Vector Legacy Converter

Technical Reference

Technical Documentation

Version 1.4.2

Authors	Thorsten Fröhlinghaus
Status	Released

Document Information

History

Author	Date	Version	Remarks
Thorsten Fröhlinghaus	2011-04-07	1.0	Created
Thorsten Fröhlinghaus	2011-04-07	1.0	Released
Thorsten Fröhlinghaus	2011-12-20	1.1	Legacy Converter V1.3.0
Thorsten Fröhlinghaus	2012-01-20	1.1	Released
Thorsten Fröhlinghaus	2012-03-22	1.2	Released
Thorsten Fröhlinghaus	2012-11-21	1.3	Released
Thorsten Fröhlinghaus	2014-04-01	1.4.1	Released
Thorsten Fröhlinghaus	2014-07-28	1.4.2	Released

Reference Documents

No.	Source	Title	Version
[1]	Autosar	Specification of the System Template, R3.1 Rev 4	V3.2.0
[2]	Autosar	Specification of the System Template, R3.2 Rev 1	V3.4.0
[3]	Autosar	System Template, R4.0 Rev 3	V4.2.0

Caution
We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector´s release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

Contents

Tables

1 Introduction

The Vector Legacy Converter (VLC) supports the migration of legacy embedded software to the AUTOSAR software architecture. The VLC is a console application which transforms one or more DBC-, LDF- and Fibex files into an AUTOSAR System Description and its ECU Extracts. The VLC is typically called by the DaVinci Project Assistant (DPA), but it can also be used as a stand-alone tool. The resulting ECU Extracts will serve as input for the Initial EcuC Generator.

2 Functional Description

The VLC analyses legacy communication databases, and it maps their communication elements to AUTOSAR System Description elements. There are no standards or established rules which define such a mapping between legacy communication databases and System Descriptions. For this reason, the VLC defines its own AUTOSAR mapping rules which aim at preserving the semantics of the original communication databases. These generic rules may be supplemented with OEM-specific rules.

The AUTOSAR System Description allows various modeling variants w.r.t. to, e.g., naming conventions and package structures. The VLC imposes fixed modeling rules which define a common namespace for DBC-, LDF- and Fibex transformations. The VLC modeling rules are not coordinated with the AUTOSAR transformations of other tools from other vendors. The transformation results of the VLC and other tools may appear rather different.

The VLC supports no user interaction, and thus the transformation between legacy formats and AUTOSAR System Descriptions is always the same. However, the VLC still identifies the manufacturer of a communication database, and it applies OEM-specific rules. These OEM-specific rules must be implemented in advance.

The calling conventions and options of the VLC are specified with the following help text:

```
Usage: LegacyDb2SystemDescrConverter [options] <file|dir> [<file|dir> ...]
[extfile]

Create an AUTOSAR System Description out of one or more DBC, LDF or Fibex
communication databases.

Options:
  -h, --help               Show this help
  -a, --adoptname          Adopt DBC filename as cluster name
  -e, --extract            Create ECU extracts
  -r, --release <31|32|40> AUTOSAR release
  -o, --output <file|dir>  Output file or directory

Parameters:
  <file|dir>               Input file (*.dbc, *.ldf, *.xml) or directory
  extfile                  Extension file (*.vsde)
```

Table 2-1   Vector Legacy Converter help text.

The options may also be specified in a file called LegacyDb2SystemDescrConverter.config which resides in the same directory as the exe file LegacyDb2SystemDescrConverter.exe. In this way, options can be defined which are processed, e.g., when calling the VLC from the DPA.

With the option '-r' or '--release' the AUTOSAR schema version can be selected. Currently, AUTOSAR 3.1.4, AUTOSAR 3.2.1 and AUTOSAR 4.0.3 are supported. The VLC implements those schema versions which are required by the Vector tool chain, especially by the Initial EcuC Generator. The VLC does not aim at supporting arbitrary AUTOSAR schema versions.

Please note that the DBC-, LDF- and Fibex transformations are rather sophisticated, and thus we can only provide a general survey with this document. To identify the AUTOSAR mapping more in detail, the user may, e.g., perform minor changes to a communication database, and then compare the transformation results before and after these changes. The VLC defines a fixed order for all AUTOSAR elements, so two System Descriptions can be easily diffed.

2.1 DBC Transformation

The DBC file format is based on a network-specific object model and on user-defined attributes. The former can be transformed in a generic way to AUTOSAR, while the latter often require an OEM-specific transformation. The table below shows how CAN network objects are mapped to AUTOSAR 3.1.4 or AUTOSAR 3.2.1 elements.

CAN network object	AUTOSAR element

Signal	SystemSignal
	  ShortName = Signal.Name
	  Length = Signal.Bitcount
	  BooleanType|IntegerType|RealType
	    ShortName = "DT_" + Signal.Name
	    LowerLimit = (Signal.Min-Signal.Offset)/Signal.Factor
	    UpperLimit = (Signal.Max-Signal.Offset)/Signal.Factor
	    CompuMethod
	      ShortName = "CM_" + Signal.Name
	      Unit
	        ShortName = "U_" + Signal.Unit
	      CompuInternalToPhys.CompuScale
	        LowerLimit = Signal.TextualEncoding.LowerBound
	        UpperLimit = Signal.TextualEncoding.UpperBound
	        CompuConst = "Cx\<Limit\>_" + Signal.TextalEncoding.Text
	      CompuInternalToPhys.CompuScale.CompuRationalCoeffs
	        CompuNumerator = Signal.Offset, Signal.Factor

SignalGroup	SystemSignalGroup ShortName = "SG_" + SignalGroup.Name

CANBus	CanCluster
	  ShortName = CANBus.Attributes.DBName
	  ProtocolName = "CAN"
	  PhysicalChannel
	    ShortName = "CHNL"

CANBus
 .CANFrame
	Frame
	  ShortName = CANFrame.Name + "__" + CanCluster.ShortName
	  FrameLength = CANFrame.DLC
	SignalIPdu|MultiplexedIPdu|DcmIPdu|NmPdu|NPdu
	  ShortName = CANFrame.Name + "__" + CanCluster.ShortName
	  Length = 8*CANFrame.DLC
	Frame.PduToFrameMapping
	  ShortName = CANFrame.Name
	  PackingByteOrder = Intel
	  StartPosition = 0
	PhysicalChannel.CanFrameTriggering
	  ShortName = "FT_" + CANFrame.Name
	  Identifier = CANFrame.ID
	PhysicalChannel.IPduTriggering
	  ShortName = "PT_" + CANFrame.Name

CANBus
 .CANFrame
 .MappedSignal
 .Signal
	ISignal
	  ShortName = Signal.Name + "__" + CANFrame.Name
	                        + "__" + CanCluster.ShortName
	SignalIPdu.ISignalToIPduMapping
	  ShortName = Signal.Name
	  PackingByteOrder = MappedSignal.Intel|Motorola
	  StartPosition = MappedSignal.Startbit
	PhysicalChannel.SignalTriggering
	  ShortName = "ST_" + Signal.Name + "__" + CANFrame.Name

CANBus
 .CANFrame
 .MappedMultiplexorSignal
 .MultiplexorSignal
	MultiplexedIPdu.SelectorField
	  ByteOrder = MappedMultiplexorSignal.Intel|Motorola
	  Length = MultiplexorSignal.Bitcount
	  StartPosition = MappedMultiplexorSignal.Startbit

CANBus
 .CANFrame
 .MappedMultiplexorSignal
 .MappedMultiplexedSignal
 .MultiplexedSignal
	MultiplexedIPdu.DynamicPart.DynamicPartAlternative
	  SelectorFieldCode = MappedMultiplexedSignal.MultiplexorValue
	  SignalIPdu
	    ShortName = CANFrame.Name + "_Mx\<Code\>__" + CanCluster.ShortName
	    Length = 8*CANFrame.DLC
	  PhysicalChannel.IPduTriggering

based on template version 4.11.1

	ShortName = "PT_" + CANFrame.Name + "_Mx<Code>"

CANNode	EcuInstance
	  ShortName = CANNode.Name
	  ComProcessingPeriod = 0.001
	EcuInstance.CanCommunicationController
	  ShortName = "CT_" + CanCluster.ShortName
	EcuInstance.CommunicationConnector
	  ShortName = "CN_" + CanCluster.ShortName
	EcuInstance.AssociatedIPduGroup(Rx)
	  ShortName = CANNode.Name + "__" + CanCluster.ShortName + "_Rx"
	EcuInstance.AssociatedIPduGroup(Tx)
	  ShortName = CANNode.Name + "__" + CanCluster.ShortName + "_Tx"

CANNode
 .RxCANFrame
	CommunicationConnector.FramePort
	  ShortName = "FP_" + RxCANFrame.Name + "_Rx"
	  Direction = In
	CommunicationConnector.IPduPort
	  ShortName = "PP_" + RxCANFrame.Name + "_Rx"
	  Direction = In

CANNode
 .RxCANFrame
 .MappedSignal
 .Signal
	CommunicationConnector.SignalPort
	  ShortName = "SP_" + Signal.Name + "__" + RxCANFrame.Name + "_Rx"
	  Direction = In

CANNode
 .TxCANFrame
	CommunicationConnector.FramePort
	  ShortName = "FP_" + TxCANFrame.Name + "_Tx"
	  Direction = Out
	CommunicationConnector.IPduPort
	  ShortName = "PP_" + TxCANFrame.Name + "_Tx"
	  Direction = Out

CANNode
 .TxCANFrame
 .MappedSignal
 .Signal
	CommunicationConnector.SignalPort
	  ShortName = "SP_" + Signal.Name + "__" + TxCANFrame.Name + "_Tx"
	  Direction = Out

Table 2-2   Transformation of CAN network objects.

The subsequent table shows how user-defined attributes are processed by the AUTOSAR transformation.

Attribute	Autosar element

CANNode.NmAsrCanMsgReducedTime	CanNmMsgReducedTime = CANNode.NmAsrCanMsgReducedTime

Table 2-3   Transformation of user-defined attributes.

In addition to the transformation steps described in the tables above, the following rules hold for the DBC transformation.

> For SignalGroups the SignalIPdu.ISignalToIPduMapping.TransferProperty is always Pending.

> The attribute CANBus.ILTxTimeout is not processed.

2.2   LDF Transformation

The LDF file format does not provide any user-defined attributes or other OEM extensions. The subsequent table shows how LIN network objects are transformed in a generic way to AUTOSAR 3.1.4 or AUTOSAR 3.2.1 elements.

LIN network object	AUTOSAR element

`Signal`	`SystemSignal`
	`  ShortName = Signal.Name`
	`  Length   = Signal.Bitcount`
	`  BooleanType|IntegerType`
	`    ShortName  = "DT_" + Signal.Name`
	`    LowerLimit = (Signal.Min-Signal.Offset)/Signal.Factor`
	`    UpperLimit = (Signal.Max-Signal.Offset)/Signal.Factor`
	`    CompuMethod`
	`      ShortName = "CM_" + Signal.Name`
	`      Unit`
	`        ShortName = "U_" + Signal.Unit`
	`      CompuInternalToPhys.CompuScale`
	`        LowerLimit = Signal.TextualEncoding.LowerBound`
	`        UpperLimit = Signal.TextualEncoding.UpperBound`
	`        CompuConst = "Cx<Limit>_" + Signal.TextalEncoding.Text`
	`      CompuInternalToPhys.CompuScale.CompuRationalCoeffs`
	`        CompuNumerator = Signal.Offset, Signal.Factor`
	`  ConstantSpecification`
	`    ShortName = "C_" + Signal.Name`
	`    BooleanLiteral|IntegerLiteral`
	`      ShortName = "C_" + Signal.Name`
	`      Value    = Signal.LDFSignalSpecialValues`

`LINBus`	`LinCluster`
	`  ShortName      = LINBus.LINChannelPostfix`
	`  ProtocolName    = "LIN"`
	`  ProtocolVersion = LINBus.ProtocolVersion`
	`  Speed          = LINBus.BaudRate`
	`  PhysicalChannel`
	`    ShortName = "CHNL"`

`LINBus`
` .LINFrame(Unconditional)`
	`Frame`
	`  ShortName  = LINFrame.Name + "__" + LinCluster.ShortName`
	`  FrameLength = LINFrame.Size`
	`SignalIPdu|DcmIPdu|NPdu`
	`  ShortName = LINFrame.Name + "__" + LinCluster.ShortName`
	`  Length   = 8*LINFrame.Size`
	`Frame.PduToFrameMapping`
	`  ShortName       = LINFrame.Name`
	`  PackingByteOrder = LINBus.ByteOrder`
	`  StartPosition   = 0(Intel)|7(Motorola)`
	`PhysicalChannel.LinFrameTriggering`
	`  ShortName   = "FT_" + LINFrame.Name`
	`  ChecksumType = LINFrame.CSModel`
	`  Identifier  = LINFrame.ID`
	`PhysicalChannel.IPduTriggering`
	`  ShortName  = "PT_" + LINFrame.Name`

`LINBus`
` .LINFrame(Unconditional)`
` .MappedSignal`
` .Signal`
	`ISignal`
	`  ShortName = Signal.Name + "__" + LINFrame.Name`
	`                         + "__" + LinCluster.ShortName`
	`SignalIPdu.ISignalToIPduMapping`
	`  ShortName       = Signal.Name`
	`  PackingByteOrder = MappedSignal.Intel|Motorola`
	`  StartPosition   = MappedSignal.Startbit`
	`PhysicalChannel.SignalTriggering`
	`  ShortName = "ST_" + Signal.Name + "__" + LINFrame.Name`

`LINBus`
` .LINFrame(Sporadic`
`        |EventTriggered)`
	`SubstitutionFrame`
	`  ShortName       = LINFrame.Name + "__" + LinCluster.ShortName`
	`  FrameLength     = LINFrame.Size`
	`  SubstitutionType = Sporadic|EventTriggered`
	`PhysicalChannel.LinFrameTriggering`
	`  ShortName   = "FT_" + LINFrame.Name`
	`  ChecksumType = LINFrame.CSModel`

based on template version 4.11.1

	`Identifier   = LINFrame.ID`

`LINNode(Master|Slave)`	`EcuInstance`
	`  ShortName = LINNode.Name`
	`EcuInstance.LinMaster`
	`  ShortName     = "CT_" + LinCluster.ShortName`
	`  TimeBase      = LINNode.Timebase`
	`  TimeBaseJitter = LINNode.Jitter`
	`EcuInstance.LinSlave`
	`  ShortName        = "CT_" + LinCluster.ShortName`
	`  ConfiguredNad    = LINSlaveNode.ConfiguredNad`
	`  LinErrorResponse = LINSlaveNode.ResponseErrorSignal`
	`  ProtocolVersion  = LINSlaveNode.ProtocolVersion`
	`EcuInstance.CommunicationConnector`
	`  ShortName = "CN_" + LinCluster.ShortName`
	`EcuInstance.AssociatedIPduGroup(Rx)`
	`  ShortName = LINNode.Name + "__" + LinCluster.ShortName + "_Rx"`
	`EcuInstance.AssociatedIPduGroup(Tx)`
	`  ShortName = LINNode.Name + "__" + LinCluster.ShortName + "_Tx"`

`LINNode`
` .RxLINFrame`
	`CommunicationConnector.FramePort`
	`  ShortName = "FP_" + RxLINFrame.Name + "_Rx"`
	`  Direction = In`
	`CommunicationConnector.IPduPort`
	`  ShortName = "PP_" + RxLINFrame.Name + "_Rx"`
	`  Direction = In`

`LINNode`
` .RxLINFrame`
` .MappedSignal`
` .Signal`
	`CommunicationConnector.SignalPort`
	`  ShortName = "SP_" + Signal.Name + "__" + RxLINFrame.Name + "_Rx"`
	`  Direction = In`

`LINNode`
` .TxLINFrame`
	`CommunicationConnector.FramePort`
	`  ShortName = "FP_" + TxLINFrame.Name + "_Tx"`
	`  Direction = Out`
	`CommunicationConnector.IPduPort`
	`  ShortName = "PP_" + TxLINFrame.Name + "_Tx"`
	`  Direction = Out`

`LINNode`
` .TxLINFrame`
` .MappedSignal`
` .Signal`
	`CommunicationConnector.SignalPort`
	`  ShortName = "SP_" + Signal.Name + "__" + TxLINFrame.Name + "_Tx"`
	`  Direction = Out`

`LINBus`
` .LINScheduleTable`
	`LinCluster.LinScheduleTable`
	`  ShortName = LINScheduleTable.Name`
	`  Priority  = 255`
	`  RunMode   = RunContinuous`

`LINBus`
` .LINScheduleTable`
` .UnconditionalFrameSlot`
	`LinFrameTriggering.RelativelyScheduledTiming`
	`  Delay         = UnconditionalFrameSlot.SlotDelay`
	`  PositionInTable = UnconditionalFrameSlot.ID`

`LINBus`
` .LINScheduleTable`
` .DiagnosticFrameSlot`
	`LinFrameTriggering.RelativelyScheduledTiming`
	`  Delay         = DiagnosticFrameSlot.SlotDelay`
	`  PositionInTable = DiagnosticFrameSlot.ID`

`LINBus`
` .LINScheduleTable`
` .AssignFrameIdSlot`
	`LinFrameTriggering.AssignFrameIdTiming`
	`  Delay         = AssignFrameIdSlot.SlotDelay`
	`  PositionInTable = AssignFrameIdSlot.ID`
	`  AssignedFrameTriggering`
	`    ShortName = "FT_" + AssignFrameIdSlot.FrameToAssign.Name`

`LINBus`
` .LINScheduleTable`
` .UnassignFrameIdSlot`
	`LinFrameTriggering.UnassignFrameIdTiming`
	`  Delay         = UnassignFrameIdSlot.SlotDelay`
	`  PositionInTable = UnassignFrameIdSlot.ID`
	`  UnassignedFrameTriggering`
	`    ShortName = "FT_" + UnassignFrameIdSlot.FrameToUnassign.Name`

`LINBus`
` .LINScheduleTable`
` .AssignNADSlot`
	`LinFrameTriggering.AssignNADTiming`
	`  Delay         = AssignNADSlot.SlotDelay`
	`  PositionInTable = AssignNADSlot.ID`
	`  NewNAD        = AssignNADSlot.NewNAD`

`LINBus`
` .LINScheduleTable`
` .ConditionalChangeNADSlot`
	`LinFrameTriggering.DataTiming`
	`  Delay             = ConditionalChangeNADSlot.SlotDelay`
	`  PositionInTable     = ConditionalChangeNADSlot.ID`
	`  FreeFormatByteValues = ConditionalChangeNADSlot.DataBytes`

LINBus
.LINScheduleTable
.FreeFormatSlot
	LinFrameTriggering.DataTiming
	  Delay = FreeFormatSlot.SlotDelay
	  PositionInTable = FreeFormatSlot.ID
	  FreeFormatByteValues = FreeFormatSlot.DataBytes

LINBus
.LINScheduleTable
.EventTriggeredFrameSlot
	LinFrameTriggering.RelativelyScheduledTiming
	  Delay = EventTriggeredFrameSlot.SlotDelay
	  PositionInTable = EventTriggeredFrameSlot.ID

LINBus
.LINScheduleTable
.SporadicFrameSlot
	LinFrameTriggering.RelativelyScheduledTiming
	  Delay = SporadicFrameSlot.SlotDelay
	  PositionInTable = SporadicFrameSlot.ID

LINBus
.LINScheduleTable
.DataDumpSlot
	LinFrameTriggering.DataTiming
	  Delay = DataDumpSlot.SlotDelay
	  PositionInTable = DataDumpSlot.ID
	  FreeFormatByteValues = DataDumpSlot.DataBytes

LINBus
.LINScheduleTable
.AssignFrameIdRangeSlot
	LinFrameTriggering.DataTiming
	  Delay = AssignFrameIdRangeSlot.SlotDelay
	  PositionInTable = AssignFrameIdRangeSlot.ID
	  FreeFormatByteValues = AssignFrameIdRangeSlot.DataBytes

LINBus
.LINScheduleTable
.SaveConfigurationSlot
	LinFrameTriggering.DataTiming
	  Delay = SaveConfigurationSlot.SlotDelay
	  PositionInTable = SaveConfigurationSlot.ID
	  FreeFormatByteValues = SaveConfigurationSlot.DataBytes

Table 2-4   Transformation of LIN network objects.

2.3   Fibex Transformation

The VLC supports the transformation of Fibex 2.0.1 files and of Fibex 3.0.0 or 3.1.0 files. The main difference between these Fibex versions is the modeling of PDUs. In Fibex 2.0.1 PDUs are modeled with signal groups, while in Fibex 3.0.0 and 3.1.0 PDUs are an explicit part of the XML schema. The table below shows how Fibex 2.0.1 elements are mapped to AUTOSAR 3.1.4 or AUTOSAR 3.2.1 elements.

Fibex element	AUTOSAR element
PhysicalDimension	PhysicalDimesion 　ShortName　　　　　　= PhysicalDimension.ShortName 　LengthExp　　　　　　= PhysicalDimension.LengthExp 　MassExp　　　　　　　= PhysicalDimension.MassExp 　TimeExp　　　　　　　= PhysicalDimension.TimeExp 　CurrentExp　　　　　= PhysicalDimension.CurrentExp 　TemperatureExp　　　= PhysicalDimension.TemperatureExp 　MolarAmoutExp　　　= PhysicalDimension.MolarAmoutExp 　LuminousIntensityExp = PhysicalDimension.LuminousIntensityExp
Unit	Unit 　ShortName　　　= Unit.ShortName 　DisplayName　　= Unit.DisplayName 　FactorSiToUnit = Unit.FactorSiToUnit 　OffsetSiToUnit = Unit.OffsetSiToUnit
Coding	BooleanType|OpaqueType|IntegerType|RealType|CharType|StringType 　ShortName　= Coding.ShortName 　LowerLimit = (Coding.CompuMethod.PhysConstr.LowerLimit 　　　　　　　　- Coding.CompuMethod.CompuRationalCoeffs[0]) 　　　　　　　　/ Coding.CompuMethod.CompuRationalCoeffs[1] 　UpperLimit = (Coding.CompuMethod.PhysConstr.UpperLimit 　　　　　　　　- Coding.CompuMethod.CompuRationalCoeffs[0]) 　　　　　　　　/ Coding.CompuMethod.CompuRationalCoeffs[1] 　InvalidValue.BooleanLiteral|OpaqueLiteral|IntegerLiteral 　　　　　　　　　|RealLiteral|CharLiteral|StringLiteral 　　ShortName = Coding.ShortName 　　Value　　= Coding.CompuMethod.InternalConstr.LowerLimit
Coding 　.CompuMethod	CompuMethod 　ShortName = "CM_" + CompuMethod.ShortName 　CompuInternalToPhys.CompuScale 　　LowerLimit = CompuMethod.CompuScale.LowerLimit 　　UpperLimit = CompuMethod.CompuScale.UpperLimit 　　CompuConst = "Cx<Limit>_" + CompuMethod.CompuScale.CompuConst 　　CompuRationalCoeffs = CompuMethod.CompuScale.CompuRationalCoeffs
Signal	SystemSignal 　ShortName = Signal.ShortName 　Length　　= Signal.Coding.BitLength ConstantSpecification 　ShortName = "C_" + Signal.ShortName 　BooleanLiteral|OpaqueLiteral|IntegerLiteral|RealLiteral 　　　　　　　　　　　　|CharLiteral|StringLiteral 　　ShortName = "C_" + Signal.ShortName 　　Value　　= Signal.DefaultValue
Frame	Frame 　ShortName　= Frame.ShortName 　FrameLength = Frame.ByteLength
SignalGroup	SignalIPdu|DcmIPdu|NmPdu|NPdu 　ShortName = SignalGroup.ShortName 　Length　　= SignalGroup.BitLength Frame.PduToFrameMapping 　ShortName　　　= SignalGroup.ShortName 　PackingByteOrder = Intel 　StartPosition　= Frame.SignalInstance.BitPosition 　　　　　　　　　- SignalGroup.OrderedSignal.BitPosition
Frame 　.SignalInstance 　.Signal	ISignal 　ShortName = Signal.ShortName + "__" + Frame.ShortName SignalIPdu.ISignalToIPduMapping

based on template version 4.11.1

| | ```
ShortName        = Signal.ShortName
PackingByteOrder = SignalInstance.Intel|Motorola
StartPosition    = SignalInstance.BitPosition
                      - Frame.PduToFrameMapping.StartPosition
``` |
| Frame<br>.Multiplexer<br>.Switch | ```
MultiplexedIPdu
  ShortName = Frame.ShortName
  Length    = 8*Frame.ByteLength
Frame.PduToFrameMapping
  ShortName        = Frame.ShortName
  PackingByteOrder = Intel
  StartPosition    = 0
MultiplexedIPdu.SelectorField
  ByteOrder     = Switch.Intel|Motorola
  Length        = Switch.BitLength
  StartPosition = Switch.BitPosition
``` |
| Frame<br>.Multiplexer<br>.Data<br>.SubFrame | ```
MultiplexedIPdu.DynamicPart.DynamicPartAlternative
  SelectorFieldCode = SubFrame.SwitchCode
  SignalIPdu
    ShortName = SubFrame.ShortName + "_Mx" + SubFrame.SwitchCode
    Length    = 8*Frame.ByteLength
``` |
| Cluster(FlexRay) | ```
FlexrayCluster
  ShortName                = Cluster.ShortName
  MaxFrameLength           = Cluster.MaxFrameLength
  ProtocolName             = Cluster.Protocol
  ProtocolVersion          = Cluster.ProtocolVersion
  Speed                    = Cluster.Speed
  ActionPointOffset        = Cluster.ActionPointOffset
  Bit                      = Cluster.Bit/1000000
  CasRxLowMin              = Cluster.CasRxLowMin
  CasRxLowMax              = Cluster.CasRxLowMax
  ColdStartAttempts        = Cluster.ColdStartAttempts
  Cycle                    = Cluster.Cycle/1000000
  DynamicSlotIdlePhase     = Cluster.DynamicSlotIdlePhase
  ListenNoise              = Cluster.ListenNoise
  MacroPerCycle            = Cluster.MacroPerCycle
  MacrotickDuration        = Cluster.Macrotick/1000000
  MaxInitialisationError   = Cluster.MaxInitializationError/100000
  MaxWithoutClockCorrectionFatal
                           = Cluster.MaxWithoutClockCorrectionFatal
  MaxWithoutClockCorrectionPassive
                           = Cluster.MaxWithoutClockCorrectionPassive
  MinislotActionPointOffset = Cluster.MinislotActionPointOffset
  MinislotDuration         = Cluster.Minislot
  NetworkIdleTime          = Cluster.NIT
  NetworkManagementVectorLength
                           = Cluster.NetworkManagementVectorLength
  NumberOfCycles           = Cluster.NumberOfCycles
  NumberOfMinislots        = Cluster.NumberOfMinislots
  NumberOfStaticSlots      = Cluster.NumberOfStaticSlots
  OffsetCorrectionMax      = Cluster.OffsetCorrectionMax/1000000
  OffsetCorrectionStart    = Cluster.OffsetCorrectionStart
  PayloadLengthStatic      = Cluster.PayloadLengthStatic
  SampleClockPeriod        = Cluster.SampleClockPeriod/1000000
  StaticSlotDuration       = Cluster.StaticSlot
  SymbolWindow             = Cluster.SymbolWindow
  SyncFrameIdCountMax      = Cluster.SyncNodeMax
  TransmissionStartSequenceDuration = Cluster.TSSTransmitter
  WakeupRxIdle             = Cluster.WakeUpSymbolRxIdle
  WakeupRxLow              = Cluster.WakeUpSymbolRxLow
  WakeupRxWindow           = Cluster.WakeUpSymbolRxWindow
  WakeupTxActive           = Cluster.WakeUpSymbolTxLow
  WakeupTxIdle             = Cluster.WakeUpSymbolTxIdle
``` |
| Channel | ```
FlexrayCluster.FlexrayPhysicalChannel
  ShortName   = Channel.ShortName
  ChannelName = Channel.FlexrayChannelName
``` |
| Channel<br>.FrameTriggering | ```
FlexrayPhysicalChannel.FlexrayFrameTriggering
  ShortName = "FT_" + FrameTriggering.AbsolutelyScheduledTiming
``` |
| Channel<br>.FrameTriggering<br>.AbsolutelyScheduledTiming | ```
FlexrayFrameTriggering.AbsolutelyScheduledTiming
  SlotID        = AbsolutelyScheduledTiming.SlotID
  BaseCycle     = AbsolutelyScheduledTiming.BaseCycle
``` |

based on template version 4.11.1

	`CycleRepetition = AbsolutelyScheduledTiming.CycleRepetition`

`Channel`
` .FrameTriggering`
` .Frame`
` .SignalGroup`
	`FlexrayPhysicalChannel.IPduTriggering`
	`    ShortName = "PT_" + SignalGroup.ShortName`

`Channel`
` .FrameTriggering`
` .Frame`
` .SignalGroup`
` .Signal`
	`FlexrayPhysicalChannel.SignalTriggering`
	`    ShortName = "ST_" + Signal.ShortName + "__"+ SignalGroup.ShortName`

`Channel`
` .FrameTriggering`
` .CyclicTiming`
	`SignalIPdu.IPduTimingSpecification.CyclicTiming`
	`    RepeatingTime = CyclicTiming.RepeatingTimeRange`

`Channel`
` .FrameTriggering`
` .EventControlledTiming`
	`SignalIPdu.IPduTimingSpecification.EventControlledTiming`
	`    RepetitionPeriod = EventControlledTiming.DebounceTimeRange`

`Channel`
` .FrameTriggering`
` .RequestControlledTiming`
	`SignalIPdu.IPduTimingSpecification.RequestControlledTiming`
	`    ResponseTime = RequestControlledTiming.ResponseTimeRange`

`Ecu`	`EcuInstance` `    ShortName = Ecu.ShortName`
`Ecu` ` .Controller`	`EcuInstance.FlexrayCommunicationController` `    ShortName                  = Controller.ShortName` `    AcceptedStartupRange       = Controller.AcceptedStartupRange` `    AllowHaltDueToClock        = Controller.AllowHaltDueToClock` `    AllowPassiveToActive       = Controller.AllowPassiveToActive` `    ClusterDriftDamping        = Controller.ClusterDriftDamping` `    DecodingCorrection         = Controller.DecodingCorrection` `    DelayCompensationA         = Controller.DelayCompensationA` `    DelayCompensationB         = Controller.DelayCompensationB` `    ExternOffsetCorrection     = Controller.ExternOffsetCorrection` `    ExternRateCorrection       = Controller.ExternRateCorrection` `    KeySlotId                  = Controller.KeySlotUsage.StartupSync` `	Controller.KeySlotUsage.Sync` ` KeySlotUsedForStartUp = Controller.KeySlotUsage.StartupSync!=null` ` KeySlotUsedForSync = Controller.KeySlotUsage.StartupSync!=null` `	Controller.KeySlotUsage.Sync !=null` ` LatestTx = Controller.LatestTx` ` ListenTimeout = Controller.ListenTimeout` ` MacroInitialOffsetA = Controller.MacroInitialOffsetA` ` MacroInitialOffsetB = Controller.MacroInitialOffsetB` ` MaximumDynamicPayloadLength = Controller.MaxDynamicPayloadLength` ` MicroInitialOffsetA = Controller.MicroInitialOffsetA` ` MicroInitialOffsetB = Controller.MicroInitialOffsetB` ` MicroPerCycle = Controller.MicroPerCycle` ` MicrotickDuration = Controller.Microtick/1000000` ` OffsetCorrectionOut = Controller.OffsetCorrectionOut` ` RateCorrectionOut = Controller.RateCorrectionOut` ` SamplesPerMicrotick = Controller.SamplesPerMicrotick` ` WakeUpPattern = Controller.WakeUpPattern`

`Ecu`
` .Connector`
	`EcuInstance.FlexRayCommunicationConnector`
	`    ShortName     = "CN_" + Cluster.ShortName`
	`                          + "__" + Connector.Channel.ShortName`
	`    TpAddress     = Ecu.DiagnosticAddress[Physical]`
	`    WakeUpChannel = Connector.WakeUpChannel`
	`EcuInstance.AssociatedIPduGroup(Rx)`
	`    ShortName = Ecu.ShortName`
	`                + "__" + Connector.Channel.ShortName + "_Rx"`
	`EcuInstance.AssociatedIPduGroup(Tx)`
	`    ShortName = Ecu.ShortName`
	`                + "__" + Connector.Channel.ShortName + "_Tx"`

`Ecu`
` .Connector`
` .InputPort`
` .FrameTriggering`
	`FlexRayCommunicationConnector.FramePort`
	`    ShortName = "FP_"`
	`                + FrameTriggering.AbsolutelyScheduledTiming + "_Rx"`
	`    Direction = In`

`Ecu`
` .Connector`
` .InputPort`
` .SignalInstance`
` .Signal`
` .SignalGroup`
	`FlexRayCommunicationConnector.IPduPort`
	`    ShortName = "PP_" + SignalGroup.ShortName + "_Rx"`
	`    Direction = In`
	`FlexRayCommunicationConnector.SignalPort`
	`    ShortName = "SP_" + Signal.ShortName`
	`                + "__" + SignalGroup.ShortName + "_Rx"`

	`Direction = In`

Ecu
 `.Connector`
 `.OutputPort`
 `.FrameTriggering`
	`FlexRayCommunicationConnector.FramePort`
	 `ShortName = "FP_"`
	`                + FrameTriggering.AbsolutelyScheduledTiming + "_Tx"`
	 `Direction = Out`

Ecu
 `.Connector`
 `.OutputPort`
 `.SignalInstance`
 `.Signal`
 `.SignalGroup`
	`FlexRayCommunicationConnector.IPduPort`
	 `ShortName = "PP_" + SignalGroup.ShortName + "_Tx"`
	 `Direction = Out`
	`FlexRayCommunicationConnector.SignalPort`
	 `ShortName = "SP_" + Signal.ShortName`
	`                + "__" + SignalGroup.ShortName + "_Tx"`
	 `Direction = Out`

Table 2-5   Transformation of Fibex 2.0.1 elements.

The subsequent table shows how Fibex 3.0.0 or 3.1.0 elements are mapped to AUTOSAR 3.1.4 or AUTOSAR 3.2.1 elements.

Fibex element	AUTOSAR Element
`PhysicalDimension`	see Table 2-5
`Unit`	see Table 2-5
`Coding`	see Table 2-5
`Signal`	see Table 2-5
`Frame`	see Table 2-5

`Frame`
`  .PduInstance`
	`Frame.PduToFrameMapping`
	`    ShortName               = PduInstance.Pdu.ShortName`
	`    PackingByteOrder        = PduInstance.Intel|Motorola`
	`    StartPosition           = PduInstance.BitPosition`
	`    UpdateIndicationBitPosition = PduInstance.PduUpdateBitPosition`

`Pdu`	`SignalIPdu|MultiplexedIPdu|DcmIPdu|NmPdu|NPdu`
	`    ShortName = Pdu.ShortName`
	`    Length    = 8*Pdu.ByteLength`

`Pdu`
`  .SignalInstance`
`  .Signal`
	`ISignal`
	`    ShortName = Signal.ShortName + "__" + Pdu.ShortName`
	`SignalIPdu.ISignalToIPduMapping`
	`    ShortName       = Signal.ShortName`
	`    PackingByteOrder = SignalInstance.Intel|Motorola`
	`    StartPosition    = SignalInstance.BitPosition`

`Pdu`
`  .Multiplexer`
`  .Switch`
	`MultiplexedIPdu.SelectorField`
	`    ByteOrder     = Switch.Intel|Motorola`
	`    Length        = Switch.BitLength`
	`    StartPosition = Switch.BitPosition`

`Pdu`
`  .Multiplexer`
`  .DynamicPart`
`  .SwitchedPduInstance`
	`MultiplexedIPdu.DynamicPart.DynamicPartAlternative`
	`    SelectorFieldCode = SwitchedPduInstance.SwitchCode`
	`    SignalIPdu`
	`      ShortName = SwitchedPduInstance.Pdu.ShortName`
	`      Length    = 8*SwitchedPduInstance.Pdu.ByteLength`

`Cluster(FlexRay)`	see Table 2-5
`Channel`	see Table 2-5
`Channel` `  .FrameTriggering`	see Table 2-5

`Channel`
`  .FrameTriggering`
`  .AbsolutelyScheduledTiming`
	see Table 2-5

`Channel` `  .PduTriggering`	`FlexrayPhysicalChannel.IPduTriggering` `    ShortName = "PT_" + PduTriggering.Pdu.ShortName`

`Channel`
`  .PduTriggering`
`  .Pdu`
`  .SignalInstance`
`  .Signal`
	`FlexrayPhysicalChannel.SignalTriggering`
	`    ShortName = "ST_" + Signal.ShortName`
	`                    + "__" + PduTriggering.Pdu.ShortName`

`Channel`
`  .PduTriggering`
`  .CyclicTiming`
	`SignalIPdu.IPduTimingSpecification.CyclicTiming`
	`    FinalRepetitions = CyclicTiming.FinalRepetitions`
	`    RepeatingTime    = CyclicTiming.RepeatingTimeRange`
	`    StartingTime     = CyclicTiming.StartingTimeRange`

`Channel`
`  .PduTriggering`
`  .EventControlledTiming`
	`SignalIPdu.IPduTimingSpecification.EventControlledTiming`
	`    NumberOfRepeats  = EventControlledTiming.FinalRepetitions`
	`    RepetitionPeriod = EventControlledTiming.DebounceTimeRange`

`Channel`
`  .PduTriggering`
`  .RequestControlledTiming`
	`SignalIPdu.IPduTimingSpecification.RequestControlledTiming`
	`    ResponseTime = RequestControlledTiming.ResponseTimeRange`

`Ecu`	see Table 2-5
`Ecu` `  .Controller`	see Table 2-5
`Ecu` `  .Connector`	see Table 2-5

Ecu .Connector .InputPort .FrameTriggering	see Table 2-5

Ecu
 .Connector
 .InputPort
 .IncludedPdu
 .PduTriggering
	FlexRayCommunicationConnector.IPduPort
	   ShortName = "PP_" + PduTriggering.Pdu.ShortName + "_Rx"
	   Direction = In

Ecu
 .Connector
 .InputPort
 .IncludedPdu
 .IncludedSignal
 .SignalInstance
	FlexRayCommunicationConnector.SignalPort
	   ShortName = "SP_" + SignalInstance.Signal.ShortName + "__"
	                   + IncludedPdu.PduTriggering.Pdu.ShortName + "_Rx"
	   Direction = In

Ecu .Connector .OutputPort .FrameTriggering	see Table 2-5

Ecu
 .Connector
 .OutputPort
 .IncludedPdu
 .PduTriggering
	FlexRayCommunicationConnector.IPduPort
	   ShortName = "PP_" + PduTriggering.Pdu.ShortName + "_Tx"
	   Direction = Out

Ecu
 .Connector
 .OutputPort
 .IncludedPdu
 .IncludedSignal
 .SignalInstance
	FlexRayCommunicationConnector.SignalPort
	   ShortName = "SP_" + SignalInstance.Signal.ShortName + "__"
	                   + IncludedPdu.PduTriggering.Pdu.ShortName + "_Tx"
	   Direction = Out

TpConfig
 .TpAddress
	FlexrayPhysicalChannel.TpAddress
	   ShortName = "TA_" + TpAddress
	   TpAddress = TpAddress

TpConfig
 .TpChannel
	FlexrayPhysicalChannel.FlexrayTpChannel
	   AckType = TpChannel.AckType
	   ExtendedAddressing = TpChannel.AddressingType == FrtpTb
	   MaxBs = TpChannel.MaxBlockSize
	   MaxRetries = TpChannel.MaxRetries
	   MaximumMessageLength = TpChannel.MaximumMessageLength
	   MulticastSegmentation = TpChannel.GroupSegmentation
	   TimeoutBs = TpChannel.TimeoutBs
	   TimeoutCr = TpChannel.TimeoutCr
	   TransmitCancellation = TpChannel.TransmitCancellation

TpConfig
 .TpChannel
 .TpConnection
	FlexrayTpChannel.FlexRayTpConnection
	FlexrayTpChannel.FlexRayTpConnection.DirectTpSdu
	   ShortName = TpConnection.ShortName + "_Rq"
	FlexrayTpChannel.FlexRayTpConnection.ReversedTpSdu
	   ShortName = TpConnection.ShortName + "_Rs"

TpConfig
 .TpNode
	FlexrayPhysicalChannel.FlexrayTpNode
	   ShortName = TpNode.ShortName
	FlexrayPhysicalChannel.FlexrayTpChannel
	   MaxAr = TpNode.MaxAr
	   MaxAs = TpNode.MaxAs
	   MaxBufferRequest = TpNode.BufferRequest
	   MaxFrIf = TpNode.MaxFrif
	   MinimumSeparationTime = TpNode.Stmin
	   TimeBuffer = TpNode.TimeBuffer
	   TimeFrIf = TpNode.TimeFrif
	   TimeoutAr = TpNode.TimeoutAr
	   TimeoutAs = TpNode.TimeoutAs

Table 2-6   Transformation of Fibex 3.0.0/3.1.0 elements.

2.4 Extension File

The Vector System Description Extension (VSDE) file is used to supplement the content of DBC-, LDF- or Fibex files. An extension file defines certain communication elements which might be missing in the original legacy communication databases, or which cannot be specified with these communication databases. The table below explains the extension elements which are supported so far.

VSDE element	Description

`<FILTERING>`
`  <FLEXRAY-CLUSTER-REF CHANNEL="A">FlexRay01`
`    </FLEXRAY-CLUSTER-REF>`
`</FILTERING>`
	The dual channel FlexRayCluster FlexRay01 is filtered for its A channel.
	This feature is supported for Fibex databases.

`<CAN-CLUSTER-NAME>`
`  <CAN-CLUSTER-REF>Can01</CAN-CLUSTER-REF>`
`  <SHORT-NAME>Can01NewName</SHORT-NAME>`
`</CAN-CLUSTER-NAME>`
	The CanCluster Can01 obtains a new name Can01NewName. Similarly, LinClusters and FlexrayClusters can be renamed.
	This feature is supported for DBC, LDF and Fibex databases.

`<ECU-INSTANCE-NAME>`
`  <ECU-INSTANCE-REF>Ecu01</ECU-INSTANCE-REF>`
`  <SHORT-NAME>Ecu01NewName</SHORT-NAME>`
`</ECU-INSTANCE-NAME>`
	The ECU Ecu01 obtains a new name Ecu01NewName.
	This feature is supported for DBC, LDF and Fibex databases.

`<SYSTEM-SIGNAL-NAME>`
`  <SIGNAL-I-PDU-REF>Pdu01</SIGNAL-I-PDU-REF>`
`  <SYSTEM-SIGNAL-REF>Sig01</SYSTEM-SIGNAL-REF>`
`  <SHORT-NAME>Sig01NewName</SHORT-NAME>`
`</SYSTEM-SIGNAL-NAME>`
	The signal Sig01 within pdu Pdu01 obtains a new name Sig01NewName. Signal renaming is used, e.g., to distinguish signals of the same name in different pdus.
	This feature is supported for DBC, LDF and Fibex databases.

`<SYSTEM-SIGNAL-GROUP>`
`  <SHORT-NAME>SG_SigGrp01</SHORT-NAME>`
`  <SYSTEM-SIGNAL-REFS>`
`    <SYSTEM-SIGNAL-REF>Sig01</SYSTEM-SIGNAL-REF>`
`    <SYSTEM-SIGNAL-REF>Sig02</SYSTEM-SIGNAL-REF>`
`  </SYSTEM-SIGNAL-REFS>`
`</SYSTEM-SIGNAL-GROUP>`
	The signals Sig01 and Sig02 are aggregated to a new signal group SG_SigGrp01. The signals must be defined in the same database. Each pdu must contain all or none of these signals.
	This feature is supported for DBC, LDF and Fibex databases.

`<SAFETY-PDU>`
`  <SIGNAL-I-PDU-REF>Pdu01</SIGNAL-I-PDU-REF>`
`  <CREATE-PDU-GAP-SIGNALS>true</CREATE-PDU-GAP-SIGNALS>`
`</SAFETY-PDU>`
	All signals of pdu Pdu01 are aggregated to a new signal group SG_Pdu01. Optionally, the pdu gaps are filled with artificial gap signals, when then also become part of the new signal group.
	This feature is supported for DBC, LDF and Fibex databases.

`<BIDIRECTIONAL-PDU>`
`  <SIGNAL-I-PDU-REF>Pdu01</SIGNAL-I-PDU-REF>`
`</BIDIRECTIONAL-PDU>`
	The pdu Pdu01 can be send and received by the same ECU.
	This feature is supported for DBC and Fibex databases.

`<PDU-GROUP>`
`  <SHORT-NAME>PduGrp01</SHORT-NAME>`
`  <SIGNAL-I-PDU-REFS>`
`    <SIGNAL-I-PDU-REF>Pdu01</SIGNAL-I-PDU-REF>`
`    <SIGNAL-I-PDU-REF>Pdu02</SIGNAL-I-PDU-REF>`
`  </SIGNAL-I-PDU-REFS>`
`</PDU-GROUP>`
	For each cluster and ECU which sends or receives the pdus Pdu01 or Pdu02 a new Tx or Rx pdu group is created. The pdus are not assigned to the standard pdu groups created by the VLC.
	This feature is supported for DBC, LDF and Fibex databases.

`<PDU-GROUP-DEFINITION>`
`  <FLEXRAY-CLUSTER-REF>FlexRay01</FLEXRAY-CLUSTER-REF>`
`  <ECU-INSTANCE-REF>Ecu01</ECU-INSTANCE-REF>`
`  <CHANNEL-SPECIFIC>true</CHANNEL-SPECIFIC>`
`</PDU-GROUP-DEFINITION>`
	The dual channel FlexRayCluster FlexRay01 obtains for its ECU Ecu01 two standard Tx pdu groups and two standard Rx pdu groups – one for each channel and direction, respectively.
	This feature is supported for Fibex databases.

| ```xml
<SIGNAL-UPDATE-DEFINITION>
  <SIGNAL-I-PDU-REF>Pdu01</SIGNAL-I-PDU-REF>
  <UPDATE-INDICATION-SIGNAL-REF>
    SigUpd01_UB</UPDATE-INDICATION-SIGNAL-REF>
  <UPDATED-SIGNALS>
    <SYSTEM-SIGNAL-REF>Sig01</SYSTEM-SIGNAL-REF>
  </UPDATED-SIGNALS>
  <UPDATED-SIGNAL-GROUPS>
    <SYSTEM-SIGNAL-GROUP-REF>
      SG_SigGrp01</SYSTEM-SIGNAL-GROUP-REF>
  </UPDATED-SIGNAL-GROUPS>
</SIGNAL-UPDATE-DEFINITION>
``` | The signal SigUpd01_UB within pdu Pdu01 serves as update signal for the signal Sig01 and the signal group SG_SigGrp01. The update signal can be used for one or more signals and signal groups within a pdu at the same time. <br><br> This feature is supported for DBC, LDF and Fibex databases. |
| ```xml
<I-PDU-TIMING>
  <SIGNAL-I-PDU-REF>Pdu01</SIGNAL-I-PDU-REF>
  <NUMBER-OF-REPETITIONS>10</NUMBER-OF-REPETITIONS>
  <REPETITION-PERIOD>0.001</REPETITION-PERIOD>
  <MINIMUM-DELAY>0.01</MINIMUM-DELAY>
  <SIGNAL-TIMINGS>
    <SIGNAL-TIMING>
      <SYSTEM-SIGNAL-REF>Sig01</SYSTEM-SIGNAL-REF>
      <SIGNAL-SEND-TYPE>ON-CHANGE</SIGNAL-SEND-TYPE>
    </SIGNAL-TIMING>
  </SIGNAL-TIMINGS>
  <ACCESS-RIGHTS>READ-ONLY</ACCESS-RIGHTS>
</I-PDU-TIMING>
``` | The timing elements NumberOfRepetitions, RepetitionPeriod and MinimumDelay of pdu Pdu01 override the corresponding database settings. Further, the SignalSendType timing element of signal Sig01 overrides the signal specific timing settings. Finally, the element AccessRights defines whether the timing data later can be changed in the Vector tool chain. <br><br> This feature is supported for DBC databases. The AccessRights element is also supported for LDF and Fibex databases. |
| ```xml
<CAN-TP-CONNECTION>
  <SHORT-NAME>Can01_Pdu01_Pdu02</SHORT-NAME>
  <CAN-CLUSTER-REF>Can01</CAN-CLUSTER-REF>
  <DATA-PDU-REF>Pdu01</DATA-PDU-REF>
  <FLOW-CONTROL-PDU-REF>Pdu02</FLOW-CONTROL-PDU-REF>
</CAN-TP-CONNECTION>
``` | The directly opposed pdus Pdu01 and Pdu02 of CanCluster Can01 are combined to a new CanTpConnection Can01_Pdu01_Pdu02. The VSDE internal CanTpConnection name can be referred by TpHighLevelRoutings. Similarly, pdus can be combined to a LinTpConnection. <br><br> This feature is supported for DBC and LDF databases. |
| ```xml
<PDUR-MESSAGE-ROUTING>
  <ECU-INSTANCE-REF>Ecu01</ECU-INSTANCE-REF>
  <SOURCE-CAN-CLUSTER-REF>Can01</SOURCE-CAN-CLUSTER-REF>
  <TARGET-CAN-CLUSTER-REF>Can02</TARGET-CAN-CLUSTER-REF>
  <I-PDU-MAPPINGS>
    <I-PDU-MAPPING>
      <ROUTE-DLC>true</ROUTE-DLC>
      <SOURCE-I-PDU-REF>Pdu01</SOURCE-I-PDU-REF>
      <SOURCE-SIGNALS>
        <SYSTEM-SIGNAL-REF>Sig01</SYSTEM-SIGNAL-REF>
      </SOURCE-SIGNALS>
      <TARGET-I-PDU-REF>Pdu02</TARGET-I-PDU-REF>
    </I-PDU-MAPPING>
  </I-PDU-MAPPINGS>
</PDUR-MESSAGE-ROUTING>
``` | The pdu Pdu01 of CanCluster Can01 is routed via the gateway ECU Ecu01 to the pdu Pdu02 of CanCluster Can02. The pdu Pdu01 will be routed by the PDUR module, and also its DLC value will be routed. The signal Sig01 of pdu Pdu01 is received by the gateway ECU Ecu01, all other signals of pdu Pdu01 are not received by Ecu01. <br><br> This feature is supported for DBC, LDF and Fibex databases. |

```xml
<COM-MESSAGE-ROUTING>
  <ECU-INSTANCE-REF>Ecu01</ECU-INSTANCE-REF>
  <SOURCE-CAN-CLUSTER-REF>Can01</SOURCE-CAN-CLUSTER-REF>
  <TARGET-CAN-CLUSTER-REF>Can02</TARGET-CAN-CLUSTER-REF>
  <I-PDU-MAPPINGS>
    <I-PDU-MAPPING>
      <PROCESSING>
```

based on template version 4.11.1

Code	Description
`</COM-MESSAGE-ROUTING>`	

`<COM-SIGNAL-ROUTING>`
`  <ECU-INSTANCE-REF>Ecu01</ECU-INSTANCE-REF>`
`  <SOURCE-CAN-CLUSTER-REF>Can01</SOURCE-CAN-CLUSTER-REF>`
`  <TARGET-CAN-CLUSTER-REF>Can02</TARGET-CAN-CLUSTER-REF>`
`  <SIGNAL-MAPPINGS>`
`    <SIGNAL-MAPPING>`
`      <PROCESSING>DEFERED</PROCESSING>`
`      <SOURCE-I-PDU-REF>Pdu01</SOURCE-I-PDU-REF>`
`      <SOURCE-SIGNAL-REF>Sig01</SOURCE-SIGNAL-REF>`
`      <TARGET-I-PDU-REF>Pdu02</TARGET-I-PDU-REF>`
`      <TARGET-SIGNAL-REF>Sig02</TARGET-SIGNAL-REF>`
`    </SIGNAL-MAPPING>`
`  </SIGNAL-MAPPINGS>`
`</COM-SIGNAL-ROUTING>`
	The signal Sig01 within pdu Pdu01 of CanCluster Can01 is routed via the gateway ECU Ecu01 to the signal Sig02 within pdu Pdu02 of CanCluster Can02. The signal Sig01 will be routed deferred by the COM module.
	
	This feature is supported for DBC, LDF and Fibex databases.

`<TP-HIGH-LEVEL-ROUTING>`
`  <ECU-INSTANCE-REF>Ecu01</ECU-INSTANCE-REF>`
`  <SOURCE-CAN-TP-CONNECTION-REF>Can01_Pdu01_Pdu02`
`    </SOURCE-CAN-TP-CONNECTION-REF>`
`  <TARGET-CAN-TP-CONNECTION-REF>Can02_Pdu03_Pdu04`
`    </TARGET-CAN-TP-CONNECTION-REF>`
`</TP-HIGH-LEVEL-ROUTING>`
	The CanTpConnection Can01_Pdu01_Pdu02 is routed via the gateway ECU Ecu01 to the CanTpConnection Can02_Pdu03_Pdu04. CanTpConnections and LinTpConnections are defined by the VSDE file, while FlexrayTpConnections are provided by Fibex databases.
	
	This feature is supported for DBC, LDF and Fibex databases.

`<TP-LOW-LEVEL-ROUTING>`
`  <ECU-INSTANCE-REF>Ecu01</ECU-INSTANCE-REF>`
`  <SOURCE-CAN-CLUSTER-REF>Can01</SOURCE-CAN-CLUSTER-REF>`
`  <TARGET-CAN-CLUSTER-REF>Can02</TARGET-CAN-CLUSTER-REF>`
`  <N-PDU-MAPPINGS>`
`    <N-PDU-MAPPING>`
`      <SOURCE-N-PDU-REF>Pdu01</SOURCE-N-PDU-REF>`
`      <TARGET-N-PDU-REF>Pdu02</TARGET-N-PDU-REF>`
`    </N-PDU-MAPPING>`
`  </N-PDU-MAPPINGS>`
`</TP-LOW-LEVEL-ROUTING>`
	The n-pdu Pdu01 of CanCluster Can01 is routed via the gateway ECU Ecu01 to the n-pdu Pdu02 of CanCluster Can02.
	
	This feature is supported for DBC databases.

`<PNC-CONFIGURATION>`
`  <PNC-VECTOR-LENGTH>3</PNC-VECTOR-LENGTH>`
`  <PNC-VECTOR-OFFSET>5</PNC-VECTOR-OFFSET>`
`  <PNC-CLUSTERS>`
`    <PNC-CLUSTER>`
`      <CAN-CLUSTER-REF>Can01</CAN-CLUSTER-REF>`
`      <PNC-ECUS>`
`        <PNC-ECU>`
`          <ECU-INSTANCE-REF>Ecu01</ECU-INSTANCE-REF>`
`          <PNC-GATEWAY-TYPE>ACTIVE</PNC-GATEWAY-TYPE>`
`          <PNC-WAKEUP-CAN-ID>452984832`
`            </PNC-WAKEUP-CAN-ID>`
`          <PNC-WAKEUP-CAN-ID-EXTENDED>true`
`            </PNC-WAKEUP-CAN-ID-EXTENDED>`
`          <PNC-WAKEUP-CAN-ID-MASK>127`
`            </PNC-WAKEUP-CAN-ID-MASK>`
`          <PNC-WAKEUP-DATA-MASK>4611686018427387904`
`            </PNC-WAKEUP-DATA-MASK>`
`          <PNC-WAKEUP-DLC>8</PNC-WAKEUP-DLC>`
`          <PNC-GROUPS>`
`            <PNC-GROUP>`
`              <PNC-IDENTIFIER>1</PNC-IDENTIFIER>`
`              <COMMUNICATION-DIRECTION>IN`
`                <COMMUNICATION-DIRECTION>`
`              <SIGNAL-I-PDU-REFS>`
`                <SIGNAL-I-PDU-REF>Pdu01`
`                  </SIGNAL-I-PDU-REF>`
`              </SIGNAL-I-PDU-REFS>`
`              <MULTIPLEXED-I-PDU-REFS>`
`                <MULTIPLEXED-I-PDU-REF>Pdu02`
`                  </MULTIPLEXED-I-PDU-REF>`
`              </MULTIPLEXED-I-PDU-REFS>`
`            </PNC-GROUP>`
`          </PNC-GROUPS>`
`        </PNC-ECU>`
`      </PNC-ECUS>`
	The pdus Pdu01 and Pdu02 of CanCluster Can01 are combined to a PNC group for ECU Ecu01 and the partial network with the ID 1. A partial network is defined by all PNC groups which refer the same partial network ID.
	
	This feature is supported for DBC and Fibex databases.

```
      </PNC-CLUSTER>
    </PNC-CLUSTERS>
</PNC-CONFIGURATION>
```

Table 2-7   Vector System Description Extension file elements.

The extension file is provided as a file parameter to the VLC. Its XML schema is described with the ExtractExtension.xsd file.

3 Glossary and Abbreviations

3.1 Glossary

Term	Description

3.2 Abbreviations

Abbreviation	Description
DPA	DaVinci Project Assistant
VLC	Vector Legacy Converter
VSDE	Vector System Description Extension

4   Contact

Visit our website for more information on

> News

> Products

> Demo software

> Support

> Training data

> Addresses

www.vector.com

based on template version 4.11.1