) is specified or a specified time period is invalid.
- A specified time offset (delay time) is invalid.

For each cyclic timing of a PDU, the communication matrix must specify a time period in the range 0 … 3600 seconds. If a time period of 0 is specified, the cyclic timing of the PDU is disabled and the PDU cannot be transmitted

according to this cyclic timing. If the communication matrix specifies a time offset, it must also be in the range 0 … 3600 seconds.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application:

  Specify valid values for the **Time Period** and/or **Time Offset** properties of the PDU, for example, in the **Conflicts Viewer**. If the **Conflicts Viewer** does not display configurable properties, no time period is specified. In this case, add a **Time Period** element to the PDU via the **Properties Browser**. This changes the communication matrix in the active ConfigurationDesk application and all element instances in all bus configurations of the application. The originally imported communication matrix remains unchanged.

  For more information, refer to:

  - Bus Manager in ConfigurationDesk: Configurable Settings of PDUs (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Configurable Settings of PDUs (Bus Manager (Stand-Alone) Implementation Guide 🕮)

  Tip

  The **Conflicts Viewer** displays the PDU triggering and not the PDU itself. If the names of the PDU and the PDU triggering match, you can use the displayed PDU triggering name to search for the PDU via the **Find** command, for example.

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Processing Unit Application Conflicts

Processing unit application: Duplicate name

The executable application contains two or more processing unit applications with the same name.

The names of processing unit applications must be unique.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, enter unique names in the Name edit field for the processing unit applications.

Processing unit application: No application process defined

There is no application process defined in the processing unit application.

A processing unit application must have at least one application process.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Create a new application process by using one of the following commands:

- In the Model Browser: Create Preconfigured Application Process - In Existing Processing Unit Application - <Processing unit application> from the context menu of a model implementation
- In the Task Configuration table: New – Application Process from the context menu of the processing unit application

Processing unit application: No processing unit assigned

The processing unit application has no processing unit assigned.

Each processing unit application must have a processing unit assigned on which it is executed.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	–	–	–

Remedy In the Conflicts Viewer, for example, assign a processing unit to the processing unit application via the Assigned processing unit list.

Processing Unit Application: There are more application processes than available processor cores

The number of application processes in the processing unit application exceeds the number of processor cores available in the assigned processing unit.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	–	–

Remedy Choose one of the following alternatives:

- Assign a processing unit that has a sufficient number of processor cores to the processing unit application.
- Reduce the number of application processes in the processing unit application.

Processing Unit Conflicts

Processing unit: Multiple assignments to different processing unit applications

Several processing unit applications are assigned to the same processing unit.

Only one processing unit application is allowed per processing unit.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Processing Resource Assignment table, for example, assign excess processing unit applications to unassigned processing units.

Runnable Function Conflicts

Runnable function: Multiple assignments to different tasks

A runnable function is assigned to two or more tasks in an application process.

A runnable function must not be assigned to several tasks.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, select only one task in the Value column (next to the Assigned to field) to assign the runnable function to.

Runnable function: No task assignment

A component that is assigned to an application process provides a runnable function that is not assigned to a task. That means, that the runnable function is never called during simulation.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy In the Conflicts Viewer, select a task in the Value column (next to the Assigned to field) to assign the runnable function to.

Signal Chain Conflicts

Signal chain: Duplicate mapping line

In the signal chain, two ports are connected by a mapping line twice.

ConfigurationDesk does not support duplicate mappings.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, select one of the mapping lines and press `Del` to remove it.

Signal chain: Mismatching failure simulation settings

The failure simulation settings of a signal port and mapped device port(s) do not match for one of the following reasons:

- One or more failure class(es) that are set to Allowed at a signal port are not set to Allowed at a device port that is directly or indirectly mapped to it. This means that the failure class(es) are not available for the affected signal during failure simulation in your experiment software. For details, refer to Specifying Failure Simulation Settings (ConfigurationDesk Real-Time Implementation Guide 🕮).
- A load signal port is mapped to a device port of a Load device type and Load rejection is set to Not enforced at the signal port of the same function block, while Load rejection is set to Enforced at the mapped device port. For details, refer to Basics on Load Rejection (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, for example, change the failure simulation settings of the device and/or signal port(s) so that none of the conditions described above is met.
- If the conflict results from an erroneous device port mapping: In a working view, remove the mapping causing the conflict by using the context menu command Delete from Application.

Signal chain: Unresolved model port block

There are unresolved model port blocks in the currently active ConfigurationDesk application.

Unresolved model port blocks can result from the following actions:

- They were added to the signal chain via function blocks.
- They were removed from the behavior model linked to your active application and afterwards the interface of the behavior model was reanalyzed by ConfigurationDesk.
- The model port block is used in a multimodel application and resides with the same identity in two or more behavior models linked to your active ConfigurationDesk application. Additionally, two or more of these behavior models are assigned to an application process.

  In addition this unresolved model port block is displayed in the model topology with the icon:

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	–	–

Remedy Choose one of the following alternatives:

- To resolve unresolved model port blocks, you must generate and analyze the model interface. For details, refer to Transferring the ConfigurationDesk Model Interface to a New Simulink Interface Model (ConfigurationDesk Real-Time Implementation Guide 🕮).
- If you no longer need the unresolved model port blocks, you can remove them from the application. To do so, right-click the blocks in the Model Browser or in a working view and select Delete from Application from the context menu.

Task Conflicts

Task: Unfavorable bus monitoring configuration for an aperiodic task

Bus monitoring is activated for an asynchronous task.

Bus monitoring must not be activated for asynchronous tasks, but only for periodic tasks.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Deactivate bus monitoring for the asynchronous task. If required, activate it for a periodic task in the relevant application process.

Task: Invalid number of accepted overruns

An Any Error Hook event of the FlexRay function block is assigned to a task whose Number of accepted overruns property is set to a value other than -1.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy In the Conflicts Viewer, for example, enter -1 for the Number of accepted overruns property of the related task.

Task: Mismatching event period and runnable function cycle time restriction

A runnable function has a cycle time restriction (refer to Basics on Modeling Tasks in ConfigurationDesk (ConfigurationDesk Real-Time Implementation Guide 🕮)), and the event type triggering the corresponding task is one of the following:

- Case 1: A timer event with a mismatching Period value specified
- Case 2: An event other than a timer event, or a software event. In this case, ConfigurationDesk cannot detect if the event has the required properties.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy

- Case 1: In the Properties Browser, enter the value indicated for the cycle time restriction provided by the runnable function in the Period edit field of the timer event.
- Case 2: Assign a timer event or software event to the task.

Note

If you are sure that the task is triggered correctly by the assigned I/O event, you can ignore this conflict.

Task: Mismatching task priority and runnable function priority restriction

The executable application contains tasks assigned to an application process that do not correspond to the task priority restrictions provided by the runnable functions that are called by the tasks. For details, refer to Basics on Modeling Tasks in ConfigurationDesk (ConfigurationDesk Real-Time Implementation Guide 🕮).

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy In the Conflicts Viewer, for example, specify the task priorities in the Priority edit field according to the task priority restrictions provided by the runnable functions called by the tasks.

Task: Multiple events assigned

Two or more resolved events are assigned to the task. At least one event assigned to the task is not an I/O event.

If a task has multiple events assigned, all of the assigned events must be I/O events.

Note

This conflict does not apply to tasks provided by a V-ECU implementation.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Choose one of the following alternatives:

- If the task must be triggered by exactly one event, select the event that must trigger the task from the Assigned events list in the Conflicts Viewer.
- If the task must be triggered by multiple events, select suitable I/O events from the Assigned events list in the Conflicts Viewer.

Task: Multiple simulation time providers (Gigalink and Timer events) in an application process

There is a Gigalink event (I/O event from a Gigalink function block) assigned to a task and a timer event assigned to another task belonging to the same application process.

Each application process can have only one time provider (Gigalink event or timer event) in the tasks belonging to it.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, delete the event assigned to one of the conflicting tasks.
- Assign an event of the same type or an available asynchronous event to one of the conflicting tasks.

Task: Multiple Tasks triggered by Gigalink events in an application process

Two or more tasks in an application process have a Gigalink event (I/O event from a Gigalink function block) assigned to them.

An application process can have only one task with assigned Gigalink events.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
✓	–	–	–	✓	–	–

Remedy Choose one of the following alternatives:

- In the Conflicts Viewer, delete the Gigalink event assigned to one of the conflicting tasks.
- Assign another of the available events to one of the conflicting tasks.

Task: No event assigned

There is a task in an application process that has no event assigned. The task is not triggered during real-time simulation.

Note

This conflict does not apply to tasks provided by a V-ECU implementation.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Choose one of the following alternatives:

- In the Properties Browser, assign an existing event to the task.
- Create a new timer event for the task.

Task: No runnable function assigned

There is a task in an application process that does not have any assigned runnable function. If the task is triggered during simulation, it does not call any runnable functions.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy In the Properties Browser, assign a runnable function to the task.

Task: Unfavorable runnable function execution order

The execution order of the runnable functions in a task in a multimodel application process might be unfavorable with regard to the interconnection of the models that provide the relevant runnable functions.

For example, if model A sends data to model B, the runnable functions provided by model A should be executed before the runnable functions provided by model B.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy To resolve this conflict, choose one of the following options:

- From the context menu of the relevant application process, select the Optimize Configuration command.
- In the Conflicts Viewer, specify a suitable runnable function execution order via the Execution Order property of the related runnable functions.

Task: Unresolved event

There is a task within the application process triggered by an event that is not provided by a component of the application process. This means that the event is unresolved. The event is ignored during code generation.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Make sure that the component providing the event is assigned to the application process. For example, in the Executable Application table, assign the model providing the event to the application process.

Task: Unresolved runnable function

In the application process, a task has a runnable function assigned that is not provided by a component of the application process. This means that the runnable function is unresolved.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy Make sure that the component providing the runnable function is assigned to the application process. For example, in the Executable Application table, assign the model providing the runnable function to the application process.

Task Group Conflicts

Task group: Invalid priority

The synchronization task priority of a FlexRay function block has a value > 0.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	–	–	✓	–

Remedy In the Conflicts Viewer, for example, change the priority value.

Time Master Conflicts

Time master: Invalid property values resulting in default code (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned time master, the communication matrix specifies at least one of the following settings:

- For the Sync period property, no value is specified.
- For the Follow-up offset property, no value is specified, while no debounce time is specified for the related global time domain.
- The Sync period and the Follow-up offset properties are specified but the specified Sync period value is smaller than the Follow-up offset value.

For each time master, the communication matrix must specify values for the Sync period and Follow-up offset properties as follows:

```
<value of Sync period> ≥ <value of Follow-up offset>
```

If you do not resolve this conflict, default code is generated, i.e., the run-time behavior depends on the following:

- No Sync period value specified: No time synchronization messages are transmitted, i.e, no synchronization (SYNC) messages and no follow-up (FUP) messages.
- No Follow-up offset value and no debounce time specified: Follow-up (FUP) messages are transmitted immediately after their related synchronization (SYNC) messages.
- Sync period value smaller than Follow-up offset value: The transmission of synchronization (SYNC) messages is delayed until the transmission of the previous follow-up (FUP) messages is scheduled, e.g., after the specified Follow-up offset has elapsed.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Resolve the conflict only for the active ConfigurationDesk application. Choose one of the following alternatives:
  - Remove the conflicting element from the bus configuration: In the **Conflicts Viewer**, set the element's **Is assigned** property to **False** to remove only this conflicting element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

    Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
  - Add the **GTS Transmission Control** feature to the global time domain: Select the related global time domain in the **Bus Configurations** table and add the feature via the context menu.

    Note

    This remedy does not resolve the conflict if the conflict cause is an unspecified **Follow-up offset** value.

    Refer to:

    Bus Manager in ConfigurationDesk: Controlling the Timing of Time Synchronization (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Controlling the Timing of Time Synchronization (Bus Manager (Stand-Alone) Implementation Guide 🕮)

    Tip

    The **Conflicts Viewer** displays the time master and not the global time domain itself. To access the related global time domain, right-click the time master and select **Select Related Elements in Bus Configurations Table** from the context menu.

- Resolve the conflict by correcting the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Time master: Invalid property values resulting in default code (communication matrix)

A communication matrix specifies at least one of the following settings for a time master:

- For the Sync period property, no value is specified.
- For the Follow-up offset property, no value is specified, while no debounce time is specified for the related global time domain.
- The Sync period and the Follow-up offset properties are specified but the specified Sync period value is smaller than the Follow-up offset value.

For each time master, the communication matrix must specify values for the Sync period and Follow-up offset properties as follows:

```
<value of Sync period> ≥ <value of Follow-up offset>
```

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Time master: Invalid property values resulting in no code (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned time master, the communication matrix specifies invalid values for at least one of the following properties:

- Sync period
- Follow-up offset

For each time master, the communication matrix must specify the values of the Sync period and Follow-up offset properties in the following range:

- Sync period: 0 … 4200 seconds
- Follow-up offset: 0 … < 4.0 seconds

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting

element. Alternatively, set the bus configuration's **Is assigned** property to **False** to remove all conflicting elements in one step.

Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, choose one of the following alternatives:
  - Resolve the conflict only for the active ConfigurationDesk application:

    Add the **GTS Transmission Control** feature to the related global time domain.

    Note

    This remedy does not resolve the conflict if the conflict cause is an invalid **Follow-up offset** value.

    Refer to:

    Bus Manager in ConfigurationDesk: Controlling the Timing of Time Synchronization (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Controlling the Timing of Time Synchronization (Bus Manager (Stand-Alone) Implementation Guide 🕮)

    Tip

    The **Conflicts Viewer** displays the time master and not the global time domain itself. To access the related global time domain, right-click the time master and select **Select Related Elements in Bus Configurations Table** from the context menu.

  - Resolve the conflict by correcting the original communication matrix. Refer to:

    Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)

    Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Time master: Invalid property values resulting in no code (communication matrix)

A communication matrix specifies invalid values for at least one of the following time master properties:

- **Sync period**
- **Follow-up offset**

For each time master, the communication matrix must specify the values of the **Sync period** and **Follow-up offset** properties in the following range:

- **Sync period**: 0 … 4200 seconds
- **Follow-up offset**: 0 … < 4.0 seconds

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Time Slave Conflicts

Time slave: Invalid property values (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned time slave, the communication matrix specifies invalid values for at least one of the following properties:

- Follow-up timeout value
- Maximum counter jump width

For each time slave, the communication matrix must specify values for the Follow-up timeout value and Maximum counter jump width properties in the following range:

- Follow-up timeout value: 0 … 4.32 seconds
- Maximum counter jump width: 1 … 15

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	✓	–	–	✓	–

Remedy Choose one of the following alternatives:

- If you do not need the conflicting element in the executable application, remove it from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.

- If you need the conflicting element in the executable application, correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Time slave: Invalid property values (communication matrix)

A communication matrix specifies invalid values for at least one of the following time slave properties:

- Follow-up timeout value
- Maximum counter jump width

For each time slave, the communication matrix must specify values for the Follow-up timeout value and Maximum counter jump width properties in the following range:

- Follow-up timeout value: 0 … 4.32 seconds
- Maximum counter jump width: 1 … 15

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Time slave: Missing property values (bus configuration)

A conflicting communication matrix element is assigned to a bus configuration. For an assigned time slave, the communication matrix does not specify values for the Follow-up timeout value and/or Maximum counter jump width properties.

For each time slave, the communication matrix must specify values for the Follow-up timeout value and Maximum counter jump width properties.

If you do not resolve this conflict, default code is generated, i.e., the Bus Manager uses the following values:

- Follow-up timeout value: 4 seconds
- Maximum counter jump width: 15

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	✓	–	✓	–	✓	–

Remedy Choose one of the following alternatives:

- Remove the conflicting element from the bus configuration: In the Conflicts Viewer, set the element's Is assigned property to False to remove only this conflicting element. Alternatively, set the bus configuration's Is assigned property to False to remove all conflicting elements in one step.

  Depending on the type of the conflicting element, related elements are removed as well, e.g., all ISignals of a conflicting ISignal IPDU.
- Correct the setting in the original communication matrix. Refer to:
  - Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
  - Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)

Time slave: Missing property values (communication matrix)

A communication matrix does not specify values for the Follow-up timeout value and/or Maximum counter jump width properties.

For each time slave, the communication matrix must specify values for the Follow-up timeout value and Maximum counter jump width properties.

Effects

Abort Build	Warning During Build	Generate No Code	Generate Default Code	Abort BSC File Generation	Warning During BSC File Gen.	Abort XLSX Export
–	–	–	–	–	–	–

Remedy Correct the setting in the original communication matrix. Refer to:

- Bus Manager in ConfigurationDesk: Resolving Bus Manager-Related Conflicts (ConfigurationDesk Bus Manager Implementation Guide 🕮)
- Bus Manager (stand-alone): Resolving Bus Manager-Related Conflicts (Bus Manager (Stand-Alone) Implementation Guide 🕮)