

# Project Scheme Documentation

## Document information

Association Name, WG	KNX ASSOCIATION
Author(s):	KNX & DEV
Status:	Valid
Version:	01.00.01
Date:	03.11.2021
Document file name:	Project Scheme21 v01.00.01.docx
Number of pages:	64

## Acronyms

DEV	KNX Development subcontractors
KNX	KNX Association
MT6	KNX Manufacturer Tool 6

## Referenced documents

[XSD]	XML scheme (KNX-Project-Scheme-v21.xsd. part of KNX MT6 → Version 6.0)
[DS]	XML DSIG documentation (xmldsig-core-schemescheme.xsd)

## List of Changes

Version	Date	Maturity	Author	Description
01.00.00	15.10.2021	Valid	KNX Association	- Initial public version, derived from KNX internal version 2.1.14, for XML scheme 2.0 (ETS6) → Version 6.0
01.00.01	03.11.2021	Valid	KNX Association	- Added note about UTF-16 encoding of the project password for the calculation of the project file encryption derived key

## **Disclaimer**

The document is subject to change without prior notice. KNX Association SHALL IN ANY CASE NOT BE LIABLE FOR DIRECT AND INDIRECT DAMAGES ARISING FROM incorrect or missing descriptions in this document, especially when basing software and or hardware developments on the content of this document.

# Contents

<b>1</b>	<b>Overview</b>	<b>4</b>
1.1	Document Purpose	4
1.2	Extended Import Restrictions	4
1.3	Extended Import Checks	4
1.4	Validity	4
1.5	Namespaces	5
<b>2</b>	<b>XSD Scheme File &amp; KNX Master Data File</b>	<b>5</b>
<b>3</b>	<b>Elements, Types and Attributes</b>	<b>6</b>
1.1	General	6
1.1.1	element KNX	6
1.1.2	Enumerations	6
1.1.3	Other simpleTypes	26
1.2	Project Data	32
1.2.1	element KNX/Project	32
1.2.2	complexType Project_t	33
1.2.3	General	34
1.2.4	Topology	39
1.2.5	Device Data	42
1.2.6	Building Structure	54
1.2.7	Group Addresses	58
1.2.8	SplitInfos	61
<b>4</b>	<b>Transfer files</b>	<b>62</b>
4.1	File extensions	62
4.2	Content	62
4.2.1	Non-XML files	62
4.2.2	Distribution to partial XML files	62
4.2.3	Naming convention	63
4.2.4	Password protection	64

# 1 Overview

With introduction of ETS4, the ex/- import format for KNX projects and products changed to a standard XML based format. The projects exported with ETS4, ETS5 and ETS6 have the file extension \*.knxproj, which has been extended since ETS4.

## 1.1 Document Purpose

This document describes all necessary elements, types and attributes of the KNX XML Scheme [XSD] for an ETS6 created project. All other –for the project scope not relevant - elements/ attributes might be missing or simply only listed (but not described).

The main use case is to read in (import) ETS6 projects into external tools (e.g. visualizations), but another use case might be to create an ETS6 project from scratch and later import into ETS6 (import is however restricted).

The document does not describe how manufacturers create and define products (parameter and/or Group Object dependencies and their visibility in correlation with download image creation) to compile valid device configurations outside ETS6, as this is an exclusive task of the Manufacturer Tool.

## 1.2 Extended Import Restrictions

ETS6 imports projects only from a trusted source, which means:

1. The project originates (exported) from an ETS
2. The project originates from a KNX member (and only products of this member are contained in the project)

This is done via a dedicated project signature, in case of 2 the KNX manufacturer shall obtain a unique signature. This implies that an ‘unreliable’ project import - from a source not trusted by ETS - is not possible!

Extended import restrictions implemented in the ETS4, ETS5 and ETS6.

## 1.3 Extended Import Checks

ETS6 checks on import if a project is valid as regards conformance to the XML conformity (syntax check), i.e. the ETS6 checks if the project format is correct. ETS6 does not check if the saved data inside the file (normally a project/ installation) is a valid project/ installation configuration (semantic check), e.g. if such a project is semantically valid<sup>1</sup>.

Hence, it is expected that saved projects & configurations are valid as regards ETS project and installation data integrity.

## 1.4 Validity

This XML documentation refers to XML scheme version 2.1 (as currently implemented in ETS 6.0).

---

<sup>1</sup> This validity covers things such as *KNX project settings used and processed by ETS* up to any *manufacturer device configuration* (with its communication object/ parameter dependencies and visibilities).

## 1.5 Namespaces

The “targetNamespace” is defined as “<http://knx.org/xml/project/21>”; the prefix knx is used here. The scheme references the name spaces <http://www.w3.org/2001/XMLSchema> (prefix xs).

## 2 XSD Scheme File & KNX Master Data File

The KNX XML scheme is normally defined and described in a file with file extension \*.xsd. This file is not part of an ETS6 installation, but of MT6 (the MT6 purpose is to build/ compile valid KNX products and therefore it uses the XML scheme as a basis).

The KNX master data contains data definitions, which describe basic KNX system properties as data point types, manufacturer IDs and other things. This data is mandatory for any KNX project and product description. The file normally has the file extension \*.xml, the current name is knx\_master.xml.

For valid owners of the MT (KNX members) it is allowed to use and distribute the KNX XML scheme and the KNX master data file as part of their own tool chain without any legal restrictions. When this KNX XML scheme or the KNX master data is updated, it lies within the responsibility of the tool owner to keep his own tool chain up to date.

The information on any update of KNX XML scheme is provided by KNX via the respective Gitlab project at <https://gitlab.knx.org>, to which KNX Members can request access.

The KNX master data will be updated in ETS on demand (online update capability), the corresponding version can be seen in the ETS overview screen.

### 3 Elements, Types and Attributes

#### 1.1 General

##### 1.1.1 element KNX

Description	Root element of the XML document.				
Children	Name	Description			
	<b>MasterData</b>	Global data created and administered by the KNX Association.			
	<b>ManufacturerData</b>	Data created and administered by the KNX manufacturers.			
	<b>Project</b>	Any number of projects.			
Attributes	Name	Type	Use	Default	Description
	CreatedBy	xs:string	optional		The tool that created this XML file may include its name here. ETS will write "ETS6".
	ToolVersion	xs:string	optional		The tool that created this XML file may include its version here. ETS6 will write "6.0.xxxx.zzzzz" (xxxx is the build number, zzzzz is the changeset).

#### 1.1.2 Enumerations

##### 1.1.2.1 simpleType Access\_t

Type	restriction of <b>xs:string</b>
Description	This enumeration encodes the rights for the ETS user to view and modify parameters.
Facets	enumeration None enumeration Read enumeration ReadWrite

##### 1.1.2.2 simpleType GroupAddressStyle\_t

Type	restriction of <b>xs:string</b>
Description	This enumeration contains the different types of representations of group addresses in ETS6.

Facets	enumeration TwoLevel enumeration ThreeLevel enumeration Free
--------	--

### 1.1.2.3 simpleType SpaceType\_t

Type	restriction of <b>xs:string</b>
Description	This enumeration contains the different types of available spaces in the ETS6.
Facets	enumeration Building enumeration BuildingPart enumeration Floor enumeration Stairway enumeration Room enumeration Corridor enumeration DistributionBoard enumeration Area enumeration Ground enumeration Segment

### 1.1.2.4 simpleType ComObjectPriority\_t

Type	restriction of <b>xs:string</b>
Description	This enumeration lists the possible transmission priorities available in the KNX protocol.
Facets	enumeration Low enumeration High enumeration Alert

### 1.1.2.5 simpleType ComObjectSize\_t

Type	restriction of <b>xs:string</b>
Description	This enumeration lists the possible data sizes for KNX group communication.
Facets	enumeration 1 Bit enumeration 2 Bit enumeration 3 Bit enumeration 4 Bit enumeration 5 Bit enumeration 6 Bit enumeration 7 Bit enumeration 1 Byte enumeration 2 Bytes enumeration 3 Bytes enumeration 4 Bytes enumeration 5 Bytes enumeration 6 Bytes enumeration 7 Bytes enumeration 8 Bytes enumeration 9 Bytes enumeration 10 Bytes enumeration 11 Bytes enumeration 12 Bytes enumeration 14 Bytes enumeration LegacyVarData enumeration 13 Bytes enumeration 15 Bytes enumeration 16 Bytes enumeration 17 Bytes enumeration 18 Bytes



enumeration 19 Bytes  
enumeration 20 Bytes  
enumeration 21 Bytes  
enumeration 22 Bytes  
enumeration 23 Bytes  
enumeration 24 Bytes  
enumeration 25 Bytes  
enumeration 26 Bytes  
enumeration 27 Bytes  
enumeration 28 Bytes  
enumeration 29 Bytes  
enumeration 30 Bytes  
enumeration 31 Bytes  
enumeration 32 Bytes  
enumeration 33 Bytes  
enumeration 34 Bytes  
enumeration 35 Bytes  
enumeration 36 Bytes  
enumeration 37 Bytes  
enumeration 38 Bytes  
enumeration 39 Bytes  
enumeration 40 Bytes  
enumeration 41 Bytes  
enumeration 42 Bytes  
enumeration 43 Bytes  
enumeration 44 Bytes  
enumeration 45 Bytes  
enumeration 46 Bytes  
enumeration 47 Bytes  
enumeration 48 Bytes  
enumeration 49 Bytes

	enumeration 50 Bytes
--	----------------------

### 1.1.2.6 simpleType CompletionStatus\_t

Type	restriction of <b>xs:string</b>
Description	Several elements contain a completion status attribute which might have one of the following values:
Facets	enumeration Undefined enumeration Editing enumeration FinishedDesign enumeration FinishedCommissioning enumeration Tested enumeration Accepted enumeration Locked

### 1.1.2.7 simpleType Enable\_t

Type	restriction of <b>xs:string</b>
Description	This enumeration is used for the group object communication flags.:
Facets	enumeration Enabled enumeration Disabled

### 1.1.2.8 simpleType LdCtrlControlVariable\_t

Type	restriction of <b>xs:string</b>
Description	This enumeration lists the internal variables accessible from the <a href="#">LdCtrlSetControlVariable</a> element
Facets	enumeration EnableSegmentWrite

	enumeration EnableVerifyOnWriteDirect enumeration EnableOptimisticWrite enumeration EnableMemoryAutoVerify
--	--

### 1.1.2.9 simpleType LdCtrlMemAddrSpace\_t

Type	restriction of <b>xs:string</b>
Description	This enumeration lists the memory address spaces available in several memory-related LdCtrl* elements
Facets	enumeration Standard enumeration User enumeration LcSlave enumeration LcFilter

### 1.1.2.10 simpleType LdCtrlProcType\_t

Type	restriction of <b>xs:string</b>
Description	This enumeration contains the possible values for the AppliesTo attribute of the LdCtrl* elements.
Facets	enumeration full enumeration par enumeration grp enumeration full,par enumeration full,grp enumeration par,grp enumeration all enumeration auto

### 1.1.2.11 simpleType LoadProcedureStyle\_t

Type	restriction of <b>xs:string</b>
Description	ETS supports three different mechanism to specify a device load procedure
Facets	enumeration DefaultProcedure enumeration ProductProcedure enumeration MergedProcedure

### 1.1.2.12 simpleType LdCtrlErrorCause\_t

Type	restriction of <b>xs:string</b>
Description	Used to provide richer error messages to the ETS user if something fails during download. A plugin is no longer required for this information.
Facets	enumeration ResourceNotFound enumeration CompareMismatch

### 1.1.2.13 simpleType MemoryType\_t

Type	restriction of <b>xs:string</b>
Description	List of memory technologies
Facets	enumeration RAM enumeration EEPROM enumeration FLASH

### 1.1.2.14 simpleType ProcedureType\_t

Type	restriction of <b>xs:string</b>
Description	List of device configuration procedures

Facets	enumeration Load enumeration Unload
--------	--

### 1.1.2.15 simpleType PropType\_t

Type	restriction of <b>xs:string</b>
Description	List of interface object property types
Facets	enumeration PDT_CONTROL enumeration PDT_CHAR enumeration PDT_UNSIGNED_CHAR enumeration PDT_INT enumeration PDT_UNSIGNED_INT enumeration PDT_KNX_FLOAT enumeration PDT_DATE enumeration PDT_TIME enumeration PDT_LONG enumeration PDT_UNSIGNED_LONG enumeration PDT_FLOAT enumeration PDT_DOUBLE enumeration PDT_CHAR_BLOCK enumeration PDT_POLL_GROUP_SETTINGS enumeration PDT_SHORT_CHAR_BLOCK enumeration PDT_DATE_TIME enumeration PDT_VARIABLE_LENGTH enumeration PDT_GENERIC_01 enumeration PDT_GENERIC_02 enumeration PDT_GENERIC_03 enumeration PDT_GENERIC_04 enumeration PDT_GENERIC_05

enumeration PDT\_GENERIC\_06  
enumeration PDT\_GENERIC\_07  
enumeration PDT\_GENERIC\_08  
enumeration PDT\_GENERIC\_09  
enumeration PDT\_GENERIC\_10  
enumeration PDT\_GENERIC\_11  
enumeration PDT\_GENERIC\_12  
enumeration PDT\_GENERIC\_13  
enumeration PDT\_GENERIC\_14  
enumeration PDT\_GENERIC\_15  
enumeration PDT\_GENERIC\_16  
enumeration PDT\_GENERIC\_17  
enumeration PDT\_GENERIC\_18  
enumeration PDT\_GENERIC\_19  
enumeration PDT\_GENERIC\_20  
enumeration PDT\_UTF-8  
enumeration PDT\_VERSION  
enumeration PDT\_ALARM\_INFO  
enumeration PDT\_BINARY\_INFORMATION  
enumeration PDT\_BITSET8  
enumeration PDT\_BITSET16  
enumeration PDT\_ENUM8  
enumeration PDT\_SCALING  
enumeration PDT\_NE\_VL  
enumeration PDT\_NE\_FL  
enumeration PDT\_FUNCTION

### 1.1.2.16 simpleType ResourceName\_t

Type	restriction of <b>xs:string</b>
------	---------------------------------

Description	List of management resource names; see also RESOURCEID in the eteC SDK documentation <a href="#">[SDK]</a>
Facets	<ul style="list-style-type: none"> <li>enumeration ManagementStyle</li> <li>enumeration DeviceManufacturerId</li> <li>enumeration DeviceBusVoltage</li> <li>enumeration DevicePeiType</li> <li>enumeration GroupAddressTableLoadControl</li> <li>enumeration GroupAddressTableLoadStatus</li> <li>enumeration GroupAddressTablePtr</li> <li>enumeration GroupAddressTable</li> <li>enumeration GroupAssociationTableLoadControl</li> <li>enumeration GroupAssociationTableLoadStatus</li> <li>enumeration GroupAssociationTablePtr</li> <li>enumeration GroupAssociationTable</li> <li>enumeration GroupObjectTablePtr</li> <li>enumeration GroupObjectTable</li> <li>enumeration GroupFilterTablePtr</li> <li>enumeration GroupFilterTable</li> <li>enumeration ApplicationId</li> <li>enumeration ApplicationLoadControl</li> <li>enumeration ApplicationLoadStatus</li> <li>enumeration ApplicationRunControl</li> <li>enumeration ApplicationRunStatus</li> <li>enumeration PeiprogramId</li> <li>enumeration PeiprogramLoadControl</li> <li>enumeration PeiprogramLoadStatus</li> <li>enumeration PeiprogramRunControl</li> <li>enumeration PeiprogramRunStatus</li> <li>enumeration ApplicationPeiType</li> <li>enumeration ReConfig</li> <li>enumeration IndividualAddress</li> </ul>

enumeration DomainAddress  
enumeration FrequencyChannel  
enumeration Sensitivity  
enumeration HardwareConfig1  
enumeration HardwareConfig2  
enumeration HardwareConfig3  
enumeration HardwareConfig4  
enumeration DeviceOrderId  
enumeration DeviceSerialNumber  
enumeration ProgrammingMode  
enumeration PollingGroupSettings  
enumeration ManagementDescriptor01  
enumeration RunError  
enumeration LcConfig  
enumeration LcGrpConfig  
enumeration LcError  
enumeration LcMode  
enumeration GroupObjectTableLoadControl  
enumeration GroupObjectTableLoadStatus  
enumeration GroupAcknowledgeTable  
enumeration HardwareType  
enumeration FirmwareVersion  
enumeration ManufacturerData  
enumeration ApplicationDataPtr  
enumeration PeiprogramDataPtr  
enumeration GroupAddressTableStamp  
enumeration GroupAssociationTableStamp  
enumeration GroupObjectTableStamp  
enumeration GroupFilterTableStamp  
enumeration ApplicationStamp  
enumeration PeiprogramStamp



enumeration	MaxApuLength
enumeration	GroupFilterTableLoadControl
enumeration	GroupFilterTableLoadStatus
enumeration	MainLcConfig
enumeration	SubLcConfig
enumeration	MainLcGrpConfig
enumeration	SubLcGrpConfig
enumeration	CouplServControl
enumeration	MaxRoutingApuLength

### 1.1.2.17 simpleType ResourceAccess\_t

Type	restriction of <b>xs:string</b>
Description	List of access specifiers for Hawk resource descriptions
Facets	enumeration remote enumeration local1 enumeration local2

### 1.1.2.18 simpleType ResourceAccessRights\_t

Type	restriction of <b>xs:string</b>
Description	List of access rights for Hawk resource descriptions
Facets	enumeration None enumeration SystemManufacturer enumeration Manufacturer enumeration Configuration

	enumeration Runtime
--	---------------------

### 1.1.2.19 simpleType ResourceAddrSpace\_t

Type	restriction of <b>xs:string</b>
Description	List of address spaces for Hawk resource descriptions
Facets	enumeration None enumeration StandardMemory enumeration UserMemory enumeration SystemProperty enumeration AppProperty enumeration LcSlaveMemory enumeration LcFilterMemory enumeration ADC enumeration Constant enumeration Pointer enumeration Property enumeration RelativeMemory

### 1.1.2.20 simpleType ResourceMgmtStyle\_t

Type	restriction of <b>xs:string</b>
Description	List of management styles for Hawk resource descriptions
Facets	enumeration simple enumeration lsm

### 1.1.2.21 simpleType ApplicationProgramType\_t

Type	restriction of <b>xs:string</b>
Description	Type of application program
Facets	enumeration ApplicationProgram enumeration PeiProgram

### 1.1.2.22 simpleType RegistrationStatus\_t

Type	restriction of <b>xs:string</b>
Description	Registration status enumeration
Facets	enumeration Unregistered enumeration Registered enumeration Certified enumeration FutureUseNotRecommended enumeration FutureUseNotAllowed

### 1.1.2.23 simpleType ProjectTracingLevel\_t

Type	restriction of <b>xs:string</b>
Description	ProjectTracingLevel enumeration
Facets	enumeration None enumeration OperationUsed enumeration Detailed

### 1.1.2.24 simpleType ToDoStatus\_t

Type	restriction of <b>xs:string</b>
Description	ToDo status enumeration

Facets	enumeration Open enumeration Accomplished
--------	--

### 1.1.2.25 simpleType Capability\_t

Type	restriction of <b>xs:string</b>
Description	Enumeration of capabilities of EtsDataHandler
Facets	enumeration AddDeleteDevice enumeration GroupCommunicationEvents enumerationGroupCommunicationLimits enumerationTransferParameters enumerationProjectCheck EnumerationPrinting

### 1.1.2.26 simpleType ApplicationProgramIPConfig\_t

Type	restriction of <b>xs:string</b>
Description	IPConfig enumeration for the application program
Facets	enumeration Custom enumeration Tool

### 1.1.2.27 simpleType IPConfigAssign\_t

Type	restriction of <b>xs:string</b>
Description	Enumeration describing whether IP configuration is performed automatically or by fixed configuration
Facets	enumeration Fixed enumeration Auto

### 1.1.2.28 simpleType ComTableExpectation\_t

Type	restriction of <b>xs:string</b>
Description	Enumeration describing whether the standard ComTable can be expected. Required for DeviceCompare
Facets	enumeration Yes enumeration No enumeration Try

### 1.1.2.29 simpleType HorizontalAlignment\_t

Type	restriction of <b>xs:string</b>
Description	Enumeration describing whether the picture shall be aligned left, centered or right, or stretched or repeated
Facets	enumeration Left enumeration Middle enumeration Right enumeration Stretch enumeration Repeat

### 1.1.2.30 simpleType TextEncoding\_t

Type	restriction of <b>xs:string</b>
Description	This enum may only contain valid codepages!
Facets	enumeration us-ascii enumeration iso-8859-1 enumeration iso-8859-2 enumeration iso-8859-3 enumeration iso-8859-4 enumeration iso-8859-5 enumeration iso-8859-6

	enumeration iso-8859-7
	enumeration iso-8859-8
	enumeration iso-8859-9
	enumeration iso-8859-10
	enumeration iso-8859-13
	enumeration iso-8859-15
	enumeration utf-8

### 1.1.2.31 simpleType CouplerCapability\_t

Type	restriction of <b>xs:string</b>
Description	This enum represents the different capabilities a coupler can have
Facets	enumeration RfReady enumeration RfMultiFast enumeration RfMultiSlow enumeration SecurityProxy enumeration SegmentCoupler

### 1.1.2.32 simpleType DownloadBehavior\_t

Type	restriction of <b>xs:string</b>
Description	This enum represents the different download behaviors for invisible parameters
Facets	enumeration None enumeration Background enumeration DefaultValue

### 1.1.2.33 simpleType SecurityMode\_t

Type	Restriction of <b>xs:string</b>
------	---------------------------------

Description	This enum represents the different options for secure communication
Facets	enumeration Auto enumeration On enumeration Off

#### 1.1.2.34 simpleType ComObjectSecurityRequirements\_t

Type	Restriction of <b>xs:string</b>
Description	<p>This enum represents the different options for the required security for ComObjects.</p> <p>The ETS6 does not distinguish Auth and AuthAndConf and will treat both enum values equally. Any other value than None means that security is required.</p> <p>Manufacturer can already define, which security level their products require, but only future ETS-Versions will distinguish those values.</p> <p>Auth: The ComObject may only communicate with authenticated partners. (Authentication required)</p> <p>AuthAndConf: The ComObject may only communicate with authenticated partners and the communication must be encrypted (Authentication and Confidentiality)</p>
Facets	enumeration None enumeration Auth enumeration AuthAndConf

#### 1.1.2.35 simpleType CellRef\_t

Type	Restriction of <b>xs:string</b>
Description	Required for non-standard layout of parameters as tabular display. This represents the position in the table, given as "row,col" (both 1-based!). See [PSR] 2.1.1
Facets	pattern \d+,\d+

#### 1.1.2.36 simpleType ParameterBlockLayout\_t

Type	Restriction of <b>xs:string</b>
Description	Possible layout types of a parameter block. See [PSR] 2.1.1

Facets	enumeration Table enumeration Grid enumeration List
--------	---

### 1.1.2.37 simpleType DeprecationStatus\_t

Type	Restriction of <b>xs:string</b>
Description	Enum that can be used to disable DatapointRoles, SpaceUsages, FunctionTypes or FunctionsGroups.
Facets	enumeration active enumeration deprecated enumeration removed

### 1.1.2.38 simpleType ModuleDefArgType\_t

Type	Restriction of <b>xs:string</b>
Description	Enum that can be used to define the argument in a module definition. Required for modular application programs.
Facets	enumeration Numeric enumeration Text enumeration AllocatorRef

### 1.1.2.39 simpleType MemberStatus\_t

Type	Restriction of <b>xs:string</b>
Description	Enum that can be used to declare active and inactive members of the KNX
Facets	enumeration Active enumeration Inactive

### 1.1.2.40 simpleType RFRxCapabilities\_t

Type	restriction of <b>xs:string</b>
------	---------------------------------



Description	This enum represents the different capabilities a
Facets	enumeration Ready enumeration ReadyFast enumeration Slow

#### 1.1.2.41 simpleType RFTxCapabilities\_t

Type	restriction of <b>xs:string</b>
Description	This enum represents the different capabilities a
Facets	enumeration Ready enumeration ReadyFast enumeration ReadFastSlow

#### 1.1.2.42 simpleType ProjectType\_t

Type	restriction of xs:string
Description	User interface specific (icons only) enumeration for a better classification of projects:
Facets	enumeration Apartment enumeration Family House enumeration Villa enumeration Other (Residential) enumeration Hotel enumeration Airport enumeration Office Building enumeration Educational enumeration Leisure enumeration Entertainment enumeration Public Building enumeration Health Care

	enumeration Other (Commercial) enumeration Manufacturer enumeration City Project enumeration Transportation enumeration Other (Other)
Type	restriction of <b>xs:string</b>
Description	This enum represents the different capabilities a
Facets	enumeration Ready enumeration ReadyFast enumeration ReadFastSlow

### 1.1.3 Other simpleTypes

#### 1.1.3.1 simpleType IDREF

Type	<b>xs:NCName</b>
Description	This type is used for references to xs:ID. In contrast to the standard XML IDREF type, the referenced element need not be in the same XML file.

#### 1.1.3.2 simpleType IDREFS

Type	<b>xs:list of knx:IDREF</b>
Description	This type is used for multiple references to xs:ID, separated by space. In contrast to the standard XML IDREFS type, the referenced elements need not be in the same XML file.

#### 1.1.3.3 simpleType RELIDREF

Type	<b>xs:NCName</b>
------	------------------

Description	This type is used for references to elements below a known application program, e.g. instead of the IDREF "M-0004_A-104E-01-5221-O000A_O-2_R-199", the RELIDREF is shortened to "O-2_R-199".
-------------	--

#### 1.1.3.4 simpleType RELIDREFS

Type	<b>xs:list of knx:RELIDREF</b>
Description	This type is used for multiple references to knx:RELIDREF, separated by space.

#### simpleType LanguageDependentIDREF

Type	<b>xs:NCName</b>
Description	This type is used for references to language dependent xs:ID. In contrast to the standard XML IDREF type, the referenced element need not be in the same XML file.

#### 1.1.3.5 simpleType Capabilities\_t

Type	<b>xs:list of knx:Capability_t</b>
Description	Used to list the actions, an EtsDataHandler is capable of.

#### 1.1.3.6 simpleType String20\_t

Type	<b>xs:string</b>
Description	Same as xs:string, but restricted to 20 unicode characters.

#### 1.1.3.7 simpleType String50\_t

Type	<b>xs:string</b>
Description	Same as xs:string, but restricted to 50 unicode characters.

### 1.1.3.8 simpleType String255\_t

Type	<b>xs:string</b>
Description	Same as xs:string, but restricted to 255 unicode characters.

### 1.1.3.9 simpleType Identifier50\_t

Type	restriction of <b>xs:string</b>
Description	This type is for specifying the name of ModuleDef\Arguments\Argument.
Facets	pattern [A-Za-z_][A-Za-z0-9_]

### 1.1.3.10 simpleType LanguageDependentString\_t

Type	<b>xs:string</b>
Description	This type is used for texts in master or product data that may be translated to different languages.

### 1.1.3.11 simpleType LanguageDependentString20\_t

Type	<b>xs:LanguageDependentString_t</b>
Description	Same as LanguageDependentString_t, but restricted to 20 unicode characters.

### 1.1.3.12 simpleType LanguageDependentString50\_t

Type	<b>xs:LanguageDependentString_t</b>
Description	Same as LanguageDependentString_t, but restricted to 50 unicode characters.

### 1.1.3.13 simpleType LanguageDependentString255\_t

Type	<b>xs:LanguageDependentString_t</b>
Description	Same as LanguageDependentString_t, but restricted to 255 unicode characters.

### 1.1.3.14 simpleType Regex\_t

Type	<b>xs:string</b>
Description	Same as string, but must obey the rules of a .NET Regex.

### 1.1.3.15 simpleType AccessLevel\_t

Type	restriction of <b>xs:unsignedByte</b>
Description	This type is for specifying the segment access level in <a href="#">LdCtrlDeclarePropDesc</a> .
Facets	minInclusive 0 maxInclusive 15

### 1.1.3.16 simpleType FloatFormat\_t

Type	restriction of <b>xs:string</b>
Description	This type is for specifying the DisplayFormat of a Parameter of Type TypeFloat
Facets	[#,]*[0,]+(\.0*)?([eE][+-]?0+)?[#,]*[0,]+(\.0*)?([eE][+-]?0+)?

### 1.1.3.17 simpleType BitOffset\_t

Type	restriction of <b>xs:unsignedByte</b>
Description	This type is for specifying the bit offset of parameters.

	The bit offset is the distance of the most significant bit of the parameter from the most significant bit of the first octet in memory.
Facets	minInclusive 0 maxInclusive 7

### 1.1.3.18 simpleType Condition\_t

Type	<b>xs:string</b>									
Description	<p>This type is for specifying conditions in <a href="#">When_t</a>.</p> <p>The following values are possible (<i>number</i> is an integer value written in decimal notation, <b>()?+*</b> are the usual EBNF symbols, <b>□</b> denotes the space character):</p> <table border="0"> <tr> <td>A single number</td> <td><i>number</i></td> <td>The condition evaluates to true, if the value of the controlling parameter is numerically equal to the given number.</td> </tr> <tr> <td>Space-separated list of numbers</td> <td><i>number</i> ( <b>□</b> <i>number</i> )*</td> <td>The condition evaluates to true, if the value of the controlling parameter is numerically equal to any one of the given numbers.</td> </tr> <tr> <td>Comparison expressions</td> <td><i>op number</i></td> <td>Compares the value of the controlling parameter to the given number using one of the comparison operators: = != &gt; &lt; &gt;= &lt;= (note that &lt; &gt; have to be written as &amp;lt; / &amp;gt; in XML attributes)</td> </tr> </table> <p>The controlling parameter must be of type TypeNumber or TypeRestriction. In the latter case, the Value attribute is used in the comparison.</p> <p>The planned MT may accept (on import only) also names instead of numbers if the parameter is of type TypeRestriction. But at latest when the data is submitted for registration, these have to be replaced by numeric values since otherwise the registration signature will get invalid on an XML → DB → XML round trip.</p>	A single number	<i>number</i>	The condition evaluates to true, if the value of the controlling parameter is numerically equal to the given number.	Space-separated list of numbers	<i>number</i> ( <b>□</b> <i>number</i> )*	The condition evaluates to true, if the value of the controlling parameter is numerically equal to any one of the given numbers.	Comparison expressions	<i>op number</i>	Compares the value of the controlling parameter to the given number using one of the comparison operators: = != > < >= <= (note that < > have to be written as &lt; / &gt; in XML attributes)
A single number	<i>number</i>	The condition evaluates to true, if the value of the controlling parameter is numerically equal to the given number.								
Space-separated list of numbers	<i>number</i> ( <b>□</b> <i>number</i> )*	The condition evaluates to true, if the value of the controlling parameter is numerically equal to any one of the given numbers.								
Comparison expressions	<i>op number</i>	Compares the value of the controlling parameter to the given number using one of the comparison operators: = != > < >= <= (note that < > have to be written as &lt; / &gt; in XML attributes)								

### 1.1.3.19 simpleType Value\_t

Type	<b>xs:string</b>								
Description	<p>This type is for storing parameter or module argument values. The different parameter types or module argument values are encoded as follows:</p> <table border="0"> <tr> <td>TypeNone</td> <td>Always the empty string.</td> </tr> <tr> <td>TypeText</td> <td>The text value, suitably escaped by character references (e.g. &amp;#x9; for the tab character) or entity references (e.g. &amp;lt; instead of &lt;). Note that all whitespace characters (newline, tab etc.) must be written as character references, otherwise input normalization would replace them by space characters.</td> </tr> <tr> <td>TypeNumber</td> <td>The numeric value, formatted as decimal string.</td> </tr> <tr> <td>TypeFloat</td> <td>The numeric value, formatted in scientific notation, with 16 significant digits and 3 exponent digits (regular expression: "-?\d\.\d{15}E[+-]\d{3}"). This corresponds to the conversion value.ToString("E15", CultureInfo.InvariantCulture) in C#.</td> </tr> </table> <p>Note: if a Value_t attribute would ever be registration-relevant, care must be taken to ensure that this attribute is reproduced exactly on all data transformations, e.g. when importing the XML into an ETS 4 database and exporting it again.</p>	TypeNone	Always the empty string.	TypeText	The text value, suitably escaped by character references (e.g. &#x9; for the tab character) or entity references (e.g. &lt; instead of <). Note that all whitespace characters (newline, tab etc.) must be written as character references, otherwise input normalization would replace them by space characters.	TypeNumber	The numeric value, formatted as decimal string.	TypeFloat	The numeric value, formatted in scientific notation, with 16 significant digits and 3 exponent digits (regular expression: "-?\d\.\d{15}E[+-]\d{3}"). This corresponds to the conversion value.ToString("E15", CultureInfo.InvariantCulture) in C#.
TypeNone	Always the empty string.								
TypeText	The text value, suitably escaped by character references (e.g. &#x9; for the tab character) or entity references (e.g. &lt; instead of <). Note that all whitespace characters (newline, tab etc.) must be written as character references, otherwise input normalization would replace them by space characters.								
TypeNumber	The numeric value, formatted as decimal string.								
TypeFloat	The numeric value, formatted in scientific notation, with 16 significant digits and 3 exponent digits (regular expression: "-?\d\.\d{15}E[+-]\d{3}"). This corresponds to the conversion value.ToString("E15", CultureInfo.InvariantCulture) in C#.								

TypeRestriction	The Value attribute of the selected <a href="#">Enumeration</a> option.
TypeTime	Same as TypeNumber
TypeDate	yyyy-mm-dd
TypeIPAddress	IPv4 addresses: decimal dotted notation IPv6 addresses: eight groups of four hexadecimal digits, separated by colons, e.g. 2001:0db8:85a3:0000:0000:8a2e:0370:7334
TypeAllocatorRefId	A module allocator refId as string

### 1.1.3.20 simpleType Guid\_t

Type	restriction of <b>xs:string</b>
Description	This type is for specifying GUIDs, e.g. the CLSIDs of Plugins.
Facets	pattern <code>\{[0-9A-F]{8}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{12}\}</code>

### 1.1.3.21 simpleType Ipv4Address\_t

Type	restriction of <b>xs:string</b>
Description	This type is for specifying IP v4 addresses, e.g. the IP routing multicast address.
Facets	pattern <code>((25[0-5] 2[0-4][0-9] 1[0-9][0-9] [1-9][0-9] [0-9])\.){3}(25[0-5] 2[0-4][0-9] 1[0-9][0-9] [1-9][0-9] [0-9])</code>

### 1.1.3.22 simpleType RegistrationNumber\_t

Type	restriction of <b>xs:string</b>
Description	This type is for specifying registration numbers in the format yyyy/n
Facets	pattern <code>\d{4}/\d+</code>

### 1.1.3.23 simpleType HardwareVersionNumber\_t

Type	restriction of <b>xs:unsignedShort</b>
Description	This type is for specifying the VersionNumber of a hardware. Restricted to ensure compatibility with ETS3
Facets	minInclusive 0 maxInclusive 32767

### 1.1.3.24 simpleType Aes128Key\_t

Type	<b>xs:string</b>
Description	Same as xs:string, but restricted to 40 characters. Used to represent a base64-encoded string of an AES128 key.

### 1.1.3.25 simpleType AccessPolicy\_t

Type	restriction of <b>xs:string</b>
Description	This type is for specifying access policies for interface object properties.
Facets	pattern [0-3][0-9A-F]{2}/[0-3][0-9A-F]{2}

### 1.1.3.26 simpleType RepeatIndex\_t

Type	restriction of <b>xs:string</b>
Description	This type is for specifying the repeat index of a module
Facets	pattern \d+\d+

## 1.2 Project Data

### 1.2.1 element KNX/Project

Description	Contains a project.
Type	<b><u>knx:Project_t</u></b>



## 1.2.2 complexType Project\_t

Description	Contains a project.				
Children	Name	Description			
	<b>ProjectInformation</b>	Contains general information about the project.			
	<b>Installations</b>	Contains the list of installations within the project.. Most project will just have one Installation. Count of installations must be in [1...16].			
	<a href="#">AddinData</a>	Contains project related data for Addins			
	UserFiles	Contains the user files that are appended to the project			
Attributes	Name	Type	Use	Default	Description
	Id	xs:ID	required		Unique ID of the project in the knxproj container. On export or conversion, this will be constructed as <b>P-<i>nnnn</i></b> , where: <i>nnnn</i> Random 16Bit Identifier, formatted as 4 hexadecimal digits . Must be unique in the knxproj container.

### 1.2.2.1 element Project\_t/UserFiles

Description	Contains the Userfiles				
Type	<b><u>knx:Userfiles_t</u></b>				

### 1.2.2.2 complexType UserFile\_t

Description	An element of the Userfile				
Attributes	Name	Type	Use	Default	Description
	Filename	knx:string255_t	required		The name of the user file
	Comment	xs:string	optional		A comment for the user file

## 1.2.3 General

### 3.1.1.1 element Project\_t/ProjectInformation

Description	Contains general information about the project.				
Children	Name	Description			
	<b>HistoryEntries</b>	Contains history entries entered by the user.			
	<a href="#">ToDoItems</a>	Contains project related ToDo notes			
	<a href="#">ProjectTraces</a>	Contains the ProjectTraces			
	<a href="#">DeviceCertificates</a>	Contains the DeviceCertificates			
Attributes	Name	Type	Use	Default	Description
	Name	knx:String50_t	required		Project Name
	GroupAddressStyle	<a href="#">knx:GroupAddressStyle_t</a>	required		Representation of group addresses in this project
	ProjectNumber	knx:String50_t	optional		Optional project number
	ContractNumber	knx:String50_t	optional		Optional contract number
	LastModified	xs:dateTime	optional		Date and time of last modification (UTC)
	ArchivedVersion	xs:dateTime	optional		LastModified timestamp of the project that was checked out of the project archive
	ProjectStart	xs:dateTime	optional		Date of project start (UTC)
	ProjectEnd	xs:dateTime	optional		Date of schedules project end (UTC)
	ProjectType	knx:ProjectType_t		Other (Commercial)	The enumeration for better classification of projects. This is user interface specific.
	ProjectId	xs:unsignedShort	optional		KNXnet/IP project ID [0 ... 4095]. Not used for other media. See KNX standard, Volume 3, Part 8, Chapter 2.
	Comment	xs:string	optional		Optional comment
	CompletionStatus	<a href="#">knx:CompletionStatus_t</a>	optional	Undefined	Completion status
	ProjectTracingLevel	<a href="#">knx:ProjectTracingLevel_t</a>	optional	None	The Level for ProjectTraces
	ProjectTracingPassword	<a href="#">knx:String20_t</a>	optional		The password for ProjectTracing. This is stored as the first 20 characters of the Base64 encoded string of the salted hash of the original password. "PT-" is used as salt.
	Hide16BitGroupsFromLegacyPlugins	xs:boolean	optional	false	If true, the project will not use 16 bit groups. This will prevent problems with older plugins that only support 15 bit groups.
CodePage	knx:TextEncoding_t	optional		Optional CodePage for correct encoding of project related texts.	

BusAccessLegacyMode	xs:Boolean	optional	false	Determines the mode of the buss access
Guid	xs:string	required		The project guid, used to secure the project data
LastUsedPuid	xs:int	required		The highest puid that is so far used in the project
Security	knx:SecurityMode_t	optional	Auto	Flag to indicate how project shall handle security: On -> each secure enabled device must be used securely Off -> no secure enabled device may be used securely Auto -> let the user decide

### 3.1.1.2 element Project\_t/ProjectInformation/Tags

Description	List of user defined tags
Children	Name Description <b><u>Tag</u></b>

### 3.1.1.3 element Project\_t/ProjectInformation/Tags/Tag

Description	User defined tag															
Attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Text</td> <td>knx:String20_t</td> <td>required</td> <td></td> <td>The displayed text for the tag.</td> </tr> <tr> <td>Color</td> <td>xs:string</td> <td>required</td> <td></td> <td>The RGB color code for displaying the tag. Pattern is #[0-9A-F]{6}.</td> </tr> </tbody> </table>	Name	Type	Use	Default	Description	Text	knx:String20_t	required		The displayed text for the tag.	Color	xs:string	required		The RGB color code for displaying the tag. Pattern is #[0-9A-F]{6}.
Name	Type	Use	Default	Description												
Text	knx:String20_t	required		The displayed text for the tag.												
Color	xs:string	required		The RGB color code for displaying the tag. Pattern is #[0-9A-F]{6}.												

### 1.2.3.1 element Project\_t/ProjectInformation/HistoryEntries

Description	List of history entries entered by the user
Children	Name Description <b><u>HistoryEntry</u></b>

### 1.2.3.2 element Project\_t/ProjectInformation/HistoryEntries/HistoryEntry

Description	History entries entered by the user				
Attributes	Name	Type	Use	Default	Description
	Date	xs:dateTime	required		Date and time of the history entry (UTC)
	User	knx:String255_t	optional		User name (optional)
	Text	xs:string	required		Text of the history entry
	Detail	xs:string	optional		Detailed text for the entry

### 1.2.3.3 element Project\_t/ProjectInformation/ProjectTraces

Description	Contains the ProjectTraces				
Type	<b><u>knx:ProjectTraces_t</u></b>				

### 1.2.3.4 complexType ProjectTrace\_t

Description	An element of the ProjectTrace				
Attributes	Name	Type	Use	Default	Description
	Date	xs:datetimerequired			The date and time of the trace's creation
	UserName	xs:string	required		The name of the user
	Comment	xs:string	required		The text of the trace

### 1.2.3.5 element Project\_t/ProjectInformation/DeviceCertificates

Description	Contains the DeviceCertificates				
Type	<b><u>knx:DeviceCertificates_t</u></b>				

### 1.2.3.6 complexType DeviceCertificate\_t

Description	An element of the DeviceCertificate				
Attributes	Name	Type	Use	Default	Description
	SerialNumber	xs:base64Binary	required		The serial number of the device

	FDSK	knx:Aes128Key_t	required	The factory default setup key of the device
--	------	-----------------	----------	---

### 1.2.3.7 element Project\_t/ProjectInformation/ToDoItems

Description	Contains the ToDoItems			
Type	<u>knx:ToDoItems</u> t			

### 1.2.3.8 complexType ToDoItem\_t

Description	An element of the ToDoItem			
Attributes	Name	Type	Use	Default Description
	Description	xs:string	required	The description of the item
	ObjectPath	xs:string	optional	The path to the object
	Status	knx:ToDoStatus_t	required	The status of the ToDoItem, either "Open" or "Accomplished"

### 1.2.3.9 element Project\_t/AddinData

Description	List of AddinData			
-------------	-------------------	--	--	--

### 1.2.3.10 complexType AddinData\_t

Description	An element of the AddinData			
Attributes	Name	Type	Use	Default Description
	Name	knx:String50_t	required	The name of the Addin
	AddinId	xs:ID	required	The identifier of the Addin

### 1.2.3.11 complexType BusAccess\_t

Description	The information for the bus access			
Attributes	Name	Type	Use	Default Description
	Name	xs:string	required	The name of the access
	Edi	knx:Guid_t	optional	The Guid of the access type. If no Edi specified, the Parameter contains the FalconConnectionString

	Parameterxs:string required	The parameters necessary for the connection
--	-----------------------------	---

### 1.2.3.12 element Project\_t/Installations

Description	Contains the list of installations within the project.	
Children	Name	Description
	<b><u>Installation</u></b>	Up to 16 installations

### 1.2.3.13 element Project\_t/Installations/Installation

Description	Contains data for one installation				
Children	Name	Description			
	<b><u>Topology</u></b>	Contains the topology structure and device data			
	<b><u>Buildings</u></b>	Contains the building structure			
	<b><u>GroupAddresses</u></b>	Contains the group address structure			
	<b><u>Trades</u></b>	Contains the trades structure			
	<b><u>SplitInfos</u></b>	Contains the split infos for the installation			
Attributes	Name	Type	Use	Default	Description
	Name	knx:String50_t	required		Name of the installation. If the project contains just one installation, this can be set to an empty string
	InstallationId	xs:unsignedShort	optional		KNXnet/IP installation ID [0...15]; not used for other media. See KNX standard, Volume 3, Part 8, Chapter 2
	BCUKey	xs:unsignedLong	optional	4294967295	The key used to lock devices supporting authentication.
	IPRoutingMulticastAddress	<a href="#">knx:Ipv4Address_t</a>	optional	224.0.23.12	The multicast address for IP communication.
	MulticastTTL	xs:byte	optional	16	The time to live for multicast telegrams, i.e.the number of routers the telegram may pass before deletion.
	IPRoutingBackboneKey	knx:Aes128Key_t	optional		For symmetric encryption the AES algorithm with a key length of 128 bit is used. For every IP multicast group, a single encryption key is used. This key is stored in every device of the IP multicast group and has an unlimited lifetime.
	IPRoutingLatencyTolerance	xs:unsignedShort	optional		To prevent replay attacks, the devices shall only accept IP telegrams that were received within a specified time after the telegram was sent. This tolerance can be specified by the user. The latency tolerance is specified in milliseconds.

IPSyncLatencyFraction	xs:float	optional	0.1	To define the latency for secure IP communication. For further information, please see KSG 616
IPRoutingBackboneSecurity	knx:IPRoutingBackboneSecurity	optional	Auto	Specifies if the communication via IP is secure or not. Can be either Auto, On or Off. On means the IP communication is performed securely, Off means the IP communication is performed normally. Auto means: If every IP device in the installation has an ApplicationProgram with IsSecureEnabled == true, the communication is performed securely.
DefaultLine	xs:string	optional		The RefId of the default line.
CompletionStatus	<a href="#">knx:CompletionStatus</a>	optional	Undefined	Completion status
SplitType	<a href="#">xs:string</a>	optional		Completion status
Context	<a href="#">xs:string</a>	optional		Optional semantics information for IoT project.

## 1.2.4 Topology

### 1.2.4.1 element Project\_t/Installations/Installation/Topology

Description	Contains the topology structure and device data
-------------	---

### 1.2.4.2 complexType Topology\_t

Description	Contains the topology structure and device data						
Children	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>Area</b></td> <td>Up to 16 Areas</td> </tr> <tr> <td><b>UnassignedDevices</b></td> <td>List of devices not assigned to a line</td> </tr> </tbody> </table>	Name	Description	<b>Area</b>	Up to 16 Areas	<b>UnassignedDevices</b>	List of devices not assigned to a line
Name	Description						
<b>Area</b>	Up to 16 Areas						
<b>UnassignedDevices</b>	List of devices not assigned to a line						

### 1.2.4.3 element Topology\_t/Area

Description	Description of a KNX area										
Children	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>Line</b></td> <td>Up to 16 lines</td> </tr> </tbody> </table>					Name	Description	<b>Line</b>	Up to 16 lines		
Name	Description										
<b>Line</b>	Up to 16 lines										
Attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Id</td> <td>xs:ID</td> <td>required</td> <td></td> <td>Unique ID.</td> </tr> </tbody> </table>	Name	Type	Use	Default	Description	Id	xs:ID	required		Unique ID.
Name	Type	Use	Default	Description							
Id	xs:ID	required		Unique ID.							

			<p>On export or conversion, this will be constructed as <i>parid_A-number</i>, where:</p> <p><i>parid</i> ID of the parent Project and InstallationID sepearted with '-'</p> <p><i>number</i> Unique number of the area within the project. This does not reflect the area address! For converted projects, this corresponds to Area.UniqueNumber in the database schema.</p>
Name	knx:String255_t	optional	Name of the area
Address	xs:int	required	Area address [0...15]
Comment	xs:string	optional	User comment
CompletionStatus	<a href="#">knx:CompletionStatus_t</a>	optional	Completion status
Description	xs:string	optional	Description of the area
Puid	xs:int	required	The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.

#### 1.2.4.4 element Topology\_t/Area/Line

Description	Description of a KNX line				
Children	Name	Description			
	<b>Segment</b>	Up to 128 segments			
Attributes	Name	Type	Use	Default	Description
	Id	xs:ID	required		<p>Unique ID.</p> <p>On export or conversion, this will be constructed as <i>parid_L-number</i>, where:</p> <p><i>parid</i> ID of the parent Project and InstallationID sepearted with '-'</p> <p><i>number</i> Unique number of the line within the project. This does not reflect the line address! For converted projects, this corresponds to Line.UniqueNumber in the database schema.</p>
	Name	knx:String255_t	optional		Name of the line
	Address	xs:int	required		Line address [0...15]
	Comment	xs:string	optional		User comment
	CompletionStatus	<a href="#">knx:CompletionStatus_t</a>	optional		Completion status
	Description	xs:string	optional		Description of the line
	Puid	xs:string	required		The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.



### 1.2.4.1 element Topology\_t/Area/Line/Segment

Description	Description of a KNX line segment				
Children	Name	Description			
	<b>DeviceInstance</b>	List of devices assigned to the line segment.			
	<b>AdditionalGroupAddresses</b>	List of additional group addresses that should be included in the filter table of this segment's line coupler.			
	<b>BusAccess</b>	Contains the bus access information for the line segment			
Attributes	Name	Type	Use	Default	Description
	Id	xs:ID	required		Unique ID.  On export or conversion, this will be constructed as <i>parid_S-number</i> , where: <i>parid</i> ID of the parent Project and InstallationID sepearted with '-' <i>number</i> Unique number of the segment within the project. This does not reflect the segment number!
	Name	knx:String255_t	optional		Name of the line
	Number	xs:int	required		Segment number [0...127]
	Comment	xs:string	optional		User comment
	MediumTypeRefId	knx:IDREF	required		Medium type of the segment, a reference to <a href="#">MediumType</a> .
	DomainAddress	xs:unsignedLong	optional		For open media (PL, RF), the domain address
	CompletionStatus	<a href="#">knx:CompletionStatus_t</a>	optional		Completion status
	Description	xs:string	optional		Description of the line segment
Puid	xs:string	required		The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.	

### 1.2.4.2 element Topology\_t/Area/Line/Segment/DeviceInstance

Description	Represents a device in the project.
Type	<b><u>knx:DeviceInstance_t</u></b>

### 1.2.4.3 element Topology\_t/Area/Line/Segment/AdditionalGroupAddresses

Description	List of additional group addresses that should be included in the filter table of this line's line coupler.				
Children	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>GroupAddress</b></td> <td>GroupAddress that is not necessarily contained in the project</td> </tr> </tbody> </table>	Name	Description	<b>GroupAddress</b>	GroupAddress that is not necessarily contained in the project
Name	Description				
<b>GroupAddress</b>	GroupAddress that is not necessarily contained in the project				

### 1.2.4.4 element Topology\_t/Area/Line/Segment/AdditionalGroupAddresses/GroupAddress

Description											
Attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Address</td> <td>xs:unsignedShort</td> <td>required</td> <td></td> <td>The address of the <a href="#">GroupAddress</a></td> </tr> </tbody> </table>	Name	Type	Use	Default	Description	Address	xs:unsignedShort	required		The address of the <a href="#">GroupAddress</a>
Name	Type	Use	Default	Description							
Address	xs:unsignedShort	required		The address of the <a href="#">GroupAddress</a>							

### 1.2.4.5 element Topology\_t/UnassignedDevices

Description	List of devices not assigned to a line				
Children	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>DeviceInstance</b></td> <td>List of devices assigned to no line.</td> </tr> </tbody> </table>	Name	Description	<b>DeviceInstance</b>	List of devices assigned to no line.
Name	Description				
<b>DeviceInstance</b>	List of devices assigned to no line.				

### 1.2.4.6 element Topology\_t/UnassignedDevices/DeviceInstance

Description	Represents a device in the project.
Type	<b><u>knx:DeviceInstance t</u></b>

## 1.2.5 Device Data

### 1.2.5.1 complexType DeviceInstance\_t

Description	Represents a device in the project.
-------------	-------------------------------------

Children	Name	Description
	<b><u>ParameterInstanceRefs</u></b>	List of parameter instances with non-default values
	<b><u>ComObjectInstanceRefs</u></b>	List of group communication object instances
	<b><u>ChannelInstances</u></b>	List of channel instances.
	<b><u>ModuleInstances</u></b>	List of module instances.
	<b><u>GroupObjectTree</u></b>	The structured content of the group object tree. This contains the channels and folders, along with the active group objects.
	<b><u>AdditionalAddresses</u></b>	Additional individual addresses of the device
	<b><u>BinaryData</u></b>	For use by plugins
	<b><u>IPConfig</u></b>	The IP configuration of the device
	<b><u>Security</u></b>	The security configuration of the device
	<b><u>BusInterfaces</u></b>	The bus interfaces of the device
	<b><u>RFFastAckSlots</u></b>	The slots for fast RF acks

Attributes	Name	Type	Use	Default	Description
	Id	xs:ID	required		<p>Unique ID.</p> <p>On export or conversion, this will be constructed as <i>parid_DI-number</i>, where:</p> <p><i>parid</i> ID of the parent Project and InstallationID separated with '-'</p> <p><i>number</i> Unique number of the area within the project. This does not reflect the device address! For converted projects, this corresponds to DeviceInstance.UniqueNumber in the database schema.</p>
	Name	knx:String255_t	optional		Device name
	ProductRefId	knx:IDREF	required		Reference to a <a href="#">Product</a> ; must be a child of the Hardware2Program element
	Hardware2ProgramRefId	knx:IDREF	optional		Reference to a <a href="#">Hardware2Program</a>
	Address	xs:int	optional		Device address [0...255]
	Comment	xs:string	optional		Device comment
	LastModified	xs:dateTime	optional		Date/time of last modification (UTC)
	LastDownload	xs:dateTime	optional		Date/time of last download (UTC)
	LastUsedAPDULength	xs:unsignedShort	optional		
	ReadMaxAPDULength	xs:unsignedShort	optional		
	ReadMaxRoutingAPDULength	xs:unsignedShort	optional		
	InstallationHints	xs:string	optional		Installation hints, may be plain text or RTF text

CompletionStatus	<a href="#">knx:CompletionStatus_t</a>	optional	Undefined	Completion status
IndividualAddressLoaded	xs:boolean	optional	false	true if the IA has been programmed
ApplicationProgramLoaded	xs:boolean	optional	false	true if the application program has been programmed
ParametersLoaded	xs:boolean	optional	false	true if the parameters has been programmed
CommunicationPartLoaded	xs:boolean	optional	false	true if the group communication part has been programmed
MediumConfigLoaded	xs:boolean	optional	false	true if the PL medium configuration has been programmed
LoadedImage	xs:base64Binary	optional		The image loaded into the device the last time (used with differential download)
Checksums	xs:base64Binary	optional		Check sums read from the device the last time (used with differential download)
Description	xs:string	optional		Device description.
DownloadCounter	xs:unsignedInt	optional		
IsActivityCalculated	xs:boolean	optional		If the <b>IsActivityCalculated</b> flag exists at the DeviceInstance and is “true”, the activity for the DeviceInstance is already claculated
Broken	xs:boolean	optional	false	true if the OnImport handler failed. A broken application program cannot be used in the ETS4.
SerialNumber	xs:base64Binary	optional		The SerialNumber is used for DownloadIndividualAddressBySerialNumber. This serial number must be provided base64 encoded.
UniqueId	knx:Guid_t	optional		The unique identifier for the device instance. This is set, if an AddIn requests the identifier and the device instance has none set so far. Otherwise, this unique identifier remains null..
IsRFRe transmitter	xs:boolean	optional		True if the device instance shall act as a RF re transmitter
Puid	xs:string	required		The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.

### 1.2.5.2 complexType IPConfig\_t

Description	IP configuration for the DeviceInstance				
Attributes	Name	Type	Use	Default	Description
	Assign	<a href="#">knx:IPConfigAssign_t</a>	optional	Auto	If the value is 'Auto', the IP configuration is fetched from DHCP, if the value is 'Fixed', the IP configuration is performed manually
	IPAddress	knx:Ipv4Address_t	optional		The IP address of the IP device
	SubnetMask	knx:Ipv4Address_t	optional		The subnet mask of the IP device
	DefaultGateway	knx:Ipv4Address_t	optional		The default gateway of the IP device
	MACAddress	knx:String50_t	optional		The MAC address of the IP device

### 1.2.5.3 complexType Security\_t

Description	Configuration for security elements				
Children	Name	Description			
	<a href="#">Role</a>	The security role of the device.			
Attributes	Name	Type	Use	Default	Description
	LoadedIPRoutingBackboneKey	knx:Aes128Key_t	optional		After the download of a device, the encryption key of the IP multicast group is written to the device. The user cannot set the key manually. This encryption key is used for the symmetric encryption within the IP multicast group.
	DeviceAuthenticationCode	knx:String20_t	optional		The device authentication code is generated when the device is instantiated .
	DeviceAuthenticationCodeHash	xs:base64Binary	optional		A hash of the device authentication code.
	LoadedDeviceAuthenticationCodeHash	xs:base64binaryt	optional		A hash of the device authentication code that was used with the last device downloaded.
	DeviceManagementPassword	knx:String20_t	optional		The management password is generated when the device is instantiated. The initial password has a length of 8 elements and consists of lower and upper case letters, numbers and the special characters "+", "-", ":", ";", "#", and "*". The device management password can be changed by the user anytime.
	DeviceManagementPasswordHash	xs:base64Binary	optional		A hash of the device management password.
	LoadedDeviceManagementPasswordHash	xs:base64Binary	optional		A hash of the device management password that was used with the last device download.
	ToolKey	knx:Aes128Key_t	optional		The tool key for the device.
	LoadedToolKey	knx:Aes128Key_t	optional		The tool key used with the last device download.
	SequenceNumber	xs:unsignedLong	optional		The value of the last received sender counter. The SequenceNumber is updated during secure online communication.
	SequenceNumberTimestamp	xs:dateTime	optional		The timestamp of the last sequence number. This could be used to check how trustworthy a sequence number is.
UnicastBroadcastBlocking	knx:SecurityMode_t	optional	Auto	Valid for devices with CouplerCapabilities including "SecurityProxy"	

### 1.2.5.4 element Security\_t/Role

Description	Group addresses assigned to a ComObjectInstanceRef for sending (and receiving)				
Attributes	Name	Type	Use	Default	Description
	RefId	knx:IDREF	required		Reference to the DataSecurity role defined in the application program.
	Address	xs:unsignedByte	required		The individual address used for this role.

### 1.2.5.5 element DeviceInstance\_t/BusInterfaces

Description	Contains bus interfaces for the device				
Children	Name	Description			
	<a href="#">BusInterface</a>	The bus interface (can be 1...n)			

### 1.2.5.6 complexType BusInterface\_t

Description	Bus interface of the device, only used for devices that have one or more tunnelling server. For more information, please see KSG 616.				
Children	Name	Description			
	<b><u>Connectors</u></b>	If the tunnelling server is used for a visualisation, the addresses that shall be visualized can be added here, so that the filter tables are calculated correctly. .			
Attributes	Name	Type	Use	Default	Description
	RefId	knx:IDREF	required		The RefId to the BusInterface in the ApplicationProgram.
	Name	xs:string	optional		The name of the additional address used as a bus interface.
	Description	xs:string	optional		The description for the additional address used as a bus interface.
	Comment	xs:string	optional		The comment for the additional address used as a bus interface.
	Password	knx:String20_t	optional		The optional password for the tunnelling server.
	PasswordHash	xs:base64Binary	optional		A hash of the optional password for the tunnelling server.
	IsSecureEnabled	xs:boolean	optional	false	Flag if if the KNX stack intended to run on a given bus interface instance supports secure group communication or not.

### 1.2.5.7 element BusInterface\_t/Connectors

Description	Group addresses assigned to the bus interface. Needed for correct calculation of filter tables.				
Children	Name	Description			
	<b><u>Connector</u></b>	Connector to a group address that shall be represented in the calculated filter table.			

### 1.2.5.8 element BusInterface\_t/Connectors/Connector

Description	Group addresses assigned to a ComObjectInstanceRef for sending (and receiving)				
Attributes	Name	Type	Use	Default	Description
	GroupAddressRefId	knx:IDREF	required		Reference to a <a href="#">GroupAddress</a>

### 1.2.5.9 element DeviceInstance\_t/ParameterInstanceRefs

Description	List of parameter instances with non-default values. If a parameter has its default value, it needs not be listed here.				
Children	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b><u>ParameterInstanceRef</u></b></td> <td></td> </tr> </tbody> </table>	Name	Description	<b><u>ParameterInstanceRef</u></b>	
Name	Description				
<b><u>ParameterInstanceRef</u></b>					

### 1.2.5.10 element DeviceInstance\_t/ParameterInstanceRefs/ParameterInstanceRef

Description	Parameter instance				
Attributes	Name	Type	Use	Default	Description
	Id	xs:ID	optional		Might be set and used by Plugins. It is recommended to use one of the following methods for constructing the attribute value: <ul style="list-style-type: none"> <li>• a GUID (without enclosing braces)</li> <li>• <i>deviceid_paramrefid</i> where <i>deviceid</i> is the Id of the parent Device and <i>paramrefid</i> is the Id of the referenced ParameterRef</li> </ul>
	RefId	knx:IDREF	required		Reference to a <a href="#">ParameterRef</a> .
	Value	<a href="#">knx:Value_t</a>	optional		The current value
	GrantUseByCustomer	xs:boolean	optional	false	For ETS Inside: The installer can grant the customer the right to change the value of this parameter.
	CustomizedText	<a href="#">xs:string</a>	optional		For ETS Inside: The installer can specify a customized text for this parameter.

### 1.2.5.11 element DeviceInstance\_t/ComObjectInstanceRefs

Description	List of group communication object instances. If a communication object instance has all default settings and no associations, it needs not be listed here.				
Children	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b><u>ComObjectInstanceRef</u></b></td> <td></td> </tr> </tbody> </table>	Name	Description	<b><u>ComObjectInstanceRef</u></b>	
Name	Description				
<b><u>ComObjectInstanceRef</u></b>					

### 1.2.5.12 element DeviceInstance\_t/ComObjectInstanceRefs/ComObjectInstanceRef

Description	Group communication object instance
-------------	-------------------------------------

Type	<b><u>knx:ComObjectInstanceRef t</u></b>
------	--

### 1.2.5.13 complexType ComObjectInstanceRef\_t

Description	Group communication object instance				
Attributes	Name	Type	Use	Default	Description
	Id	xs:ID	optional		The identifier
	RefId	knx:RELIDREF	required		Reference to a <a href="#">ComObjectRef</a> RELIDREF means, the Id is stripped of the parent part, e.g. "O-2_R-9"
	Text	knx:String255_t	optional		Visible communication object name. If missing, the attribute of the underlying ComObjectRef or ComObject is used
	FunctionText	knx:String255_t	optional		Visible communication object function name. If missing, the attribute of the underlying ComObjectRef or ComObject is used
	Priority	<a href="#">knx:ComObjectPriority t</a>	optional		Transmission priority. If missing, the attribute of the underlying ComObjectRef or ComObject is used.
	ReadFlag	<a href="#">knx:Enable t</a>	optional		Read flag. If missing, the attribute of the underlying ComObjectRef or ComObject is used.
	WriteFlag	<a href="#">knx:Enable t</a>	optional		Write flag. If missing, the attribute of the underlying ComObjectRef or ComObject is used.
	CommunicationFlag	<a href="#">knx:Enable t</a>	optional		Communication flag. If missing, the attribute of the underlying ComObjectRef or ComObject is used.
	TransmitFlag	<a href="#">knx:Enable t</a>	optional		Transmit flag. If missing, the attribute of the underlying ComObjectRef or ComObject is used.
	UpdateFlag	<a href="#">knx:Enable t</a>	optional		Update flag. If missing, the attribute of the underlying ComObjectRef or ComObject is used.
	ReadOnInitFlag	<a href="#">knx:Enable t</a>	optional		ReadOnInit flag. If missing, the attribute of the underlying ComObjectRef or ComObject is used.
	DatapointType	knx:IDREFS	optional		May be a reference to (one or more) <a href="#">DatapointType</a> or <a href="#">DatapointSubtype</a> . If missing, the attribute of the underlying ComObjectRef or ComObject is used.
	Description	xs:string	optional		Description
	ChannelId	knx:IDREF	optional		The reference to the ApplicationProgramChannel in which the ComObjectInstance is located. If the ComObjectInstance is located in the ChannelIndependentBlock, the ChannelId is null.
	Links	knx:RELIDREFS	optional		The list of (shortened) group address ids that are linked with this object. The first group address in the list is always the sending one.
Acknowledges	knx:RELIDREFS	optional		The list of (shortened) group address ids that have the acknowledge flag set (used in PL).	

### 1.2.5.14 element DeviceInstance\_t/ChannelInstances

Description	<p>List of channel instances, can be 0...n.</p> <p>ChannelInstances are only available, if PreEts4Style of the referenced ApplicationProgram is false and the ApplicationProgram does not only contain the ChannelIndependentBlock.</p>
-------------	---



Children	Name	Description
	<b><u>ChannelInstance</u></b>	List of channel instances.

### 1.2.5.15 element DeviceInstance\_t/ChannelInstances/ChannelInstance

Description	The channel instances are used to visualize the logical structure of the ComObjectInstances of the device.				
Attributes	Name	Type	Use	Default	Description
	Id	xs:ID	required		The unique identifier for the ChannelInstance. Is a combination of Device ID and unique Channel ID.
	RefId	knx:RELIDREF	optional		Reference to a <a href="#">Channel in the dynamic part of the ApplicationProgram</a> . If the channel is user defined, the RefId is null. RELIDREF means, the Id is stripped of the parent part, e.g. "CH-1"
	Name	<a href="#">knx:String255_t</a>	optional		The name of the channel.Can only be edited, if RefId == null (i.e. only names of user defined ChannelInstances can be edited)
	Description	<a href="#">knx:String255_t</a>	optional		The description of the channel.
	IsActive	<a href="#">xs:boolean</a>	optional		The indicator whether the channel is currently active

### 1.2.5.16 element DeviceInstance\_t/ModuleInstances

Description	List of module instances, can be 0...n.	
Children	Name	Description
	<b><u>ModuleInstance</u></b>	List of module instances.

### 1.2.5.17 element DeviceInstance\_t/ModuleInstances/ModuleInstance

Description	The module instances are used to persist the structure of active modules.
Type	<a href="#">knx:ModuleInstance_t</a>

### 1.2.5.18 complexType ModuleInstance\_t

Description	Description of a module instance
-------------	----------------------------------

Children	Name	Description			
	<b><u>Arguments</u></b>	The list of argument with which the module instance was instantiated.			
Attributes	Name	Type	Use	Default	Description
	Id	knx:RELID	required		The shortened id of the module instance. For Modules: <b>MD-ModuleDefUniqueNumber_M-ModuleUnqjueNumber_MI-ModuleInstance@RepeatIndex</b> For SubModules: <b>MD-ModuleDefUniqueNumber_M-ModuleUnqjueNumber_MI-ModuleInstance@RepeatIndex_SM-SubModuleDefUniqueNumber_M-SubModuleUniqueNumber_MI-SubModuleInstance@RepeatIndex</b> Examples for the ID are shown <a href="#">here</a>
	RefId	knx: String255_t	required		The shortened Id of the Module
	RepeatIndex	xs:list of knx:RepeatIndex_t	optional		The repeat index of the module. The index contains a list of order infos, the order info consists of the XmlOrder and the repeat counter, separated by an 'x', (e.g. 37x2, meaning the XmlOrder is 37 and the repeat counter is 2). For nested repeats, each nesting level requires an order info.

### 1.2.5.19 element ModuleInstance\_t/Arguments

Description	The list of arguments used for the creation of the module instance			
Children	Name	Description		
	<b><u>Argument</u></b>	A specific argument used for creation of the module instance		

### 1.2.5.20 element ModuleInstance\_t/Arguments/Argument

Description	Represents a argument that was used for creation of the module instance				
Attributes	Name	Type	Use	Default	Description
	RefId	knx:RELIDREF	required		The shortened ID of the specified argument
	Value	knx:Value_t	required		The value that was used for instantiation of the module

### 1.2.5.21 element DeviceInstance\_t/GroupObjectTree

Description					
Children	Name	Description			
	<b>Node</b>	List of nodes in the group object tree (Channels and Folder).			
Attributes	Name	Type	Use	Default	Description
	GroupObjectInstances	knx:RELIDREFS	optional		The list of group object instances that are active in the ChannelIndependentBlock

### 1.2.5.22 element DeviceInstance\_t/GroupObjectTree/Node

Description	The list of nodes that are in the root level of the group object tree.
Type	<a href="#">knx:Node_t</a>

### 1.2.5.23 element Node\_t

Description	The node element in the GroupObjectTree				
Children	Name	Description			
	<b>Nodes</b>				
Attributes	Name	Type	Use	Default	Description
	Type	xs:string	required		The type of the node. Can be: <ul style="list-style-type: none"> <li>- Folder (ParameterBlock with "ShowInComObjectTree")</li> <li>- Channel</li> </ul>
	RefId	knx:RELIDREF	required		The shortened RefId to the Channel or ParameterBlock
	GroupObjectInstances	knx:RELIDREFS	optional		The list of shortened RefIds

### 1.2.5.24 element DeviceInstance\_t/RfFastAckSlots

Description	.
-------------	---

Children	Name	Description
	<b>Slot</b>	List of fast ACK RF slots.

### 1.2.5.25 element DeviceInstance\_t/ RfFastAckSlots /Slot

Description					
Attributes	Name	Type	Use	Default	Description
	GroupAddressRefId	knx:IDREF	required		
	Number	<a href="#">xs:unsignedByte</a>	required		

### 1.2.5.26 element DeviceInstance\_t/AdditionalAddresses

Description	Contains additional device addresses used by the device (maximum 254)	
Children	Name	Description
	<b>Address</b>	Device address

### 1.2.5.27 element DeviceInstance\_t/AdditionalAddresses/Address

Description	Additional device address (individual address) used by the device				
Attributes	Name	Type	Use	Default	Description
	Address	xs:unsignedByte	required		The additional device address (individual address) used by the device.
	Name	knx:String255_t	optional		The name of the additional address.
	Description	xs:string	optional		The description of the additional address.
	Comment	xs:string	optional		A comment for the additional address.

### 1.2.5.28 element DeviceInstance\_t/BinaryData

Description	For use by plugins	
Children	Name	Description
	<b><u>BinaryData</u></b>	

### 1.2.5.29 element DeviceInstance\_t/BinaryData/BinaryData

Description	For use by plugins				
Children	Name	Description			
	<b><u>Data</u></b>	Any data (optional)			
Attributes	Name	Type	Use	Default	Description
	Id	xs:string	optional		Might be set and used by Plugins. It is recommended to use one of the following methods for constructing the attribute value: <ul style="list-style-type: none"> <li>• a GUID (without enclosing braces)</li> <li>• <i>deviceid_id</i> where <i>deviceid</i> is the Id of the parent Device and <i>id</i> is the Id of the referenced BinaryData or the suitably escaped name .</li> </ul>
	RefId	knx:IDREF	optional		Reference to a <a href="#">BinaryData</a> .
	Name	knx:String255_t	optional		
	DoNotCopy	xs:boolean	optional	false	Allows DCAs to specify if this binary data contains instance-specific data and hence if this binary data shall not be copied to the new device instance on device copy operations.

### 1.2.5.30 element DeviceInstance\_t/BinaryData/BinaryData/Data

Description	
Type	<b>xs:base64Binary</b>

## 1.2.6 Building Structure

### 1.2.6.1 element Project\_t/Installations/Installation/Locations

Description	Contains the building structure				
Type	<b><u>knx:Locations_t</u></b>				
Children	<table><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td><b><u>BuildingPart</u></b></td><td></td></tr></tbody></table>	Name	Description	<b><u>BuildingPart</u></b>	
Name	Description				
<b><u>BuildingPart</u></b>					

### 1.2.6.2 complexType Locations\_t

Description	Contains the building structure (locations structure)				
Children	<table><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td><b><u>Space</u></b></td><td>Any number of spaces</td></tr></tbody></table>	Name	Description	<b><u>Space</u></b>	Any number of spaces
Name	Description				
<b><u>Space</u></b>	Any number of spaces				

### 1.2.6.3 element Locations\_t/Space

Description	A space. Space elements directly below Locations_t will normally have Type "Area" or "Building" or "Ground"
Type	<b><u>knx:Space_t</u></b>

### 1.2.6.4 complexType Space\_t

Description	An element of the building structure								
Children	<table><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td><b><u>Space</u></b></td><td>Child space</td></tr><tr><td><b><u>DeviceInstanceRef</u></b></td><td>List of devices in this building part.</td></tr><tr><td><b><u>Function</u></b></td><td>List of functions in this building part.</td></tr></tbody></table>	Name	Description	<b><u>Space</u></b>	Child space	<b><u>DeviceInstanceRef</u></b>	List of devices in this building part.	<b><u>Function</u></b>	List of functions in this building part.
Name	Description								
<b><u>Space</u></b>	Child space								
<b><u>DeviceInstanceRef</u></b>	List of devices in this building part.								
<b><u>Function</u></b>	List of functions in this building part.								

Attributes	Name	Type	Use	Default	Description
	Id	xs:ID	required		Unique ID. On export or conversion, this will be constructed as <i>parid_BP-number</i> , where: <i>parid</i> ID of the parent Project and InstallationID sepearted with '-' <i>number</i> Unique number of the building part within the project.
	Name	knx:String255_t	required		Name
	Type	<a href="#">knx:Space_t</a>	required		One of: "Building", "BuildingPart", "Floor", "Room", "RoomPart", "DistributionBoard", "Stairway", "Corridor", "Area", "Ground" and "Segment".
	Usage	knx:IDREF	optional		The optional usage for this space.
	Number	knx:String255_t	optional		Optional number
	Comment	xs:string	optional		Cptional comment
	CompletionStatus	<a href="#">knx:CompletionStatus_t</a>	optional	Undefined	Completion status
	DefaultLine	xs:string	optional		The RefId of the line, to which devices will be added if added to the BuildingPart
	Description	xs:string	optional		Description
	Puid	xs:string	required		The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.

### 1.2.6.5 element Space\_t/Space

Description	Child building part.
Type	<a href="#">knx:BuildingPart_t</a>

### 1.2.6.6 element BuildingPart\_t/DeviceInstanceRef

Description	References a device contained in a building part.
Type	<a href="#">knx:DeviceInstanceRef_t</a>

### 1.2.6.7 element BuildingPart\_t/Function

Description	References a function contained in a building part.
-------------	---

Type	<a href="#">knx:Function_t</a>
------	--------------------------------

### 1.2.6.8 complexType DeviceInstanceRef\_t

Description											
Attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>RefId</td> <td>knx:IDREF</td> <td>required</td> <td></td> <td>Reference to <a href="#">DeviceInstance</a></td> </tr> </tbody> </table>	Name	Type	Use	Default	Description	RefId	knx:IDREF	required		Reference to <a href="#">DeviceInstance</a>
Name	Type	Use	Default	Description							
RefId	knx:IDREF	required		Reference to <a href="#">DeviceInstance</a>							

### 1.2.6.9 complexType Function\_t

Description	A function containing group addresses																																																							
Children	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><a href="#">GroupAddressRef</a></td> <td>List of functions in this building part.</td> </tr> </tbody> </table>					Name	Description	<a href="#">GroupAddressRef</a>	List of functions in this building part.																																															
Name	Description																																																							
<a href="#">GroupAddressRef</a>	List of functions in this building part.																																																							
Attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Id</td> <td>xs:ID</td> <td>required</td> <td></td> <td></td> </tr> <tr> <td>Name</td> <td>knx:String255_t</td> <td>required</td> <td></td> <td>Name</td> </tr> <tr> <td>Type</td> <td><a href="#">knx:String255_t</a></td> <td>optional</td> <td></td> <td>The optional type of the function</td> </tr> <tr> <td>Implements</td> <td>knx:IDREFS</td> <td>optional</td> <td></td> <td>RefIds to the function types this function implements.</td> </tr> <tr> <td>Number</td> <td>knx:String255_t</td> <td>optional</td> <td></td> <td>Optional number</td> </tr> <tr> <td>Comment</td> <td>xs:string</td> <td>optional</td> <td></td> <td>Optional comment</td> </tr> <tr> <td>Description</td> <td>xs:string</td> <td>optional</td> <td></td> <td>Description</td> </tr> <tr> <td>CompletionStatus</td> <td><a href="#">knx:CompletionStatus_t</a></td> <td>optional</td> <td>Undefined</td> <td>Completion status</td> </tr> <tr> <td>DefaultGroupRange</td> <td>xs:IDREF</td> <td>optional</td> <td></td> <td>The RefId of the default GroupRange</td> </tr> <tr> <td>Puid</td> <td>xs:string</td> <td>required</td> <td></td> <td>The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.</td> </tr> </tbody> </table>	Name	Type	Use	Default	Description	Id	xs:ID	required			Name	knx:String255_t	required		Name	Type	<a href="#">knx:String255_t</a>	optional		The optional type of the function	Implements	knx:IDREFS	optional		RefIds to the function types this function implements.	Number	knx:String255_t	optional		Optional number	Comment	xs:string	optional		Optional comment	Description	xs:string	optional		Description	CompletionStatus	<a href="#">knx:CompletionStatus_t</a>	optional	Undefined	Completion status	DefaultGroupRange	xs:IDREF	optional		The RefId of the default GroupRange	Puid	xs:string	required		The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.
Name	Type	Use	Default	Description																																																				
Id	xs:ID	required																																																						
Name	knx:String255_t	required		Name																																																				
Type	<a href="#">knx:String255_t</a>	optional		The optional type of the function																																																				
Implements	knx:IDREFS	optional		RefIds to the function types this function implements.																																																				
Number	knx:String255_t	optional		Optional number																																																				
Comment	xs:string	optional		Optional comment																																																				
Description	xs:string	optional		Description																																																				
CompletionStatus	<a href="#">knx:CompletionStatus_t</a>	optional	Undefined	Completion status																																																				
DefaultGroupRange	xs:IDREF	optional		The RefId of the default GroupRange																																																				
Puid	xs:string	required		The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.																																																				

### 1.2.6.10 complexType GroupAddressRef\_t

Description	A type containing information of the referenced group address
-------------	---



Attributes	Name	Type	Use	Default	Description
	Id	xs:ID	required		Unique identifier of the GroupAddressRef
	RefId	knx:IDREF	required		Reference to <a href="#">GroupAddress</a>
	Name	knx:String255_t	required		Name
	Role	knx:String255_t	optional		The optional name of the role of that group address
	Puid	xs:string	required		The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.

### 1.2.6.11 complexType Trades\_t

Description	Contains the trades structure				
Children	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>Trade</b></td> <td>Any number of trades</td> </tr> </tbody> </table>	Name	Description	<b>Trade</b>	Any number of trades
Name	Description				
<b>Trade</b>	Any number of trades				

### 1.2.6.12 element Trades\_t/Trade

Description	A Trade.
Type	<b><u>knx:Trade t</u></b>

### 1.2.6.13 complexType Trade\_t

Description	An element of the trades structure										
Children	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b>Trade</b></td> <td>Child Trades</td> </tr> <tr> <td><b>DeviceInstanceRef</b></td> <td>List of devices in this trade.</td> </tr> </tbody> </table>					Name	Description	<b>Trade</b>	Child Trades	<b>DeviceInstanceRef</b>	List of devices in this trade.
Name	Description										
<b>Trade</b>	Child Trades										
<b>DeviceInstanceRef</b>	List of devices in this trade.										
Attributes	Name	Type	Use	Default	Description						
	Id	xs:ID	optional		Unique ID.  On export or conversion, this will be constructed as <i>parid_T-number</i> , where:						

			<i>parid</i>	ID of the parent Project and InstallationID sepearted with '-'
			<i>number</i>	Unique number of the Trade within the project.
Name	knx:String255_t	required		Name of the trade
Number	knx:String255_t	optional		Optional number
Comment	xs:string	optional		Cptional comment
CompletionStatus	<a href="#">knx:CompletionStatus_t</a>	optional	Undefined	Completion status
Description	xs:string	optional		Description of the trade
Puid	xs:string	required		The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.

### 1.2.6.14 element Trade\_t/Trade

Description	
Type	<b><u>knx:Trade_t</u></b>

### 1.2.6.15 element Trade\_t/DeviceInstanceRef

Description	References a device contained in a trade.
Type	<b><u>knx:DeviceInstanceRef_t</u></b>

## 1.2.7 Group Addresses

### 1.2.7.1 element Project\_t/Installations/Installation/GroupAddresses

Description	Contains the group address structure
Type	<b><u>knx:GroupAddresses_t</u></b>

### 1.2.7.2 complexType GroupAddresses\_t

Description	Contains the group address structure	
Children	Name	Description
	<b>GroupRange</b>	List of named group address ranges

### 1.2.7.3 element GroupRange\_t/GroupAddress

Description	Describes a group address				
Attributes	Name	Type	Use	Default	Description
	Id	xs:ID	required		<p>Unique ID.</p> <p>On export or conversion, this will be constructed as <i>parid_GA-number</i>, where:</p> <p><i>parid</i> ID of the parent Project and InstallationID sepearted with '-'</p> <p><i>number</i> Unique number of the group address within the project. This does not reflect the address value! For converted projects, this corresponds to GroupAddress.UniqueNumber in the database schema.</p>
	Address	xs:unsignedInt	required		Group address [1...65535]
	Name	knx:String255_t	required		Name
	Unfiltered	xs:boolean	optional	false	<p>If true, the group addresses in the range will not be filtered by routers.</p> <p>Note that if a group address is part of one or more GroupRanges with Unfiltered=true, it will not be filtered irrespective of the setting of Unfiltered in the GroupAddress.</p>
	Central	xs:boolean	optional	false	If true, the group address will be treated as central address during copy operations.
	Global	xs:boolean	optional	false	<p>If true, the group address will be used in all installations of the project.</p> <p>Global groups must have the same address and type in all installations of a project.</p>
	Description	xs:string	optional		Description
	Comment	xs:string	optional		Comment
	DatapointType	knx:IDREF	optional		<p>Optional datapoint type specification. A reference to <a href="#">DatapointType</a> or <a href="#">DatapointSubtype</a>.</p> <p>If the group address is linked to any DeviceCommunicationObjects, the sizes must match.</p>
	Puid	xs:string	required		The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.
	Key	knx:Aes128_t	optional		The key used for data security communication. All senders and receivers of this group address use the same key.
	Security	knx:SecurityMode	optional	Auto	Defines the security mode for the group address. Can be either Auto, On or Off.

### 1.2.7.4 element GroupAddresses\_t/GroupRanges/GroupRange

Description	Top-level named group range
Type	extension of <b><u>knx:GroupRange_t</u></b>

### 1.2.7.5 complexType GroupRange\_t

Description	Element of the group address structure										
Children	<table border="0"> <thead> <tr> <th>Name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><b><u>GroupRange</u></b></td> <td>Child group ranges</td> </tr> <tr> <td><b><u>GroupAddress</u></b></td> <td>GroupAddresses located within the GroupRange</td> </tr> </tbody> </table>					Name	Description	<b><u>GroupRange</u></b>	Child group ranges	<b><u>GroupAddress</u></b>	GroupAddresses located within the GroupRange
Name	Description										
<b><u>GroupRange</u></b>	Child group ranges										
<b><u>GroupAddress</u></b>	GroupAddresses located within the GroupRange										
Attributes	Name	Type	Use	Default	Description						
	Id	xs:ID	required		Unique ID. On export or conversion, this will be constructed as <i>parid_GR-number</i> , where: <i>parid</i> ID of the parent Project and InstallationID sepearted with '-' <i>number</i> Unique number of the group range within the project.						
	Name	knx:String255_t	required		Name						
	RangeStart	xs:unsignedShort	required		First possible group address in the range						
	RangeEnd	xs:unsignedShort	required		Last possible group address in the range						
	Unfiltered	xs:boolean	optional	false	If true, all group addresses in the range will not be filtered by routers; irrespective of the individual setting of GroupAddress/@Unfiltered.						
	Description	xs:string	optional		Description						
	Comment	xs:string	optional		Comment						
	Puid	xs:string	required		The project wide unique identifier. After deletion of the element, no other element will receive the same Puid.						
	Security	knx:SecurityMode	optional	Auto	Defines the security mode for the group addresses within the range or any child range.						

### 1.2.7.6 element GroupRange\_t/GroupRange

Description	Child named group address range
Type	extension of <b><u>knx:GroupRange_t</u></b>

## 1.2.8 SplitInfos

### 1.2.8.1 element Project\_t/Installations/Installation/SplitInfos

Description	The required information about a split installation..
Type	<b><u>knx:SplitInfos_t</u></b>

### 1.2.8.2 complexType SplitInfos\_t

Description	Collection of SplitInfo elements, used for Split & Merge				
Type	extension of <b><u>knx:SplitInfo_t</u></b>				
Children	<table border="1"><thead><tr><th>Name</th><th>Description</th></tr></thead><tbody><tr><td><b><u>SplitInfo</u></b></td><td>Any number of split infos</td></tr></tbody></table>	Name	Description	<b><u>SplitInfo</u></b>	Any number of split infos
Name	Description				
<b><u>SplitInfo</u></b>	Any number of split infos				

### 1.2.8.3 element SplitInfo\_t/SplitInfo

Description	The required information about a split installation..
Type	<b><u>knx:SplitInfo_t</u></b>

### 1.2.8.4 complexType SplitInfo\_t

Description	An element with information for Split & Merge
-------------	---

Attributes	Name	Type	Use	Default	Description
	ObjectPath	xs:string	required		
	Cookie	xs:string	required		Pattern for the cookie: [0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}

## 4 Transfer files

For export and import scenarios, the generated XML file(s) will be packed into a ZIP archive. This has the following advantages:

- By compression, the files have a manageable size
- Not everything needs to be in a single XML. This is important since current XML parsers and XPath implementations do not work well or do not work at all on huge XML files.

knx:IDREF need not resolve within each individual XML file within the archive, but within the whole archive.

For import, the individual XML files may also be present unzipped, but in the same file system directory.

### 4.1 File extensions

As file extension, the following is used:

*.knxprod	If just master and manufacturer product data is included
*.knxproj	If master, product and project data is included.

## 4.2 Content

### 4.2.1 Non-XML files

The following data is not stored within the XML files but as external files

- Baggage data
- BinaryData and BinaryDataRef data within device instance data
- UserFile data

The corresponding XML elements omit the Data child element.

### 4.2.2 Distribution to partial XML files

When distributing the data to different XML files, the following restrictions apply:

- All MasterData is in one XML file.

- Together with an ApplicationProgram element, all child elements must be in the same XML file.
- Together with a Project element, all child elements must be in the same XML file.

Logically, the files can be thought of as a merged XML file.

In principle, starting from the KNX element, the files are merged recursively, with the following rules:

- The following elements will be identified (within a recursion level); they must have identical attributes in each partial XML.
  - Elements with same tag and same “Id”
  - Elements with same tag without “Id” (this is for the container-type elements like e.g. “ManufacturerData”).
  - Language elements with same “Identifier”
  - Language/Translation elements with same “RefId”
  - Language/Translation/Translation elements with same “AttributeName”
  - Exception: Project is never merged (two projects even with the same name will stay two distinct projects)
  - Below ApplicationProgram no merging is required; here everything must be identical.

The converter will produce the partial XML files according to the following rules:

- Each ApplicationProgram element will be written to a separate XML file
- Each Baggage element will be written to a separate XML file
- Each Project element will be written to a separate XML file

### 4.2.3 Naming convention

To avoid name conflicts between the individual XML files within the archive, the following naming convention should be obeyed:


knx_master.xml	Created by KNX; contains only master data.
M- <i>iiii</i> /Baggages.xml	Created by manufacturer <i>iiii</i> (manufacturer ID, formatted as 4 hex digits); contains baggage data.
M- <i>iiii</i> /Catalog.xml	Created by manufacturer <i>iiii</i> (manufacturer ID, formatted as 4 hex digits); contains catalog data.
M- <i>iiii</i> /Hardware.xml	Created by manufacturer <i>iiii</i> (manufacturer ID, formatted as 4 hex digits); contains hardware data.
M- <i>iiii</i> /M- <i>iiii</i> _A- <i>nnnn</i> - <i>vv</i> - <i>ffff</i> .xml	Created by manufacturer <i>iiii</i> (manufacturer ID, formatted as 4 hex digits); contains the data for the application program <i>nnnn</i> in version <i>vv</i> with fingerprint <i>ffff</i> .
P- <i>iiii</i> /project.xml	Created by user; contains the global data for project <i>iiii</i> (internal project ID, formatted as 4 hex digits).
P- <i>iiii</i> / <i>n</i> .xml	Created by user; contains the data for installation <i>n</i> of project <i>iiii</i> (internal project ID, formatted as 4 hex digits).
*.xml	Created by user; contains project data (* should not contain – and _ characters).

#### 4.2.4 Password protection

When exporting a password-protected project, the P-XXXX folder inside the \*.knxproj zip-file container will itself be zip compressed and AES-256 encrypted. To minimize the chance of dictionary attacks on potentially weak project passwords, the zip-encryption password will not be the project password directly, but derived using a PBKDF2 calculation:

```
ZipPassword := Base64( PBKDF2( HMAC-SHA256, ProjectPassword, "21.project.ets.knx.org", 65536, 32))
```

For the calculation of the hash data the project password needs to be encoded in little-endian UTF-16 without BOM and the salt needs to be encoded in ASCII (little-endian UTF-8 without BOM).

E.g., the project password `a` would result in the zip password string `+FAwP4iI7/Pu4WB3HdIHbbFmteLahPAVkjJShKeozAA=`, test in `2+IIP7ErCPPKxFjJXc59GFx2+w/1VTLHjJ2duc04CYQ=` and `Penn¥w1se`  in `ZjlYlh+eTtoHvFadU7+EKvF4jOdEm7WkP49uanOMMk0=`.

**Note that there is no way to recover or reset a lost ZIP password!**