

L-VIS™

LOYTEC Visualization

User Manual

LOYTEC electronics GmbH



Contact

LOYTEC
Blumengasse 35
A-1170 Vienna
AUSTRIA/EUROPE
support@loytec.com
<http://www.loytec.com>

Version 7.2

Document No. 88068524

LOYTEC MAKES AND YOU RECEIVE NO WARRANTIES OR CONDITIONS, EXPRESS, IMPLIED, STATUTORY OR IN ANY COMMUNICATION WITH YOU, AND LOYTEC SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THIS PRODUCT IS NOT DESIGNED OR INTENDED FOR USE IN EQUIPMENT INTENDED FOR SURGICAL IMPLANT INTO THE BODY OR OTHER APPLICATIONS INTENDED TO SUPPORT OR SUSTAIN LIFE, FOR USE IN FLIGHT CONTROL OR ENGINE CONTROL EQUIPMENT WITHIN AN AIRCRAFT, OR FOR ANY OTHER APPLICATION IN WHICH IN THE FAILURE OF SUCH PRODUCT COULD CREATE A SITUATION IN WHICH PERSONAL INJURY OR DEATH MAY OCCUR.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of LOYTEC.

AST™, LC3020™, L-Chip™, L-Core™, L-DALI™, L-GATE™, L-INX™, L-IOB™, LIOB Connect™, L-IP™, L-MBUS™, L-OPC™, LPA™, L-POW™, L-ROC™, L-STUDIO™, L-Switch™, L-Term™, L-VIS™, L-WEB™, ORION™ stack, buildings under control™ are trademarks of LOYTEC electronics GmbH.

LonTalk®, LONWORKS®, Neuron®, LONMARK®, LonMaker®, *i.LON*®, and LNS® are trademarks of Echelon Corporation registered in the United States and other countries.

Table of Contents

1	Introduction.....	15
1.1	What is L-VIS?	15
1.2	Available Models	17
2	What's New in L-Vis.....	19
2.1	New in L-Vis 7.2.0.....	19
2.1.1	New Control: Pie Chart	19
2.1.2	Stacked and Side-By-Side Bar Charts.....	19
2.1.3	Auto Line Break	19
2.1.4	Color Gradient Mapping	19
2.1.5	New Option for Pop-Up Action	19
2.1.6	Font Embedding Information	19
2.2	New in L-Vis 7.0.0.....	20
2.2.1	Multilingual Projects	20
2.2.2	Localized Controls	20
2.2.3	Improved Trend Control	20
2.2.4	Improved Bitmap Control	20
2.2.5	Improved Alarm List.....	20
2.2.6	Improved Font System	21
2.2.7	Scripting Capability	21
2.3	New in L-Vis 6.2.0.....	21
2.3.1	Localized date and number controls.....	21
2.3.2	Extended date control.....	21
2.3.3	Improved trend control.....	21
2.3.4	Knob and Webcam controls	21
2.3.5	Bidirectional data point references	22
2.3.6	Improved BACnet support	22
2.3.7	New system registers.....	22
2.4	New in L-Vis 6.1.0.....	22
2.4.1	Support for complex writing systems.....	22
2.4.2	New Control: Webcam.....	22
2.4.3	New Control: Rotating Knob	22
2.4.4	New Action: Show Pop-Up.....	23
2.4.5	Number Control: Show Units.....	23
2.4.6	Sound Action: Playback volume adjustment.....	23
2.4.7	LWEB projects: Option to use client-side fonts.....	23
2.4.8	Improved handling of templates.....	23

2.5	New in L-Vis 5.1.0.....	23
2.5.1	Support for new models	23
2.5.2	Math objects as input data points	23
2.5.3	New pre-installed fonts	24
2.5.4	More operations supporting undo	24
2.5.5	L-WEB projects use true-type fonts for the preview.....	24
2.5.6	Improved performance and memory usage	24
2.5.7	Dual-Ethernet with Separate Networks	24
2.5.8	WLAN Interface	24
2.5.9	OPC UA Server.....	24
2.5.10	Alarm Server Ack-All Data Point	24
2.5.11	Format Strings in E-Mails	24
2.5.12	Output NVs with Integrated Feedback	24
2.6	New in L-Vis 5.0.0.....	25
2.6.1	Usability Improvements	25
2.6.2	Modbus Support.....	25
2.6.3	Improved Web-UI	25
2.6.4	Improved OPC Client.....	25
2.6.5	Improved OPC Server Management	25
2.6.6	Nested Template Paths	25
2.6.7	Position Lock	25
2.6.8	Absolute Paths for Show Page	26
2.6.9	Improved Boot Update Handling	26
2.6.10	Improved Usage Report	26
2.6.11	Improved Project Builder	26
3	Getting Started	27
3.1	Installing the L-VIS Configuration Application	27
3.2	Connecting to the Device.....	27
3.2.1	TCP / IP (Ethernet) Connections	27
3.2.2	CEA-709 Devices only	28
3.3	Quick Start Tutorial CEA-709	31
3.3.1	Add the device to the network.....	31
3.3.2	Create Dynamic NVs (LNS only)	32
3.3.3	Start the L-VIS Configuration Software	32
3.3.4	Create Static NVs (if no dynamic NVs available).....	33
3.3.5	Create Menu Structure and Pages	33
3.3.6	Add Controls to the Page	34
3.3.7	Connect the Data-Points.....	35
3.3.8	Write the Project to the L-VIS device	36
3.3.9	Test the Project.....	36

3.4	Quick Start Tutorial BACnet	37
3.4.1	Add the device to the network.....	37
3.4.2	Start the L-VIS Configuration Software	37
3.4.3	Create Data Points.....	38
3.4.4	Create Menu Structure and Pages	38
3.4.5	Add Controls to the Page	39
3.4.6	Connect the Data-Points.....	39
3.4.7	Write the Project to the L-VIS device	40
3.4.8	Test the Project.....	41
4	Mechanical Installation	42
4.1	Dimensions and Mounting	42
4.2	Theft Protection	42
5	Electrical Installation.....	43
5.1	Connection diagram	43
5.2	Electrical Characteristics LVIS-3E100/ME200	43
5.3	Electrical Characteristics LVIS-3E11x/ME21x	43
5.4	Electrical Characteristics LVIS-3MEx.....	44
5.5	Touch Panel Cleaning Instructions.....	46
5.5.1	Appropriate cleaning measures	46
5.5.2	Inappropriate cleaning measures	46
5.6	Terminals and Jumpers	46
5.7	LEDs and Buttons	47
5.8	Control Network Interfaces	49
6	Interface Configuration.....	50
6.1	Selecting the Interface.....	50
6.2	Configuring the IP-852 interface.....	52
7	Operating L-VIS	53
7.1	Touch Screen.....	53
7.1.1	Operation.....	53
7.1.2	Calibration.....	54
7.2	Data Input	54
7.2.1	Knob Control.....	56
7.2.2	Simple Schedule Control.....	57
7.2.3	Extended Schedule Control.....	60
7.3	Setup Menu	65
7.3.1	Common Pages	65
7.3.2	CEA-709 Models	66
7.3.3	BACnet Models.....	67
7.3.4	Custom Setup Menu	67

8	Configuration Software	68
8.1	Main Window	68
8.2	Object Tree	68
8.3	LCD Preview	69
8.3.1	Selection Candidate	69
8.3.2	Object Selection	70
8.3.3	Selection Indicator	71
8.3.4	Resize Frame	72
8.3.5	Guides	72
8.3.6	Snap to Edges	72
8.3.7	Snap to Grid	73
8.3.8	Pan and Zoom	73
8.3.9	Edge Scrolling	74
8.3.10	Move and Resize	74
8.3.11	Vector drawing	75
8.4	Main Menus	75
8.4.1	Undo / Redo	75
8.4.2	Find Objects	75
8.4.3	Find Next	76
8.4.4	Find by UID	77
8.4.5	Device Model and Firmware	77
8.5	Context Menus	78
8.5.1	Add Objects	78
8.5.2	Enable/Disable Access Control	78
8.5.3	Cut / Copy / Paste	78
8.5.4	XML Export / Import	80
8.5.5	Manage Data Points	80
8.5.6	Find Instances	82
8.5.7	Expand / Fold Tree	82
8.5.8	Lock / Unlock Position	82
8.5.9	Hide / Show, Hide Contents, Show Contents	83
8.5.10	Add Items from Data Points	83
8.5.11	Add Pages from Data Points	83
8.5.12	Update Instances	83
8.5.13	Update Symbol(s)	83
8.5.14	Update Subproject	83
8.5.15	Restore Defaults	83
8.6	Tool Bar	84
8.6.1	Preview Settings	84
8.6.2	Layout Tools	84

8.7	Images.....	85
8.7.1	Image Manager	85
8.7.2	Image Size and Auto-Scaling	86
8.7.3	Embedded Sources	87
8.7.4	Color Reduction	87
8.7.5	Transparency	87
8.7.6	Animations	88
8.7.7	Supported File Types	88
8.8	Fonts	88
8.8.1	Font Import	89
8.8.2	Font Manager	89
8.8.3	Font Embedding	91
8.8.4	Font File Location	92
8.8.5	Font Recoding	92
8.9	Sounds	93
8.10	Template-System	93
8.10.1	Efficient Project Development	93
8.10.2	Improve Marketing.....	94
8.10.3	Corporate Design	94
8.10.4	Solving Complex Tasks	94
8.10.5	Improved Structure.....	94
8.11	Workflow.....	95
8.12	Common Property Pages	96
8.12.1	General	96
8.12.2	Common Properties.....	97
8.12.3	Editor.....	100
8.12.4	Color	101
9	Object Description	102
9.1	Introduction	102
9.2	Menu.....	102
9.2.1	Menu Properties	103
9.3	Menu Item.....	103
9.3.1	Menu Item Properties	104
9.4	Page.....	104
9.4.1	Page Properties.....	105
9.5	Controls.....	105
9.5.1	Text Control	106
9.5.2	Bitmap Control.....	107
9.5.3	Push Button	108
9.5.4	Numeric Control	109

9.5.5	Bar Control.....	113
9.5.6	Trend Control.....	118
9.5.7	Date Control.....	126
9.5.8	Data Log Control	132
9.5.9	Schedule Control.....	133
9.5.10	Alarm-List Control.....	138
9.5.11	Vector Drawing Control.....	142
9.5.12	Analog Meter	144
9.5.13	Knob Control.....	146
9.5.14	Pie Chart Control	152
9.5.15	Webcam Control	154
9.6	Data Point.....	157
9.6.1	Data Point Properties.....	159
9.6.2	System Startup	161
9.6.3	Data Point Icons	162
9.7	Mapping Table.....	163
9.7.1	Mapping Table Properties	163
9.8	Color Mapping.....	164
9.8.1	Color Mapping Properties	164
9.9	Action Objects.....	165
9.9.1	Action Properties.....	166
9.9.2	Page Links.....	171
9.10	Collections	172
9.10.1	Collection Properties	172
9.10.2	Collection Handling	172
9.11	Alarm Generators.....	173
9.11.1	Alarm Generator Properties	174
9.11.2	Application Hints	175
9.12	Data Point Connectors	176
9.13	Mathematic Objects	177
9.13.1	Math Object Properties	177
9.13.2	Application Hints	178
9.13.3	Function List	179
9.14	Trigger Objects.....	181
9.14.1	Trigger Object Properties	181
9.15	Templates	183
9.15.1	Template Properties	183
9.15.2	Creating a Template	183
9.15.3	Instantiating a Template	184
9.15.4	Changes to Data Point Folders	185

9.15.5 Nested Templates	186
9.15.6 Nested Base Paths	187
9.15.7 Local Changes	187
9.15.8 Nested Changes	188
9.15.9 Base Path Placeholder Tags	189
10 Data Point Configuration	191
10.1 Data Points	191
10.1.1 Overview	191
10.1.2 Timing Parameters	192
10.1.3 Default Values	192
10.1.4 Persistency	193
10.1.5 Parameters	193
10.1.6 Behavior on Value Changes	193
10.1.7 Custom Scaling	194
10.1.8 User Registers	194
10.1.9 Structures	195
10.1.10 Property Relations	195
10.1.11 Engineering Units	197
10.2 Connections	197
10.2.1 Local Connections	197
10.2.2 Multi-Slot Connections	198
10.2.3 Global Connections	199
10.3 AST Features	200
10.3.1 Alarming	200
10.3.2 Scheduling	201
10.3.3 Trending	203
10.3.4 E-mail	204
10.3.5 Historic Filters	204
10.4 CEA-709 Technology	205
10.4.1 CEA-709 Data Points	205
10.4.2 CEA-709 Schedulers	206
10.4.3 CEA-709 Alarming	206
10.4.4 CEA-709 Trend Logs	206
10.4.5 Dynamic Polling in CEA-709	207
10.4.6 CEA-709 Data Points in Connections	207
10.4.7 Static Interface Changes	207
10.5 BACnet Technology	207
10.5.1 BACnet Data Points	207
10.5.2 BACnet Schedulers	208
10.5.3 BACnet Alarming	208

10.5.4	BACnet Trend Logs	209
10.5.5	Dynamic Polling in BACnet	209
10.5.6	BACnet Data Points in Connections	210
10.6	Data Point Manager	210
10.6.1	Folder List	210
10.6.2	Data Point List	212
10.6.3	Property View	213
10.6.4	CEA-709 Properties	215
10.6.5	BACnet Properties	216
10.6.6	Modbus Properties	217
10.6.7	Data Point Usage	218
10.6.8	Multistate Maps	219
10.6.9	Property Relations	219
10.6.10	Favorites	220
10.6.11	Buttons	221
10.7	OPC Device Manager	221
10.7.1	Tool Bar	222
10.7.2	Device Properties	222
10.8	Manage Connections	223
10.8.1	Create a New Connection	223
10.8.2	Modify Connections	225
10.8.3	Create a Multi-Slot Connection	227
10.8.4	Create a Math Block Adapter	228
10.8.5	Create a Global Connection	230
10.9	E-mail Templates	231
10.9.1	Common E-Mail Properties	232
10.9.2	E-Mail Trigger	232
10.9.3	Attachments	233
10.9.4	Limit E-mail Send Rate	233
10.10	Local Schedule and Calendar	233
10.10.1	Create a local Calendar	233
10.10.2	Create Calendar Pattern	234
10.10.3	Create a Local Scheduler	234
10.10.4	Configure Scheduled Data Points	235
10.10.5	Configure Scheduled Events	236
10.10.6	Configure Exception Days	238
10.10.7	Configure Control Data Points	239
10.10.8	Using the SNVT_tod_event	240
10.10.9	Schedule Configuration Data	240
10.10.10	System Resources	241

10.10.11	Resource allocation	242
10.10.12	Using the Scheduler	243
10.11	Local Alarming	243
10.11.1	Create an Alarm Server	243
10.11.2	Create an Alarm Condition	244
10.11.3	Deliver Alarms via E-mail	246
10.11.4	Multi-Edit Alarm Conditions	247
10.12	Local Trending	248
10.12.1	Create a Local Trend.....	248
10.12.2	Configure Trended Data Points	249
10.12.3	Trend Triggers.....	250
10.12.4	Deliver Trend Data via E-mail	251
10.12.5	Technology Trends.....	252
10.13	Trend Data Points and Trend Controls.....	252
10.13.1	Recorded by a Trend Control	252
10.13.2	Recorded by a Trend Data Point	252
10.14	Remote AST Objects	253
10.14.1	Remote Scheduler and Calendar	253
10.14.2	Alarm Clients	254
10.14.3	Remote Trend Logs.....	255
10.15	Historic Filters	255
10.15.1	Create Historic Filters	255
10.15.2	Managing Historic Filter Resources.....	256
10.16	Data Points in L-Vis Projects	257
10.16.1	User Register.....	258
10.16.2	Local Network Variable (CEA-709)	259
10.16.3	Local Server-Object (BACnet).....	259
10.16.4	Remote Network Object.....	262
11	Project Settings.....	265
11.1	Basic Settings	265
11.1.1	Page Protection	265
11.1.2	Timeout Values	265
11.1.3	Sound	266
11.1.4	OEM Bitmaps	266
11.1.5	System Strings.....	266
11.2	Extended Settings	266
11.2.1	System Flags	266
11.2.2	System Settings	267
11.2.3	System Colors	268
11.2.4	Unit System.....	268

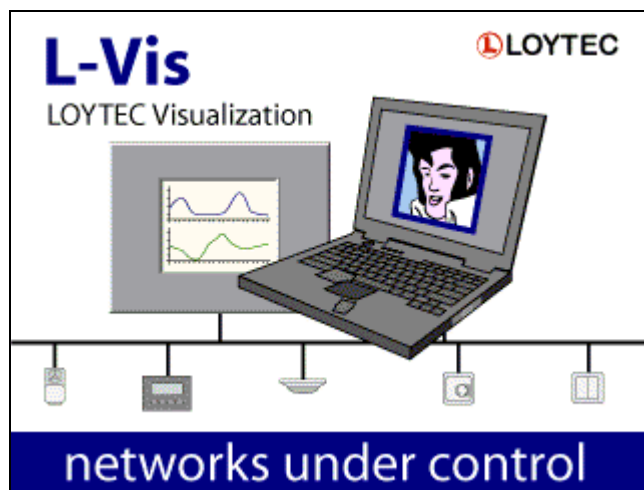
11.2.5 Time Zone and DST	268
11.3 Localization.....	268
11.3.1 Project Languages	268
11.3.2 Locale Settings	269
11.3.3 Options	269
11.3.4 Manual Locale Control	270
11.4 Display	273
11.4.1 LCD Backlight	273
11.4.2 Display Properties	273
11.4.3 Editor Properties	273
11.5 Connectivity	273
11.5.1 Remote Display Access.....	273
11.5.2 Remote Data Access (FTP).....	274
11.5.3 Data Point Update	274
11.5.4 OPC-XML-DA.....	275
11.5.5 Auto-Convert to LWEB	275
11.6 Modbus.....	276
11.7 Mail Account.....	276
11.8 Info.....	277
12 Standard Procedures (CEA-709).....	278
12.1 Device Integration	278
12.1.1 New Device.....	278
12.1.2 Pre-Programmed Device	280
12.2 Device Replacement	281
12.2.1 Dynamic NV Projects	281
12.2.2 Static NV Projects	281
12.2.3 Mixed Projects (static and dynamic)	282
12.3 Configuration Change.....	283
12.3.1 Same NV Configuration.....	283
12.3.2 Dynamic NV Change	283
12.3.3 Static NV Change.....	284
12.4 Configuration Recovery	285
12.4.1 Change Device Configuration	285
12.4.2 Change Network Configuration	286
12.4.3 Recover from a failed Update	286
13 Special Functions (CEA-709).....	289
13.1 Binding Management	289
13.2 XIF Export	289
13.2.1 Offline	290
13.2.2 Online.....	290

13.3	Remote Network Interface.....	290
13.4	CEA-709.1-A Operation.....	290
13.5	Feedback Data Points.....	291
13.6	Configuration Properties	291
13.7	UNVTs and UCPTs	293
13.8	Dynamic Network Variables	293
13.8.1	LonMaker® Network Management Tool.....	293
13.8.2	NL220	296
13.8.3	Alex.....	297
13.9	Built-in LONMARK® Objects.....	300
13.9.1	Timer Object	300
13.9.2	Switch Object.....	300
13.9.3	Relay / LCD Backlight control.....	301
13.9.4	HVAC Temperature Sensor	302
14	Special Functions (BACnet)	303
14.1	EDE Export of BACnet Objects.....	303
14.2	BACnet Network Settings.....	303
15	Advanced Topics	305
15.1	Access Control	305
15.2	Multi-Language Projects	306
15.2.1	Configuration	306
15.2.2	Editing multi-language Projects	306
15.2.3	Translation	307
15.2.4	Copy/Paste	309
15.2.5	Compatibility	309
15.3	Project Statistics	309
15.4	Automatic Project Builder.....	310
15.4.1	Options	310
15.4.2	Page Design.....	311
15.4.3	Page Template.....	312
15.4.4	Custom Pages.....	313
15.5	Local Data Point Capacity	313
15.6	Avoiding Update Loops.....	314
15.7	XML Support.....	314
15.8	Format strings.....	315
15.8.1	Number format strings	315
15.8.2	Date format strings	317
15.8.3	Regular Expressions.....	319
15.9	External Storage.....	321
15.10	Security Hardening Guide	321

15.10.1	Device Installation	321
15.10.2	Ports	322
15.10.3	Services	322
15.10.4	Logging and Auditing	323
15.11	Resource Limits	323
15.12	Technical Support	325
16	Script Engine	326
16.1	Overview	326
16.2	Script Resources	327
16.3	Script Objects	329
16.4	Developing Scripts	330
16.4.1	Data Point Integration	330
16.4.2	Using Libraries.....	331
16.4.3	Write a Script Object Resource	332
16.4.4	Open Ports in the Firewall.....	333
16.4.5	Prototyping.....	333
16.4.6	Debugging.....	333
16.5	Prototyping on the PC.....	335
16.5.1	Set up a Project	335
16.5.2	Setting up an Attach Configuration in Visual Studio Code.....	336
17	List of Figures.....	337
18	Revision History	339

1 Introduction

1.1 What is L-VIS?



L-VIS is a highly flexible and easy to configure device to display and control data in BACnet and CEA-709 networks. The L-VIS configuration software can be run as a standalone program as well as an LNS[®] plug-in and supports the use of both static and dynamic network variables in a project to allow easy integration in any kind of CEA-709 network.

The key features of the L-VIS device family are:

- Simple, graphical programming via the supplied configuration software.
- Support for alarming, scheduling and trending.
- Direct interface to other LOYTEC devices via OPC XML-DA. Available data points on the servers are easily imported for use in the LVIS project from the corresponding LINX or L-DALI project files.
- Support for Modbus/TCP in master or slave mode and in parallel to OPC and the control network interfaces (BACnet and CEA709).
- Support for international projects by using UCS-2 encoding, a character set defined by the Unicode standard (ISO-10646), which covers the entire basic multilingual plane (BMP) and allows text in any language to be displayed.

- Display of data as numeric value, user-defined text or graphic elements, horizontal and vertical bars, trend logs, analog meters, or rotating knobs. A special element is available to display date and time as digital or analog clock.
- Input of data via a numeric keypad, alphanumeric keyboard, direct drag of a bar or rotating knob, selection from a dropdown list of texts or graphic elements, or touch, release, short, or long press events (via action objects).
- Mapping of values to colors, to allow an element to change any of its colors depending on the current value of one or more data points.
- Supports layers and transparency for all display elements. Layering is simply defined by the order of objects in the tree view, which may be changed easily by moving the objects around as required.
- Support for numeric operations on data points, including a simple if/else statement.
- Possibility to change location and visibility of objects during device operation based on data point values, allowing implementation of pop-up panels.
- Template system for easy reuse of similar building blocks, with automatic rerouting of data points to a different data set for each instance and limited possibility for individual changes of position, color, or other properties at each template instance. Make sure to read **Section 8.10** for an introduction, and **Section 9.15** for technical details about the template system.
- Input voltage: 24V AC or DC

The CEA-709 control network interface provides:

- Up to 1000 network variables and alias variables.
- 524 address table entries.
- Up to 200 LonMark scheduler objects. Each scheduler object may control up to 64 data points in 107 daily schedules and manage up to 1000 entries in the time/value table. Up to 256 value templates may be defined for each scheduler, with a total data size of up to 1kB per value template.
- An optional LonMark calendar object, defining exception days for the local schedulers. The calendar object supports up to 100 date patterns with a total of up to 1000 date entries.
- Optional alarm server functionality provided by a SNVT_alarm_2 output NV contained in the node object of the device.
- FT-10/LPT-10 or Ethernet-IP852 operation selectable via a jumper on the device or via a setting in the device configuration. When in FT-10 mode, the device can still be reached via its Ethernet port for fast project download.
- A lamp actuator object, to control the backlight from the network. This object was used to control the internal relay on the older LVIS-3ECTB devices.
- A switch object to provide a standard switch and dimmer function, as available on other CEA-709 devices. The switch input for this object is the IN0 connector. Both IN0 and IN1 inputs are also available as system registers, for internal use.

- A temperature sensor object which outputs the current value of the external temperature sensor (optional).
- A real time keeper object, which outputs the current system time.
- 8 L-VIS functional blocks which can be used to place static or dynamic NVs related to the data points used in the project.
- Network scanning to find other device on the network and help in creating the required network variables on L-VIS.

The BACnet interface provides:

- Up to 1000 server objects on current models, and 512 on the LVIS-ME200. COV subscriptions are supported. Available object types are analog, binary and multi-state, each as input, output, or value object.
- BACnet/IP or MSTP connectivity (selectable via device configuration setting) as well as integrated router, BBMD, and slave proxy functions when the device is connected to both networks at the same time.
- Network scanning to find other devices and their objects in the network and use them on the L-VIS device without the need to manually enter device and object instance numbers.
- Supports COV subscription and polling of remote server objects with automatic selection of the best method (COV if available). See also the PICS document for details regarding the BACnet protocol implementation.
- Support for Modbus/RTU and Modbus/TCP. Master and slave mode are supported. Note that Modbus/RTU cannot be used with BACnet/MSTP at the same time, since both protocols use the same physical interface.

All LVIS-3MEx models have the following extended features:

- Support BACnet and CEA709 operation at the same time. LVIS-3ME devices maintain full binary backward compatibility to all projects written for LVIS-3E11x/ME21x devices.
- Provides an OPC UA Server in addition to the OPC XML-DA server.
- Additional Controls like webcam or knob control, as well as support for complex writing systems, which is not available on older hardware due to resource limitations.
- Significantly improved CPU, memory, and storage capacity to support larger and more complex projects than their predecessors.

1.2 Available Models

All available LVIS models along with their basic characteristics are shown in Table 1:

Model	Network Type	Display	Colors	Touch
LVIS-3ME7-Gx	Universal	7" 800x480	256k color ARGB	capacitive
LVIS-3ME12-Ax	Universal	12" 800x600	256k color ARGB	resistive
LVIS-3ME15-Ax	Universal	15" 1024x768	256k color ARGB	resistive
LVIS-3ME15-Gx	Universal	15" 1024x768	256k color ARGB	capacitive
LVIS-3E112 *	CEA-709	12" 800x600	65k color ARGB	resistive
LVIS-3E115 *	CEA-709	15" 1024x786	65k color ARGB	resistive
LVIS-ME212 *	BACnet	12" 800x600	65k color ARGB	resistive
LVIS-ME215 *	BACnet	15" 1024x786	65k color ARGB	resistive
LVIS-3E100 ^{EOL}	CEA-709	5.7" 320x240	256 color VGA	resistive
LVIS-ME200 ^{EOL}	BACnet	5.7" 320x240	256 color VGA	resistive

Table 1: L-Vis Device Models

^{EOL} These models are end-of-life and will not receive any further software updates. The most recent firmware version available for the LVIS-3E100 and LVIS-ME200 is 6.2.8.

* These models are still supported and will receive software updates, but can no longer be ordered. The direct replacement for the 12" models (CEA-709 and BACnet) is the LVIS-3ME12-A1, replacement for the 15" models is either the LVIS-3ME15-A1 (mechanically and optically identical), or one of the models with glass front (LVIS-3ME15-Gx).

This manual also applies to the L-WEB product family, since this configuration software is also used to design the user interface for L-WEB OPC clients, like LWEB-802/803, and the graphical views of LWEB-900.

NOTE:	<i>It is strongly suggested to connect the device to an Ethernet network for management and firmware upgrade, and to allow IP based communication between the touch panel and the control devices in the building. While firmware upgrade and project download is technically possible via the FT-10 interface, any such data transfer will be very slow due to the limited bandwidth. Device management and configuration capabilities will be severely limited since the Web-UI and other web services cannot be used without IP connectivity.</i>
--------------	--

2 What's New in L-Vis

2.1 New in L-Vis 7.2.0

This section describes major changes and new features in L-Vis 7.2.0. Note that this firmware version is not available for the 5" devices LVIS-3E100 and LVIS-ME200. Some features may not be available on older 12" and 15" devices (LVIS-3E11x/ME21x).

2.1.1 New Control: Pie Chart

A new type of control was added to generate a pie or donut chart from a number of connected data points. Refer to Section 9.5.14 for details about pie charts.

2.1.2 Stacked and Side-By-Side Bar Charts

The bar control was extended to support new display modes, like stacked or side-by-side bars from a number of connected data points, bars with a fixed base of zero, growing up or down depending on the displayed value, as well as bars filled with multiple colors or color gradients based on a color mapping table connected to the data point. Please review Section 9.5.5 regarding bar controls to learn more about these new features.

2.1.3 Auto Line Break

Controls producing text output, like number, date, or text controls, can use a new line break feature to automatically insert additional line breaks as needed to make the content fit the control. Please review Section 8.12.2 for a description of this new option.

2.1.4 Color Gradient Mapping

Color map objects provide a new gradient mode. In this mode, the given input values are mapped to intermediate colors, calculated from two adjacent colors in the color table instead of mapping to the exact color assigned to the interval corresponding to the input value. This can be used to create smooth color transitions without the need to define a lot of color entries. Please review Section 9.8 for details.

2.1.5 New Option for Pop-Up Action

Pop-Up actions connected to controls provide a new option to close, when their parent control is hidden. This is useful to automatically close nested popups when their parent popup is closed, or the parent control becomes invisible due to other reasons. Please review Section 9.9.1 to learn more about this new option.

2.1.6 Font Embedding Information

The font manager now displays the font embedding information contained in true type and open type fonts. Check out Section 8.8.3 regarding font embedding to learn more.

2.2 New in L-Vis 7.0.0

This section describes major changes and new features in L-Vis 7.0.0. Note that this firmware version is not available for the 5" devices LVIS-3E100 and LVIS-ME200. Some features may not be available on older 12" and 15" devices (LVIS-3E11x/ME21x).

NOTE:	<i>LWEB projects using version 7.0 format need at least LWEB-802/803 client version 3.4.0. Make sure to update any locally installed LWEB client software before deploying LWEB projects or LWEB-900 graphical views saved in 7.0 format. For better orientation, the firmware version menu was improved and now shows firmware version as well as the associated LWEB client version(s) for each of the available selections.</i>
--------------	---

2.2.1 Multilingual Projects

Projects can now be prepared for multiple display languages. A new system register allows to change the language during run time instantly, without reloading the project. Export and import facilities allow external translation of string resources. Refer to Section 11.3 for information regarding the setup of a multi-language project, and Section 15.2 for a general guide to get started and work with projects in multiple languages.

2.2.2 Localized Controls

Localized display of dates and values is now available consistently across all different types of controls. Labels of trend graphs, bars, knobs, or analog meters respect the current locale. Scheduler and alarm list controls, which only provided a limited number of predefined date formats so far, are now able to display date and time according to the current locale. Please review Section 9.5.9.4 regarding dates in scheduler controls, and Section 9.5.10.2 regarding localized time stamps in alarm list controls.

2.2.3 Improved Trend Control

Firmware 7.0 adds some new features and enhancements to trend controls. Curves can be shifted left and right on the time axis and stacked on top of each other. The area between the curve and its baseline can be filled, display of individual curves can be enabled and disabled during run time, and the trend cursor shows state texts for binary and multistate trends instead of just the numeric value. A new type of data rendering is available to display a dot graph, and a new display mode for the time axis provides two lines of labels with automatic selection of label type (year, month, day, or time) and localized date formatting. Please review Section 9.5.6 to learn about the new capabilities of trend controls.

2.2.4 Improved Bitmap Control

Bitmap controls can be resized even after images were assigned. Any assigned content will automatically be scaled to fit the new control size. Instead of adding copies of the same image in different sizes to the image manager and picking the desired image size for the control, only one size of each image is required. Please review Sections 8.7.2 and 9.5.2 for details regarding this new capability.

2.2.5 Improved Alarm List

Alarm list controls provide new colors for active acknowledged alarms, and alarms that are no longer active but pending acknowledgement. Using the value from a connected analog data point, the entries of the alarm list can be filtered by reporting alarm server, or alarm log data point. Also, actions connected to the control can now use touch and release triggers, usually together with a value condition. These actions can be used to switch to a certain page when the user touches a certain alarm list entry, play a sound, or do other things. Review Section 9.5.10 on alarm list controls for details.

2.2.6 Improved Font System

New fonts can be added to the project directly from the list of installed Windows fonts, using a dropdown list with font preview. Since the font import is now directly available from the font selector dropdown, it is no longer necessary to open the font manager to start using a new font. In addition, the maximum number of fonts was increased to 64. Please review Section 8.8 about fonts to learn more.

2.2.7 Scripting Capability

Scripting capability via *node.js* scripts was added to allow complex calculations or program logic which exceeds what math objects can do. Scripts can also be used to interface with proprietary equipment, using foreign IP based protocols. The script engine is described in Chapter 16.

2.3 New in L-Vis 6.2.0

This section describes major changes and new features in L-Vis 6.2.0

2.3.1 Localized date and number controls

Release 6.2.0 adds support for localized display of date, time, and numbers. A desired locale, consisting of a language, a region, possibly a preferred script, and a calendar system can be set in the project settings and changed later on during run time via a new system register. Depending on the selected region, calendar systems other than Gregorian, like Hebrew, Islamic, Chinese, or Japanese are available to display localized dates.

For number controls, the preferred number system can either be taken from the current locale (default), or a specific number system can be set in the controls properties. The user can choose from over 30 different systems to display numbers, including many decimal systems like Western, Arabic, Devanagari, or Thai digits, as well as algorithmic systems like Roman, Hebrew, Chinese, or Japanese numerals.

Due to the amount of data needed to support localization, this feature is only available on LVIS-3ME devices and LWEB-802/803.

Please review Sections 9.5.4 (Number Control), 9.5.7 (Date Control), and 11.3 (Project Settings: Localization) to get started with this new feature.

2.3.2 Extended date control

Apart from localization support on modern devices, the date control was extended on all devices to add more features like local updates, where incoming time stamps from a remote device are only used to synchronize the date control with external time, while the date control itself is updated independently of any data point updates, an option to choose between UTC time (1:1 display of the time stamp) or local time (add the local UTC offset before display), or a new modern style for the analog clock.

2.3.3 Improved trend control

The trend control received some improvements, like auto-zoom for the Y-axis, different line-width settings per curve, solid grid lines, improved graphical display (buttons always in front of curves), and improved performance.

2.3.4 Knob and Webcam controls

The knob control now supports a new mode of input, where the knob can only be turned incrementally from its current position, to avoid sudden changes of the value. The webcam control was improved to significantly reduce the initial delay before playback of certain

camera streams can start, and to better handle some corner cases with hidden controls and use of input and output data points to control the playback source.

2.3.5 Bidirectional data point references

Data point references on controls can now be marked bidirectional, in which case they are used to receive incoming updates for display, as well as write out new values which were entered via the control. It is no longer necessary to connect the same data point twice in input and output direction.

2.3.6 Improved BACnet support

Some improvements were made to support a project-wide configurable default write priority, as well as a new client map type to receive unsolicited COV updates from other devices.

2.3.7 New system registers

Apart from the new system registers to display and set the current project locale on LVIS-3ME devices, there are new registers to read the current local time and UTC time from date/time structures instead of a single timestamp value, as well as registers which output the current calculated sun position based on device location and local time.

2.4 New in L-Vis 6.1.0

This section describes major changes and new features in L-Vis 6.1.0

2.4.1 Support for complex writing systems

Release 6.1.0 adds support for complex writing systems such as Arabic, Devanagari, Bengali, Thai, Hebrew and many others belonging to the family of Abjad or Abugida scripts. These scripts are often bidirectional and require complex text shaping and layout engines to display correctly.

Even though great care was taken to archive the same layout results for previously supported scripts when using the new layout engine, there are some cases where the new engine will produce different output due to the use of ligatures or other advanced features regarding text layout and composition. To avoid unexpected changes in text layout after a firmware upgrade, existing projects continue to use the old text layout engine until support for complex writing systems is enabled in the project settings. New projects start with the option enabled by default, if the target device supports it.

For preview of LWEB projects, the new layout engine is used at all times, since it closer resembles the expected rendering of the LWEB-802/803 client software.

2.4.2 New Control: Webcam

A new control to display live video streams of web cams in many different formats, including RTSP, was added for LVIS-3ME devices and LWEB projects. Due to hardware limitations, this control is not supported by older LVIS hardware like the LVIS-3E112/115 or the even older LVIS-3E100 model. See Section 9.5.14 for details.

2.4.3 New Control: Rotating Knob

A new control to build rotating knobs was added for LVIS-3ME devices and LWEB projects. Due to hardware limitations, this control is not supported by older LVIS hardware like the LVIS-3E112/115 or the even older LVIS-3E100 model. See Section 9.5.13 for a description of the controls properties.

2.4.4 New Action: Show Pop-Up

A new action type was added to easily build pop-up windows. Instead of placing a collection on the page and controlling visibility via a trigger, the collection can now be moved below a pop-up action and visibility will be controlled by the action. In case the action is connected to a control on the page, the pop-up can be auto-placed near the control triggering the pop-up action. This enables new design patterns, especially for libraries, which were not possible before.

2.4.5 Number Control: Show Units

The number control now supports place holder tags in the format string to automatically insert the effective engineering unit of the connected data point. Together with the new option to select SI or U.S. unit system during run time, projects can now be designed to handle both unit systems without any changes to the project itself.

2.4.6 Sound Action: Playback volume adjustment

To improve control of sound volume for individual sounds, every sound and buzzer action now provides a new option to adjust the playback volume by up to +12dB and -40dB. This allows the project designer to use the same sound in different situations and play it louder or softer as needed. This option is only available on LVIS-3ME devices.

2.4.7 LWEB projects: Option to use client-side fonts

To reduce the size of LWEB projects and speed up loading on the client, fonts in LWEB projects may now be marked as 'installed on the client'. In this case, the original font data will no longer be embedded in the project. Instead, meta-information is stored for the LWEB-802 client to be able to find a suitable pre-installed font on the client machine.

2.4.8 Improved handling of templates

Various improvements have been made to the way templates and their instances are handled. Most importantly, renaming a data point folder which is part of a template or template instance base path no longer breaks the base path. Instead, affected base paths are adjusted to reflect the change. Also, changing the base path of a template tries to remap any data points which were relative to the old path to match available points under the new path. This simplifies changes in data point structure after templates and their instances were already created.

2.5 New in L-Vis 5.1.0

This section describes major changes and new features in L-Vis 5.1.0

2.5.1 Support for new models

This is the first version supporting the new LVIS-3ME model series. Note that since the LVIS-3ME family is based on a different hardware platform, a different firmware image must be used (lvis_3me_5_1_0.dl). The new devices support CEA709 as well as BACnet communication at the same time, so there are no separate models for CEA709 and BACnet. The LVIS-3ME12 and LVIS-3ME15 are 100% backwards compatible to projects created for their predecessors LVIS-3E112/ME212 and LVIS-3E115/ME215 respectively.

2.5.2 Math objects as input data points

L-Vis math objects can now be used in all places where input data points can be used, including cascaded math objects. This saves on internal registers for calculation results and keeps the project simple, because the formula is located at the exact place where its result is being used. It is no longer necessary to locate the math object, identify the register used for the calculation result and searching for the place where this register is finally used as input.

2.5.3 New pre-installed fonts

The setup package contains a number of new true-type fonts for use in L-Vis and L-WEB projects. Have a look at the new choices in *fonts\loytec* and *fonts\truetype*.

2.5.4 More operations supporting undo

The new configuration software now allows changes done to base paths of templates or template instances as well as changes done to an entire batch of data points via the data point reference manager to be undone. For example, a batch reassign of data points can now be undone conveniently in case the reassign lead to undesired results.

2.5.5 L-WEB projects use true-type fonts for the preview

Since recent LWEB-802/803 client software is able to use embedded true type fonts in most cases, the configuration software now also shows the project preview using true type fonts, instead of the pre-rendered pixel fonts which LWEB-802 would use as a fallback in case the embedded true font cannot be used. Note that some combinations of web browser and font may still cause LWEB-802 to fall back to the pre-rendered pixel font.

2.5.6 Improved performance and memory usage

Especially with larger projects, the new configuration software provides a significant performance improvement for many operations, as well as reduced memory consumption.

2.5.7 Dual-Ethernet with Separate Networks

L-VIS models with two Ethernet interfaces can now be configured to work with separate and isolated IP networks. For example, one Ethernet interface can be accessed over HTTPS from a WAN connected to Ethernet 2 while the building network services are running locally on the LAN connected to Ethernet 1. For configuration, the device provides separate Ethernet tabs in the port configuration which allow selecting the offered services on each interface.

2.5.8 WLAN Interface

In combination with the external LWLAN-800 interface, all models equipped with a USB host interface provide new configuration tabs for wireless IP networks. Similar to the second Ethernet interface, one can choose which protocols are available on the wireless network. The wireless interface can be configured as a WLAN client, access point, or mesh node. Using the latter, a wireless mesh network of LOYTEC devices can be built.

2.5.9 OPC UA Server

The new LVIS-3ME model series provide an OPC UA server. This server supports the OPC UA binary protocol and exposes the same OPC tags as the OPC XML-DA server. In addition, OPC UA offers superior security features as well as a more efficient data transfer. For more information on the OPC UA server please refer to Section

2.5.10 Alarm Server Ack-All Data Point

Alarm servers now provide a special `ackAll` property data point. When writing `TRUE` on that data point all currently active alarms on the alarm server are acknowledged.

2.5.11 Format Strings in E-Mails

Data point variables used in e-mail templates can now use format strings to specify their numeric appearance in the e-mail text.

2.5.12 Output NVs with Integrated Feedback

Output network variables (NVs) are used to send updates to remote nodes. In order to poll back the actual value of the remote variable, a feedback data point had to be created for the

output NV. Now changing the direction on the NV from output to value, an integrated feedback function is activated without the need for an extra data point. This becomes especially useful in data point templates.

2.6 New in L-Vis 5.0.0

This section describes major changes and new features in L-Vis 5.0.0.

2.6.1 Usability Improvements

The LCD preview and the object tree were optimized and extended for better usability. It is now easier to work directly in the LCD preview and both views provide better support for working with multiple selected objects.

See the new Sections 8.2 and 8.3 for a detailed description of the new features. There are also changes regarding vector control drawing mode, like the new angle locking function. Please review Section 9.5.11.1 for an update.

2.6.2 Modbus Support

All L-Vis devices are now supporting Modbus. Both Master and Slave mode is supported. In addition to Modbus/TCP, which is supported on all devices and in parallel to all other communication interfaces, BACnet devices can also be configured to support Modbus/RTU instead of BACnet MS/TP on the RS485 interface. Refer to Section 10.6.6 for special properties of Modbus data points and Section 11.6 for new project settings regarding Modbus.

2.6.3 Improved Web-UI

There are new configuration pages to manage the HTTPS server certificate as well as to update the device firmware. Display of the system log and some of the statistics pages have been improved. On some devices, there are now two options to restart the device. The **Reboot Device** command restarts the L-Vis application program only, whereas the **Cold Reboot** command reboots the entire operating system.

2.6.4 Improved OPC Client

Performance and stability of the integrated OPC client, which is used to communicate with LINX devices using OPC XML-DA, has been improved.

2.6.5 Improved OPC Server Management

Especially in combination with the LINX configuration software 5.0.0 or newer, handling of OPC server references in LWEB projects was simplified to be completely automatic for all common use cases. In cases where user intervention is needed, the dialogs presented to the user are now easier to use and understand.

2.6.6 Nested Template Paths

It is now allowed to specify base paths for a template such that one base path is a sub directory of another base path. The new algorithm will assign each data point reference to the most specific base path. See the new section 9.15.6 for a detailed discussion.

2.6.7 Position Lock

Position locking was improved and can now be applied to individual controls, not only to collections. When a control is locked, a small lock sign is added to the normal icon in the tree view. The combined lock/unlock command was changed into separate lock and unlock commands, to make it clear what happens for example when the command is applied to a multi selection. Please review Section 8.5.8 for updated information.

2.6.8 Absolute Paths for Show Page

When using an object path instead of a direct link for a show page action, the path can now start with a forward slash, making it relative to the root menu instead of relative to the page on which the action is located. Please review Section 9.9.2 for updated information on page links in show page actions.

2.6.9 Improved Boot Update Handling

The data point update flag **System Startup** is now independent of the flag **Poll on Startup** which may be available for certain network technologies such as CEA709. In previous versions, setting the System Startup update flag also enabled the Poll on Startup function of the referenced data source. Please see Section 9.6.2 for details on data point updates during system start.

2.6.10 Improved Usage Report

The data point usage report window, which can be accessed via the context menu of a used data point, was improved to show not only usage by other data points, like a connection to a trend or schedule, but also references from objects in the L-Vis project. The button to navigate to one of the referencing objects can be used to move the selection in the object tree and the LCD preview to the corresponding object in the L-Vis project. See Section 10.6.4 for details.

2.6.11 Improved Project Builder

The automatic project builder was improved with new options. Please review Section 15.4.1 for a description of the new options and see Section 15.4.4 about the new possibility to use the project builder to create custom pages for a set of similar devices.

3 Getting Started

3.1 Installing the L-VIS Configuration Application

To install the L-VIS Configuration software, execute the setup program and follow the instruction of the installation wizard. The setup program adds an entry for the L-VIS Configuration software to the start menu. For use with LNS[®] based network management tools, it registers the configuration software as LNS[®] plug-in. Before the LNS[®] plug-in can be used in a network management tool, it has to be registered in the LNS[®] project. Please refer to the documentation of your network management tool how to register an LNS[®] plug-in in a network management project. During the registration process of the L-VIS Configuration software plug-in, LNS[®] device templates are created for the FT-10, and IP-10L (Ethernet) versions of the device.

3.2 Connecting to the Device

There are several ways in which the configuration software can be run and used to configure a device. While this is straight forward in most cases, it is important to know the advantages and limitations of each method for devices connected to a CEA-709 network.

NOTE:	<i>All devices running firmware 4.0 or newer will require user authentication to establish a fully functional connection. The default password for firmware 4.0 is admin. All newer firmware versions use loytec4u as the factory default. Note that a basic connection without password may be possible (inquiry of device model, firmware, and status), but any access to the device configuration will fail if a password is required and was not provided during initial connection.</i>
--------------	--

3.2.1 TCP / IP (Ethernet) Connections

This method connects to the device through a standard TCP/IP channel. It is the recommended way to connect to the device in most cases. Depending on the device model and firmware version, different protocols are available to choose from:

- **Active or passive FTP:** This method is supported by all devices. Note that FTP is an insecure protocol. It is deprecated and disabled by default. If required, it can be enabled via the Web-UI. The PC running the configuration software must be able to establish an FTP and a telnet connection to the device. Depending on the routers and firewalls present between the PC and the device, only active or only passive mode may be working. Active FTP requires special FTP protocol support to be enabled in NAT routers. Passive FTP requires the FTP data port to be open for incoming connections in the firewall. To avoid these requirements, it is highly recommended to use HTTP, HTTPS, or SSH instead.

- **HTTP/HTTPS:** This method uses web services to communicate with the device over a single port (usually port 80 or 443). The device must be running firmware 4.6.0 or newer. Since the same port and protocol is also used by the Web-UI, no additional setup in the firewalls of PCs or routers is necessary. Devices which support encrypted connections via SSH (see below) also support encrypted connections via HTTPS (enable the option to use a secure connection together with the HTTP option). Note that this connection method requires the web server and the application process to run and be responsive to web service requests.
- **SSH:** All devices except LVIS-3M100 and LVIS-ME200 support encrypted connections via SSH (enable the option to use a secure connection but do not enable the HTTP option). In this case, only one TCP port (usually port 22) is used for the entire communication between the PC and the device. This method can be used in cases where the web server of the device is disabled for security reasons or the application is not responding to web service requests, rendering the HTTP method unusable.

If NAT routers or firewalls are located between the PC and the device, the SSH and HTTP/HTTPS protocols are strongly preferred over FTP/Telnet. Check your firewall rules if necessary. Note that SSH and HTTPS connections are not possible for the LVIS-3E100 and LVIS-ME200 devices. Use HTTP for these devices, or make use of a VPN tunnel if connecting to a device in a remote building over the internet.

NOTE:

*In case FTP is used over a wide area network and a connection is not possible, try to change the passive FTP option. In some cases only one of the two methods is working, depending on firewall settings and NAT router capabilities on both ends of the connection. In the worst case, when the configurations on both ends are contradicting each other, neither active nor passive FTP may be possible. If possible, always use encrypted protocols like SSH or HTTPS for WAN connections and **avoid FTP**.*

The IP address, network mask and gateway must be configured via the setup menu on the device. The device must be restarted once for the new settings to take effect.

In the configuration software, the IP address or DNS name of the device can be used to establish a connection. Each connection can be named and saved in a connection list, which is stored in the system so that the list is available when the software is run the next time. When connecting to devices running firmware version 4.0.0 or newer, the correct admin password must be entered in this dialog to establish a connection for project up- and download.

3.2.2 CEA-709 Devices only

Devices supporting the CEA-709 protocol offer more connection methods:

- Through LNS® (software run as LNS® plug-in).
- Via a CEA-709 connection (FT-10 or IP-852).
- TCP/IP (as discussed in the previous section)

Each of these methods has some advantages and some limitations in what they can do. Please see the following detailed description of the individual connection methods.

3.2.2.1 LNS® Connections

This method connects to the device from within the LNS® based network management tool. The software is run as a plug-in and all communication with the device is done through LNS®, meaning that you can use any network interface which is supported by LNS®.

If your device is integrated into the network using an LNS[®] based tool (your device is managed by LNS[®]) and you need to add or remove static network variables from your project or otherwise change the static interface of the device, you **must** use this connection method to download your project, since the configuration software needs to do several adjustments to the LNS[®] database during this process.

If your changes to the project do not cause changes to the static network interface, for example you are just working on the graphics, adding or modifying controls and so on, it is more efficient to use one of the other connection methods. The configuration software will detect any change in the static network interface beforehand and display a warning before you commit to the project download.

Starting with version 4.0.2 of the configuration software, this operating mode can be combined with a parallel TCP/IP connection to transfer project data efficiently while maintaining a connection to the LNS database for database updates.

ADVANTAGES:

- No need to set up TCP/IP connectivity (optional, to speed up project download).
- Easy device selection from within your management software.
- Automatically updates the LNS[®] database and re-commissions the device, if the static interface of the device changes due to the downloaded project.
- Functions are available to export and import current bindings to and from the device into CSV files.
- Any dynamic network variables created on the device will automatically show up as usable data points in your project, including the name of the functional block in which they are located.
- Network scans can be done in the LNS[®] database. This scan method is fast and provides the most complete information about other devices in the network, compared to other scan methods (see also section 3.2.2.3).
- Option to automatically create network bindings from local static NVs to their counterparts on other devices in the network, if the local NVs were created from a network scan and therefore contain the required remote NV information.

LIMITS:

- If no parallel IP connection is used, the transfer of large projects or a firmware upgrade is slow due to the limits of the communication channel. This is especially true for devices in FT-10 mode. Use a recent version (3.2.0 or newer) of the configuration software and the device firmware, to allow the use of ZIP compression for project files. This will considerably speed up the download process. If possible (version 4.0.2 and newer), use a parallel IP connection for project download.

3.2.2.2 CEA-709 (FT-10 / CEA-852) Connections

In this mode, the communication channel is CEA-709 via a LOYTEC NIC such as a NIC-USB, NIC-PCI or NIC-852. The device must first be commissioned (assigned a network address), either by using the same network management tool as used for the other nodes in the network or by entering a suitable unused address in the CEA-709 setup menu. In the connection dialog of the configuration software, a suitable unused subnet/node address must be entered for the network interface card of the PC. The address of the device may either be entered manually as well, or can be detected automatically by either entering the unique node ID or by pressing the status button on the device. Once the address is known,

the connection can be given a name and saved just like the TCP/IP connections, so that the procedure is only necessary when the address of the device changes.

Connecting via CEA-709 is most suitable when the device is not connected to the IP network and a non-LNS network management tool is used for integration.

ADVANTAGES:

- Works even when the device is not connected to an IP network, thus no need to set up TCP/IP settings for the device.
- Download is faster compared to LNS® connections, due to an optimized file transfer protocol.
- Works independent of the used network management tool.

LIMITS:

- Only works when the device is commissioned. A download of a new project which changes the static interface of the device will therefore break the connection, since the device will have to reset its CEA-709 network configuration in that case. The device must then be re-commissioned in the network management tool and the connection must be re-established.
- Especially in FT-10 mode, project downloads can still be slow. The same is true for firmware downloads. If possible, the TCP/IP connection method should be used for such tasks. Transfer speed is considerably improved when there are no routers with limited buffer size between the PC and the device. Routers which are able to handle large packets (254 byte) will have no real impact on the transfer speed. Routers which allow a maximum packet size of 64 bytes only will degrade the transfer speed.
- Only network interface hardware from LOYTEC can be used for this type of connection. Any LOYTEC NIC may be used to run the configuration software, that is, the software is not registered to specific network interface hardware.
- There is currently no way to conduct a network scan using this connection method, since the built-in scanner engine on the device can only be controlled using a TCP/IP connection.

3.2.2.3 TCP / IP Connections

As discussed earlier, this method connects to the device through a standard TCP/IP channel. Regarding integration of CEA-709 capable devices, the following advantages and disadvantages should be considered.

ADVANTAGES:

- Very fast data transfer, suitable for large projects.
- Best way to update the firmware of a device fast and secure.
- Communication is possible even when the device is not commissioned or otherwise not reachable on the CEA-709 port.
- No LNS® or special network interface hardware is required. The standard Ethernet port which is available on every PC is enough.

LIMITS:

- No connection to LNS (if present) and therefore no way to update the database in case the new project changes the static interface of the device. Projects which add or remove static network variables or otherwise change the static interface while dynamic NVs are present may not be loaded via this method, since the database will be out of sync after the download and cannot be repaired by the user (only the configuration software can apply the necessary changes). Use the LNS connection method in this case (see section 3.2.2.1).
- Any dynamic network variables which were created on the device using a network management tool will show up as usable data points in the project, but the available information is limited to the data stored in the device. For example, the functional block in which the NVs were created will be unknown since this information is only stored in the LNS database and is not available on the device.
- Network scans are conducted by the device itself, using a built-in scanner engine. This scan is slower than an LNS[®] scan and cannot provide as much information, for example device names will have to be set manually.

3.3 Quick Start Tutorial CEA-709

This tutorial describes the steps to create a first demo project and load the project into a CEA-709 device, like the LVIS-3E100. The tutorial assumes that the configuration software is already installed and the LNS[®] plug-in is registered as described in the prior section. For the LNS[®] specific part of this tutorial, the LonMaker[®] 3.1 network management tool is used. Other network management tools, which are based on LNS[®] 3 or above, which support dynamic network variables and the LNS[®] Plug-in Interface can be used as well for this tutorial (e.g. Newron's NL220, SPEGA's Alex 3 Professional).

3.3.1 Add the device to the network

The first step is to integrate the device into an existing network. Depending on the device type and the type of network used, this step is either done on the device or on the PC using network management software. In any case, the device first needs to be physically connected to the network and a suitable power supply.

3.3.1.1 CEA-709 with LNS: Create the L-VIS device in the LNS database

Add the L-VIS device to your LNS network project. Since the L-VIS configuration plug-in already creates the device templates, the device configuration can be read from the device template. Choose the device template which matches the network interface mode (FT-10 or IP-10L) of your device. The Node ID of the device can be entered manually. Optionally, a service pin message can be sent by pressing the status button located at the bottom of the device, or by selecting the Send Service Pin Message command in the Setup→Commands menu of the L-VIS device (see 7.3).

Set the device configured online during the commission phase, so that the device will be able to communicate with other nodes on the network and with the configuration software.

NOTE: *Do not integrate the device into your project by uploading the device interface properties from the device before you registered the plug-in. This is important because if you upload the definitions from the device, you will create a template of your own and the configuration software will then fail to create the real device template, once you register it. In this case, you must remove the device and the template from the database and re-run the plug-in registration.*

3.3.1.2 CEA-709 without LNS: Assign a network address

Use your network management tool to add the device to the database and assign a network address to it. Alternatively, go to the CEA-709 page of the setup menu and enter a suitable network address manually, then press the button ‘SET ONLINE’ to activate the address.

3.3.2 Create Dynamic NVs (LNS only)

In this example, we will create two dynamic network variables of type SNVT_switch on the device, one input and one output, and bind them together (turnaround binding) to show how to display and send out data.

NOTE: *If your network management tool does not support dynamic NVs, just skip this step. The required data points will then be created from within the configuration software in a later step (as static NVs).*

First, we need a functional block to place the new network variables in. Create one of the 8 available L-VIS functional blocks (for example L-VIS[0]) to hold the two variables. Next, create the two switch variables. For the complementary NV required to create the dynamic NV, browse to the Switch object of your L-VIS device and select one of the two switch variables available there (the direction does not matter). More information about dynamic network variables can be found in section 13.5.

NOTE: *Make sure to set the poll attribute to CLEAR when you create the output variable, otherwise you will not be able to send out any values through this NV. Also, the poll attribute of the input NV should be SET, so that L-VIS is able to fetch an initial value when it boots up.*

Now bind the output NV to the input NV, so that we can send/receive values.

3.3.3 Start the L-VIS Configuration Software

If possible, run the configuration software in plug-in mode from within your network management tool. Open the context menu of the L-VIS device block in your drawing and select Configure to start the configuration software.

NOTE: *For network management tools which do not support plug-ins, start the program in standalone mode and connect to your device via TCP/IP or CEA-709 over a LOYTEC NIC.*

When started in plug-in mode, a connection dialog will appear and ask for an optional IP connection to the device. If the device can be reached via TCP/IP, enter the required data here and press ‘Connect’, otherwise just press ‘Cancel’ to continue without IP connection.

The configuration software contacts the device, finds out what model and firmware version it is, selects the correct settings and disables the corresponding menus, so that you may not change them. The main window will show an empty L-VIS project.

The list of dynamic network variables available on the device is automatically read in and a list of corresponding data points to access the NVs is generated. These data points will be used in a later step. If no dynamic NVs were created in the previous step, the required NVs and data points will be created in the configuration software later on.

NOTE: *In case the device was not commissioned yet, or there are other communication problems, the plug-in will display a warning message, saying that it could not communicate with the device and therefore was not able to determine model or firmware version. In this case, you have to commission the device or find out why it cannot be commissioned. Typical errors are devices in the wrong mode (jumper set for CEA-852 but connected to FT-10 or vice versa) or communication problems on the channel itself. Especially on a channel as complex as CEA-852, there are a number of possible reasons why you may not be able to communicate with the device.*

The main window is divided in three major sections:

Tree View: On the left side of the main window is a tree view showing all objects you created so far and how they relate to each other, for example a menu object may have menu item objects attached to it, a menu item object may have page objects and sub-menu objects attached, and a page object contains the elements to display and enter data (these elements are called *controls* in this manual). You may navigate through this tree and select an object from it or drag and drop objects to move them around in the hierarchy, for example grab a control from one page and drop it on another page. An empty project contains only a root menu and a folder object where global objects can be placed, which are not directly related to the user interface, for example alarm generators.

Property View: On the top right of the main window is a property view, which shows a number of property pages, where you can adjust the properties of the currently selected object. The number and kind of property pages changes according to the selected object, but some of the more basic property pages are available all the time. Feel free to browse through the available pages and see what you can adjust there.

LCD Preview: To the lower right of the main window is the preview of the project as seen on the LCD when the project is downloaded. This is not a functional simulation of the L-VIS device, but a layout guide and preview of individual menus and pages. The preview changes according to the currently selected object. Since the configuration software comes with a library of the layout code used in the various firmware versions of the device, the preview is always accurate to the pixel. Should the layout or appearance of controls change with firmware versions, the configuration software is able to track these changes and display the page exactly as it will appear on the device you are currently connected to.

NOTE: *You may **select, move, and resize** your controls directly inside the LCD preview. Also, there are context menus available for each control, which you can open by a right click on the control in the preview. The cursor keys can be used to move controls one pixel at a time and to resize the control, when the shift key is held down.*

3.3.4 Create Static NVs (if no dynamic NVs available)

In case the network management tool used for the tutorial does not support dynamic creation of NVs on the device, the NVs required for this tutorial must now be created manually using the data point manager window. This window is accessible either from the tool bar (the icon left to the connect button), or from the main menu *Edit -> Datapoints....* Please skip forward to section 10.16.2 for a description of how to create local network variables. Following this description, create one input NV and one output NV, both of type *SNVT_switch*.

3.3.5 Create Menu Structure and Pages

The next step is to design a simple test page for this tutorial and use the two data points. The empty project shows the root menu with no entries (menu items). You may change the name of the root menu as well as its colors and other properties, such as the width of the menu or its scroll-bar, using the property pages to the right of the tree view (make sure the root menu object is selected, which should be the case right after you started the software). The menu object is described in detail in section 9.2.

First, add a menu item to the root menu. To do this, right click on the root menu object in the tree view and select *Add Item* from the context menu. A new menu item is created and connected to the menu object. The new menu item is automatically selected, so that you may browse the property pages and see what you want to change. The name of the item object can be changed in the *Name* box of the **General** property page. This changes the name of the object in the object tree and has no effect on the appearance of the menu item on the device. To change the text of the item itself, go to the **Common Properties** page and change the contents of the *Text* box. The display in the preview window changes as you type, so you can immediately see the results. You may want to change the font to the 12x16-ROM-Fixed-R font, so you get a larger item which is easier to select on the touch screen. The menu item object is described in detail in section 9.3.

Repeat the steps above to add another menu item and call it 'Setup'. Again, change the *Text* of the item to read something like 'Setup' and change the font to make the item a comfortable size. This item will be used to enter the devices setup menu.

NOTE:	<i>The device will check the last item of the root menu and see if it has any pages or sub-menus attached to it. If the item has no further objects attached (this is called a return item) and is located at the last position in the root menu, then the device will connect its setup menu to this item. If the item is used, the device will create a default item to reach the setup menu, but this will most likely not fit the design of your other menu items, so it is best to always add a setup item yourself and configure it the way you want it to look.</i>
--------------	--

Finally, add a page to the first of the two menu items. This page will be displayed when the user selects the first menu item from the root menu. Open the context menu of the menu item (right click on the item in the tree view) and select *Add Page*. In order to see this page right after the device started, make this page the projects default page. Open the context menu of the page in the tree view and select *Set as Default*. The page object is described in detail in section 9.4.

3.3.6 Add Controls to the Page

The next step is to place some objects on the page to view and control the values of the two network variables. These objects are called *controls* in this manual, since they allow the user to control the data points and display their current value. A number of different control styles are available to choose from. In this example, we want to display the current status of the switch input using a text control for the state and a numeric control for the value. To control the switch output, we use a bitmap control for the state and a bar control for the value.

To create the required controls, either right-click on the page object in the tree view and choose the control type you want from the context menu, or select the page in the tree view and right click in the LCD preview area to reach the same context menu. Add one text control, one bitmap control, one numeric and one bar control to your page.

Now place the controls on your page so that they do not overlap each other. Grab and drag them with the mouse or enter position and size in the *Screen Coordinates* section of the **Common Properties** page. To resize the controls with the mouse, move the pointer to the lower right corner of the controls area in which the value is displayed, that is, disregard any decorations around the value area, like the scale of a bar control. For most control types, the value area fills the entire space of the control, except for the selection frame. For more complex controls, like bars and trend logs, you can view the value area by changing the *Container* color on the **Color** property page. Then grab the control at the lower right corner of the container area.

NOTE: *The position and size you set on the **Common Properties** page also refers to the value area of the control, not to the selection frame around it. This is done so that for example the width and height of a bar does not change when you enable or disable tick marks or scale. These are just seen as decoration which can be turned on and off while the area of the value display remains stable.*

Adjust the properties of each control as you like or leave them at the defaults, which should be fine for this demo. Bitmaps will be added to the bitmap control in a later step.

3.3.7 Connect the Data-Points

At this point, connect data points to the controls. Especially for text and bitmap controls, it helps to connect the data points early, since the configuration of the controls property will depend on the data point connected (you will see why this is the case in a minute).

To add a data point to a control, select *Add Data Point...* from the controls context menu. A dialog will appear which is used to manage and select data points. This dialog is explained in detail in chapter 9.14, for now we just use it to select our data points, which should have been created for us already. Select the folder *Local NVs* from the tree view at the left of the dialog. There should be two items, one for the input NV and one for the output NV we created earlier.

NOTE: *In case you created the dynamic NVs after you started the plug-in, the data points will not be there yet. In this case, close the data point dialog and push the button two positions right from the connect button. The tool-tip of this button reads 'Update Data Points'. It will re-scan the currently available dynamic NVs on the device and create the required data points for you. If you added NVs while the plug-in was open, hit the update button to get the required data points added to your project automatically.*

From the two items available in the data point list, select the input data point and expand its items by a left click on the small plus sign left of the data point. The expanded view will show two sub-elements, one for the state component of the NV and one for the value component. A data point is selected by a double click or by selecting one or more points and clicking the *Select* button.

Use the above procedure to add the state part of the input NV to the text control, the value of the input NV to the numeric control, the state of the output NV to the bitmap control and the value of the output NV to the bar control.

NOTE: *When you add the state of the input NV to the text control, another object called a mapping table will be created automatically and will be initialized to map the three known states of the switch state to standard state texts. You may select this mapping table and change the texts as well as assign different colors to the individual texts in the mapping table. Mapping tables are discussed in detail in section 9.7, for now you can leave the table alone.*

All the controls should now have one data point connected, with the text control and the bitmap control having a data point and a mapping table. The bitmap control needs some more work, since we need to assign graphics to the individual values of the connected data point.

To do this, select the mapping table object which was automatically created and attached to the bitmap control. On the **Mapping** property page, select one of the entries in the list, for example the entry for the value -1, which maps to the text SW_NUL. Click on the button *Select...* and select a suitable graphic to show for this state. A library of icons in PNG format is installed together with the configuration software. You may also draw your own graphics as well. For best results, save them in PNG format for icons including transparency or JPG format for full-page background images.

NOTE: *Almost any bitmap format like BMP, JPG, GIF, PNG, TIF, and so on may be used. Vector graphics in SVG format can be imported as well, but some of the more advanced features of SVG are currently not supported. However, these are normally not present in icon images. Working with graphics is covered in section 15.7 of the manual.*

Once you selected a bitmap file, the bitmap control will resize itself automatically to fit the selected icon. Repeat the process for the remaining states until all states have a bitmap assigned to them. Use different bitmaps, so that you can distinguish between the individual states. The size of the control will always be the width of the widest icon and the height of the tallest icon.

You can test the look of your bitmap control by selecting the individual states in the mapping table and watch the LCD preview. The state which is selected when you write the project to the device will become the default state of the control.

3.3.8 Write the Project to the L-VIS device

We are now ready to write the project to the L-VIS device, but first the project should be saved in a file on the PC. Although it is possible to read the project back from the device, it should not be considered a 100% safe place for the project to be, since the device may erase the project in case it finds a severe error which would otherwise prevent it from booting up again. If the erase jumper at the back of the device is set, or the *Remove Configuration...* command is selected from the **File** menu, the project in the device may be lost as well, so it is a good idea to have a backup copy ready on the PC. To save the project, select *Save as...* from the **File** menu and enter a project name. To write the project to the L-VIS device, press the button with the red down arrow in the toolbar or select *Write Project to the Device...* from the **Connections** menu and confirm the dialog.

NOTE: *If you created static NVs, a warning message will appear, stating that the static interface of the device will change and that the software will need to adjust the definition of the device in the database. If you commissioned the device in the database already, make sure to download the project in plug-in mode, otherwise it will be necessary to replace the device in the database (new program ID) and re-commission. See Chapter 11 about standard procedures for CEA-709 devices for detailed information about this topic.*

Once the project file is written to the device, the L-VIS reboots and shows the tutorial project on the display.

If the NVs were created statically in the project, they should now be available on the device in the network management tool (a refresh of the display or database in the network management tool may be required to make them appear). You can now bind the output NV and the input NV together. In case of dynamic NVs, this was already done earlier.

3.3.9 Test the Project

You should now be able to use the device to set the state and the value of the output switch NV and immediately see the results on the input NV, since the two were bound together.

To change the state, press the bitmap control. A dropdown list with the tree icons you assigned to the tree states in the mapping table will appear. Now select the desired icon to set the state.

NOTE: *Whenever an element is selected from a dropdown list of icons or texts, the device will search the mapping table to find the value associated with the selected element. This value will then be assigned to the output data points which are connected to the control.*

The new state will be sent out and will be received immediately by the input NV. The new state should now be displayed on the text control, using the texts which you specified in the mapping table.

To change the value, select the bar control and move the bar up and down. The current value will be sent out, received back in, and should be visible immediately on the numeric control.

3.4 Quick Start Tutorial BACnet

This tutorial describes the steps to create a first demo project and load the project into a BACnet model, like the LVIS-ME200. The tutorial assumes that the configuration software is already installed on the PC.

3.4.1 Add the device to the network

The first step is to integrate the device into the network. Power on the device and wait until the boot process has finished. Open up the menu and go to the TCP/IP setup page to set a suitable IP address, network mask and gateway for the device or turn on DHCP to enable automatic IP configuration, if available in your network. Switch to the BACnet setup page and enter the desired device ID number. If the device is connected via MSTP, set the correct baud rate and node number, if connected via BACnet/IP set the desired port number (the default should be OK in most cases).

NOTE:	<i>The device can only operate either on MSTP or BACnet/IP. The currently selected interface is shown on the BACnet setup page for your reference. To select the desired interface, either set the jumper at the back of the device as explained in section 6.1 (LVIS-ME200) or activate the desired interface directly on the BACnet setup page or via the web UI.</i>
--------------	---

Reboot the device for the changes to take effect.

3.4.2 Start the L-VIS Configuration Software

Run the installed L-VIS configuration software from the start menu of your desktop. Connect to the device by pressing the *Connect* button from the tool bar, or select the menu item *Connect to the Device...* from the *Connection* menu of the main window.

In the connection dialog, enter the IP address or DNS name of the device, the desired connection mode (FTP/Telnet or secure connection via SSH if your device is not an LVIS-ME200), adjust the port numbers if necessary and enter the admin password if the device firmware is 4.0.0 or newer.

The configuration software contacts the device, finds out what model and firmware version it is, selects the correct settings and disables the corresponding menus, so that you may not change them. The main window will show an empty L-VIS project.

The main window is divided in tree major sections:

Tree View: On the left side of the main window is a tree view showing all objects you created so far and how they relate to each other, for example a menu object may have menu item objects attached to it, a menu item object may have page objects and sub-menu objects attached, and a page object contains the elements to display and enter data (these elements are called *controls* in this manual). You may navigate through this tree and select an object from it or drag and drop objects to move them around in the hierarchy, for example grab a control from one page and drop it on another page. An empty project contains only a root menu and a folder object where global objects can be placed, which are not directly related to the user interface, for example alarm generators.

Property View: On the top right of the main window is a property view, which shows a number of property pages, where you can adjust the properties of the currently selected object. The number and kind of property pages changes according to the selected object, but some of the more basic property pages are available all the time. Feel free to browse through the available pages and see what you can adjust there.

LCD Preview: To the lower right of the main window is the preview of the project as seen on the LCD when the project is downloaded. This is not a functional simulation of the L-VIS device, but a layout guide and preview of individual menus and pages. The preview changes according to the currently selected object. Since the configuration software comes with a library of the layout code used in the various firmware versions of the device, the preview is always accurate to the pixel. Should the layout or appearance of controls change with firmware versions, the configuration software is able to track these changes and display the page exactly as it will appear on the device you are currently connected to.

NOTE:	<i>You may select, move, and resize your controls directly inside the LCD preview. Also, there are context menus available for each control, which you can open by a right click on the control in the preview. The cursor keys can be used to move controls one pixel at a time and to resize the control, when the shift key is held down.</i>
--------------	---

3.4.3 Create Data Points

In order to have some data points which can be controlled and displayed, it is necessary to either create some server objects on the L-VIS device or have other BACnet devices accessible in the network. To keep the tutorial independent of other devices, we use the data point manager window to create a few local server objects. This window is accessible either from the tool bar (the icon left to the connect button), or from the main menu *Edit -> Datapoints....* Please skip forward to section 10.16.3 for a description of how to create local server objects. Following this description, create one analog input object and one binary input object. You should get one output data point for each of the input server objects. Using these data points, the value of the server objects can be written.

To demonstrate how to display the value of remote server objects (usually located on other devices in the network), we will use our own server objects and pretend they are located on a different device. To do so, we first need to download the project to the device, so that the server objects are actually created on the device. Press the button with the red down arrow in the toolbar or select *Write Project to the Device...* from the **Connections** menu and confirm the dialog. After the device rebooted, open the data point manager window again, right-click on the folder named *BACnet Network Scan* and select the item *Scan BACnet Network....* Discover the devices on the network, select the L-VIS device, and scan it for objects. Close the scan dialog and use the two server objects found by the scanner to create corresponding client mappings. Please see section 10.16.4 for a detailed description.

At this point, there should be two data points in the folder called *Server Objects* and two data points in the folder called *Client Mappings*. The client mapping points are input points which basically refer to the current value of the server objects located on our own device, but the same procedure can be used to refer to any other server objects on other devices.

3.4.4 Create Menu Structure and Pages

The next step is to design a simple test page for this tutorial and use the data points to set and display values. The empty project shows the root menu with no entries (menu items). You may change the name of the root menu as well as its colors and other properties, such as the width of the menu or its scroll-bar, using the property pages to the right of the tree view (make sure the root menu object is selected, which should be the case right after you started the software). The menu object is described in detail in section 9.2.

First, add a menu item to the root menu. To do this, right click on the root menu object in the tree view and select *Add Item* from the context menu. A new menu item is created and connected to the menu object. The new menu item is automatically selected, so that you may browse the property pages and see what you want to change. The name of the item object can be changed in the *Name* box of the **General** property page. This changes the name of the object in the object tree and has no effect on the appearance of the menu item on the device. To change the text of the item itself, go to the **Common Properties** page and change the contents of the *Text* box. The display in the preview window changes as you type, so you can immediately see the results. You may want to change the font to the

12x16-ROM-Fixed-R font, so you get a larger item which is easier to select on the touch screen. The menu item object is described in detail in section 9.3.

Repeat the steps above to add another menu item and call it 'Setup'. Again, change the *Text* of the item to read something like 'Setup' and change the font to make the item a comfortable size. This item will be used to enter the devices setup menu.

NOTE:	<i>The device will check the last item of the root menu and see if it has any pages or sub-menus attached to it. If the item has no further objects attached (this is called a return item) and is located at the last position in the root menu, then the device will connect its setup menu to this item. If the item is used, the device will create a default item to reach the setup menu, but this will most likely not fit the design of your other menu items, so it is best to always add a setup item yourself and configure it the way you want it to look.</i>
--------------	--

Finally, add a page to the first of the two menu items. This page will be displayed when the user selects the first menu item from the root menu. Open the context menu of the menu item (right click on the item in the tree view) and select *Add Page*. In order to see this page right after the device started, make this page the projects default page. Open the context menu of the page in the tree view and select *Set as Default*. The page object is described in detail in section 9.4.

3.4.5 Add Controls to the Page

The next step is to place some objects on the page to view and control the values of the two local server objects. The objects on the page are called *controls* in this manual, since they allow the user to control the data points and display their current value. A number of different control styles are available to choose from. In this example, we want to display the current status of the binary object using a text control and the value of the analog object using a numeric control. To control the binary objects value, we use a bitmap control and for the analog objects value a bar control.

To create the required controls, either right-click on the page object in the tree view and choose the control type you want from the context menu, or select the page in the tree view and right click in the LCD preview area to reach the same context menu. Add one text control, one bitmap control, one numeric and one bar control to your page.

Now place the controls on your page so that they do not overlap each other. Grab and drag them with the mouse or enter position and size in the *Screen Coordinates* section of the **Common Properties** page. To resize the controls with the mouse, move the pointer to the lower right corner of the controls area in which the value is displayed, that is, disregard any decorations around the value area, like the scale of a bar control. For most control types, the value area fills the entire space of the control, except for the selection frame. For more complex controls, like bars and trend logs, you can view the value area by changing the *Container* color on the **Color** property page. Then grab the control at the lower right corner of the container area.

NOTE:	<i>The position and size you set on the Common Properties page also refers to the value area of the control, not to the selection frame around it. This is done so that for example the width and height of a bar does not change when you enable or disable tick marks or scale. These are just seen as decoration which can be turned on and off while the area of the value display remains stable.</i>
--------------	---

Adjust the properties of each control as you like or leave them at the defaults, which should be fine for this demo. Bitmaps will be added to the bitmap control in a later step.

3.4.6 Connect the Data-Points

At this point, connect data points to the controls. Especially for text and bitmap controls, it helps to connect the data points early, since the configuration of the controls property will depend on the data point connected (you will see why this is the case in a minute).

To add a data point to a control, select *Add Data Point...* from the controls context menu. The dialog which was used before to create the required data points will appear for selection of the desired data points. This dialog is explained in detail in chapter 9.14. For now we just use it to select the data points which should be connected to the control. A data point is selected by a double click or by selecting one or more points (multi-select is possible using the shift key) and using the *Select* button.

Use the above procedure to add the binary client mapping to the text control, the analog client mapping to the numeric control, the binary server object to the bitmap control and the binary client mapping to the bar control.

NOTE:	<i>When you add the binary point to the text control, another object called a mapping table will be created automatically and will be initialized to map the two known states of the binary data point to the state texts listed in the objects properties. You may select this mapping table and change the texts as well as assign different colors to the individual texts in the mapping table. Mapping tables are discussed in detail in section 9.7, for now you can leave the table alone.</i>
--------------	---

All the controls should now have one data point connected, with the text control and the bitmap control having a data point and a mapping table. The bitmap control needs some more work, since we need to assign graphics to the individual values of the connected data point.

To do this, select the mapping table object which was automatically created and attached to the bitmap control. On the **Mapping** property page, select one of the entries in the list, for example the entry for the value 0, which maps to the text OFF. Click on the button *Select...* and select a suitable graphic to show for this state. A library of icons in PNG format is installed together with the configuration software. You may also draw your own graphics as well. For best results, save them in PNG format for icons including transparency or JPG format for full-page background images.

NOTE:	<i>Almost any bitmap format like BMP, JPG, GIF, PNG, TIF, and so on may be used. Vector graphics in SVG format can be imported as well, but some of the more advanced features of SVG are currently not supported. However, these are normally not present in icon images. Working with graphics is covered in section 15.7 of the manual.</i>
--------------	--

Once you selected a bitmap file, the bitmap control will resize itself automatically to fit the selected icon. Repeat the process for the remaining states until all states have a bitmap assigned to them. Use different bitmaps, so that you can distinguish between the individual states. The size of the control will always be the width of the widest icon and the height of the tallest icon.

You can test the look of your bitmap control by selecting the individual states in the mapping table and watch the LCD preview. The state which is selected when you write the project to the device will become the default state of the control.

3.4.7 Write the Project to the L-VIS device

We are now ready to write the project to the L-VIS device, but first the project should be saved in a file on the PC. Although it is possible to read the project back from the device, it should not be considered a 100% safe place for the project to be, since the device may erase the project in case it finds a severe error which would otherwise prevent it from booting up again. If the erase jumper at the back of the device is set, or the *Remove Configuration...* command is selected from the **File** menu, the project in the device may be lost as well, so it is a good idea to have a backup copy ready on the PC. To save the project, select *Save as...* from the **File** menu and enter a project name. To write the project to the L-VIS device, press the button with the red down arrow in the toolbar or select *Write Project to the Device...* from the **Connections** menu and confirm the dialog.

Once the project file is written to the device, the L-VIS reboots and shows the tutorial project on the display.

3.4.8 Test the Project

You should now be able to use the device to set the state and the value of the two server objects and immediately see the results on the client mapping input points, since they refer to the local server objects.

To change the binary server object state, press the bitmap control. A dropdown list with the two icons you assigned to the ON and OFF states in the mapping table will appear. Now select the desired icon to set the state.

NOTE:	<i>Whenever an element is selected from a dropdown list of icons or texts, the device will search the mapping table to find the value associated with the selected element. This value will then be assigned to the output data points which are connected to the control.</i>
--------------	--

The new state will be written to the server object and will be received immediately by the client mapping input. The new state should now also be displayed on the text control, using the texts which you specified in the mapping table.

To change the value of the analog server object, select the bar control and move the bar up and down. The current value will be written to the object, received back in through the client mapping, and should be visible immediately on the numeric control.

4 Mechanical Installation

4.1 Dimensions and Mounting

For information about unit installation, please refer to the installation sheet that came with your unit. This document is also available for download from the LOYTEC website.

4.2 Theft Protection

To protect devices of type LVIS-3E100 and LVIS-ME200, they can be fixed to the wall mount frame with the supplied chain as shown in Figure 1.



Figure 1: Connecting L-VIS to the wall mount frame.

5 Electrical Installation

5.1 Connection diagram

For information about unit installation, please refer to the installation sheet that came with your unit. This document is also available for download from the LOYTEC website.

5.2 Electrical Characteristics LVIS-3E100/ME200

The electrical characteristics of the LVIS-3E100 and LVIS-ME200 devices are shown in Table 2.

Characteristics	Value
Operating voltage	24 VDC or 24 VAC \pm 10%
Power consumption	4 W, idle: 2W (backlight off)
In-rush current	Up to 1000 mA @ 24 VAC
Operating temperature (ambient)	10°C to +40°C
Storage temperature	-10°C to +50°C
Humidity (non-condensing) operating	10 to 90 % RH @ 50°C
Humidity (non-condensing) storage	10 to 90 % RH @ 50°C
Switch input	Floating contact
Ethernet connection	100Base-T
CEA-709	FT-10/LPT-10
BACnet/MSTP	RS485, 1 Unit Load
Enclosure	Stainless steel, anodized aluminum front panel E6C0
Electrical Appliance Protection Class (IEC 61140)	Class I
IP Code (IEC 60529)	Front: IP54 / Back: IP10
LCD display	320x240 pixels, 256 colors
LCD backlight	LED backlight, 40 000 hours / auto off
Installation	Wall mount

Table 2: Electrical characteristics 3E100/ME200

5.3 Electrical Characteristics LVIS-3E11x/ME21x

The electrical characteristics of the LVIS-3E11x and LVIS-ME21x devices are shown in Table 3.

Characteristics	Value
Operating voltage	24 VDC \pm 10%, or 90-265 VAC
Power consumption	12 inch: 14W, 15 inch: 17W, idle: 3.5W
In-rush current	Up to 3.7A @ 24 VAC, max. 40ms
Operating temperature (ambient)	10°C to +40°C
Storage temperature	-10°C to +50°C
Humidity (non-condensing) operating	10 to 90 % RH @ 50°C
Humidity (non-condensing) storage	10 to 90 % RH @ 50°C
Switch input	Floating contact
Ethernet connection	2x 100Base-T, integrated Ethernet switch for daisy-chaining up to 20 devices
CEA-709	FT-10/LPT-10
BACnet/MSTP	RS485, 1 Unit Load
Enclosure	Stainless steel, anodized aluminum front panel E6C0
Electrical Appliance Protection Class (IEC 61140)	Class I
IP Code (IEC 60529)	Front: IP54 / Back: IP10
LCD display	12 inch: 800x600 pixel, 15 inch: 1024x768 pixel, 65k color TFT
LCD backlight	LED backlight, 50 000 hours / auto off
Installation	Wall mount

Table 3: Electrical characteristics 3E11x/ME21x

5.4 Electrical Characteristics LVIS-3MEx

The electrical characteristics of the LVIS-3MEx devices are shown in Table 4.

Characteristics	Value
Operating voltage	24 VDC \pm 10%, or 90-265 VAC (except 7" models)
Power consumption	See Table 5
Operating temperature (ambient)	10°C to +40°C
Storage temperature	-10°C to +50°C
Humidity (non-condensing) operating	10 to 90 % RH @ 50°C
Humidity (non-condensing) storage	10 to 90 % RH @ 50°C
Switch input	Floating contact
Ethernet connection	2x 100Base-T, integrated Ethernet switch for daisy-chaining up to 20 devices
CEA-709	FT-10/LPT-10
BACnet/MSTP	RS485, 1 Unit Load
Enclosure	Stainless steel
Front (-A models)	Anodized aluminum front panel E6C0
Front (-G models)	Glass
Electrical Appliance Protection Class (IEC 61140)	Class I
IP Code (IEC 60529)	Front: IP54 / Back: IP10
LCD display	See Table 1
LCD backlight	LED backlight, 50 000 hours / auto off
Installation	Wall mount

Table 4: Electrical characteristics 3E11x/ME21x

The following table summarizes power consumption for the LVIS-3ME family.

LVIS-3ME7	Supply	Power	Current	Peak
Active	24 VDC	5.0 W	250 mA	310 mA
Standby	24 VDC	2.5 W	160 mA	220 mA
LVIS-3ME12	Supply	Power	Current	Peak
Active	24 VDC	9.3 W	410 mA	490 mA
	110 VAC	11.3 W	210 mA	880 mA
	230 VAC	13.3 W	140 mA	720 mA
Standby	24 VDC	2.9 W	120 mA	240 mA
	110 VAC	5.4 W	90 mA	550 mA
	230 VAC	5.9 W	70 mA	370 mA
LVIS-3ME15	Supply	Power	Current	Peak
Active	24 VDC	9.4 W	440 mA	510 mA
	110 VAC	11.3 W	210 mA	860 mA
	230 VAC	13.1 W	150 mA	670 mA
Standby	24 VDC	4.2 W	220 mA	280 mA
	110 VAC	7.0 W	140 mA	560 mA
	230 VAC	8.2 W	80 mA	460 mA

Table 5: Power consumption LVIS-3MEx

5.5 Touch Panel Cleaning Instructions

5.5.1 Appropriate cleaning measures

Normal dirt can be removed using a clean and soft cloth or chamois leather. When using water, only neutral wetting agents that do not contain any abrasive additives should be added to the water, for example:

- Cleaners that contain surfactants such as Alklar (ECOLAB), Pril (HENKEL)
- Cleaners that contain ammonia such as AJAX glass cleaner (Colgate Palmolive)
- Mild alkaline cleaning agents such as Deconex FPD 111 (BORER)

Stubborn residues (e.g. color or adhesive residues etc.) should be removed using suitable solvents only, according to the following list:

- Ethanol
- Mineral Spirits (e. g. Centralin® household benzine)
- Graffiti cleaners that contain solvents (e. g. GRAFFINET ® cleaner)

5.5.2 Inappropriate cleaning measures

All strong alkaline detergent solutions and acids are unsuitable, especially hydrofluoric acid or cleaning agents that contain fluoride. They attack the anti-reflective glass surfaces and cause irreparable chemical burns. Rough cleaning agents (e.g. abrasives, steel wool, blades, etc.) should not be used either. Refer to the following list of unsuitable cleaning agents:

- Strongly alkaline detergent solutions
- Acids, e.g. hydrofluoric acid
- Detergents that contain fluoride
- Abrasive cleaners and cleaning agents that contain abrasive milk
- Steel wool, scrubbing sponges, or blades
- Fabrics with in-woven metal wires
- Pumice powder

5.6 Terminals and Jumpers

All connectors and configuration jumpers are accessible from the rear of the device. Check the installation instructions that came with your unit. The following is a short summary of the most important terminals:

LVIS-3E100: Select IP-852 or FT-10 channel first. LVIS-ME200: Select BIP or MSTP mode first.	The unit can be operated on an IP-852 (IP-10) channel as well as on an FT-10 or LPT-10 channel (BIP or MSTP for BACnet devices). The jumper labeled FT/IP or BIP/MSTP distinguishes between the 2 modes of operation. Please set Jumper in order to operate the device on an IP-852 or BIP channel (IP mode) and remove the Jumper in order to operate the device on an FT-10 / LPT-10 or MSTP channel.
Connect the Earth Ground wire to the earth ground stud.	Power is connected to the 3 position terminal labeled POWER. Please use only power supplies with the characteristics shown in Table 2. Connect earth ground to the stud labeled with the earth ground symbol. Loosen the top nut and attach the earth ground wire to this location.
FT-10, LPT-10 connection (LVIS-3E100/3E11x)	The FT-10 or LP-10 network is connected to the terminal labeled FT/LPT. The signal wires are connected to terminals A and B, the shield if available is connected to terminal EARTH. This EARTH terminal is internally connected to the main earth terminal.
BACnet MSTP connection (LVIS-ME200/ME21x)	When in MSTP mode, the RS485 bus is connected to the terminal labeled MSTP.
External temp sensor	Up to four external temperature sensors can be connected to the terminal labeled TEMP. Compatible temperature sensors are available from LOYTEC under the part number L-TEMP1. The temperature range of this electronic sensor specified as -55 to +125°C and the accuracy is +/-0.5°Celsius within the range from -10 to +85°C. The resolution of the sensor is 0.0625°C. The sensor device contains an integrated circuit which is connected via a digital interface to the main unit so that no further calibration is needed. Multiple sensor devices are connected in parallel (bus topology). The measured temperature is available in four internal system registers (<i>Sensor 1 Temp.</i> to <i>Sensor 4 Temp.</i>) where the sensors are assigned to the registers by increasing serial numbers of the sensor hardware. The terminal cannot be used for direct connection of passive NTC or PTC elements. DO NOT CONNECT ANY DEVICES OTHER THAN L-TEMP1 SENSORS to this terminal.
External switch inputs	Two external switches (push button or open-close) can be connected between terminals IN0 and GND as well as IN1 and GND on the connector labeled SWITCH. The status of the switches is available on the device in two system registers <i>Switch Input 1</i> and <i>Switch Input 2</i> . On the LVIS-3E100 device, IN0 is also used to control the integrated Lamp Actuator object which is always present as part of the static interface of the device.
TCP/IP connection	In case a CEA-709 unit is connected to an IP-852 channel or a BACnet unit is connected via BACnet/IP, connect the Ethernet cable to the connector labeled 100Base-T. The device also accepts 10Base-T connections via this port. All 7 inch, 12 inch and 15 inch devices have two Ethernet ports, which can be connected internally through a switch or operate in separate networks (WAN/LAN).
CLEAR jumper (LVIS-3E100/ME200)	To remove the currently stored project from the device, set the jumper labeled CLEAR, disconnect power and connect power again. Wait until the unit has started and remove the jumper again.

Table 6: Terminals and Jumpers

5.7 LEDs and Buttons

All LEDs and buttons are either located at the bottom side of the unit enclosure (LVIS-3E100 and LVIS-ME200) or at the top side (12 inch and 15 inch models). Please see the descriptions below for their functionality.

Status button	<p>Pressing the Status button sends out a service pin message on the IP-852 or the FT-10/LPT-10 channel (CEA-709 models) or send out an 'I Am' message (BACnet models).</p> <p>To reset the device to factory defaults, hold the status button pressed while the unit boots and release the button within a few seconds when all interface LEDs light up orange.</p>
Reset button	<p>Behind the small hole next to the status button is the reset button. Use a pin to reach the reset button in order to hard-reset the device. Doing this may cause trend log data or other persistent data to be lost or reverted back to earlier data. To avoid data loss, reset the device via the configuration software or the setup menu (command page).</p>
Power LED	The power LED lights up green as soon as power is connected.
Status LED	The status LED lights up red when the internal persistent storage device is accessed. Also, on CEA-709 models operated on an FT-10/LPT-10 channel, this LED indicates the node status. The LED is off if the node is configured online and flashing red with a period of 1 Hz if the node is in the un-configured state.
ACT 709 LED ACT BAC LED	This LED indicates incoming and outgoing data packets on the currently active communications channel. Only packets which are addressed to the device are shown.
LINK Ethernet LED	The LINK ETH. LED indicates a successful Ethernet link.
ACT Ethernet LED	The ACT ETH LED shows activity on the Ethernet network.
ONLINE LED (LVIS-3E1xx)	The ONLINE LED lights-up green if the node is in the configured online state (CEA-709 models before firmware 4.0.0 only). Starting with firmware 4.0.0, this LED indicates an active connection to the built-in OPC server.
CNIP LED (LVIS-3E1xx)	<p>This LED either shows the current status of the IP-852 interface (in IP-852 mode) or the status of the remote network interface, RNI (in FT-10 mode).</p> <p>IP-852: The LED lights green if the device is properly configured and member of an IP-852 channel. The LED lights orange when the device is configured in a channel but is waiting for updated channel information from the configuration server. In case of errors, the LED lights red.</p> <p>RNI: The LED is dark if RNI is not supported by this device (older devices do not have enough node IDs to support RNI). The LED is green if the remote network interface is ready for connections and orange if the device is currently in use. In case of errors, the LED is red.</p>
MSTP LED (LVIS-ME2xx)	This LED shows the status of the MSTP interface, if the device is operating in BACnet/MSTP mode. The LED is green for normal operation, orange if there is no token and red if there are communication errors.

Table 7: LEDs and Buttons

5.8 Control Network Interfaces

For details regarding the various supported control network interfaces, as well as special topics like redundant Ethernet cabling or WLAN setup, please refer to the Chapter *Network Media* in the LOYTEC Device User Manual.

6 Interface Configuration

6.1 Selecting the Interface

All L-Vis devices provide at least two different physical network interfaces. Depending on the model, number and type of the available interfaces may vary. There are two basic types of communication on these interfaces, which should be clearly distinguished:

- Communication with the configuration software or a building management system like LWEB-900, to read and write configuration data, upgrade the firmware, and retrieve diagnostic data from the device (configuration).
- Communication with other devices on various control networks, to send out new data the user entered through the device and to receive new data which is then displayed (operation).

One of the interfaces which are always available is an IP interface (Ethernet 10/100Base-T). It is normally used for configuration but may also be used for operation, although different protocols will be used for the two tasks, as shown in Table 8.

Depending on the model, there are specialized control network interfaces available, like FT-10 on devices supporting the CEA-709 protocol, BACnet MS/TP for devices supporting BACnet, or Modbus RTU on devices equipped with an RS-485 interface. Only the CEA-709 interface is capable of being used for configuration, although performance will be very low due to the limited bandwidth of the FT-10 interface. It is therefore advisable to use the IP interface for configuration, if possible.

Which interfaces are used for normal operation is normally selected directly on the panel or through the web interface starting with firmware version 4.0.0. For LVIS-3E100 and LVIS-ME200 devices, there is also the option to set the IP jumper. The following table either lists the required jumper setting (SET/OFF) or the interface mode to select on the panel or via the Web-UI.

NOTE:	<i>To activate the BACnet-IP/MSTP router of LVIS-ME200 or LVIS-ME21x devices, the Web-UI must be used to enable both ports and set up BDT, ACL, and Slave Proxy as needed. Do not use the simple port selector on the panel, since it will always enable the selected port only, deactivating the other port and the router. Make sure to configure the correct MSTP network number when enabling the router!</i>
--------------	---

Note that most of the TCP/IP based protocols are active by default, but can be disabled individually via the Web-UI, if needed. While different TCP/IP based protocols can be used at the same time, the RS-485 interface can only run one protocol at a time, which has to be selected via the Web-UI. The FT-10 interface is always used for CEA-709 communication.

Model	Interface	Task	Protocol	Setting
LVIS-3E100	10/100Base-T	Configuration	FTP/HTTP	Via Web-UI
		Operation	OPC XML-DA	Via Web-UI
			Modbus TCP	Via Web-UI
			CEA-852	Jumper SET
	FT-10	Configuration	CEA-709	Jumper OFF
		Operation	CEA-709	Jumper OFF
LVIS-ME200	10/100Base-T	Configuration	FTP/HTTP	Via Web-UI
		Operation	OPC XML-DA	Via Web-UI
			Modbus TCP	Via Web-UI
			BACnet/IP	Jumper SET
	RS-485	Router	BIP / MSTP	Via Web-UI
		Operation	BACnet MS/TP	Jumper OFF
LVIS-3E11x	10/100Base-T	Configuration	FTP/HTTP SSH/HTTPS	Via Web-UI
		Operation	OPC XML-DA	Via Web-UI
			Modbus TCP	Via Web-UI
			CEA-852	IP852 active
	FT-10	Configuration	CEA-709	FT-10 active
		Operation	CEA-709	FT-10 active
LVIS-ME21x	10/100Base-T	Configuration	FTP/HTTP SSH/HTTPS	Via Web-UI
		Operation	OPC XML-DA	Via Web-UI
			Modbus TCP	Via Web-UI
			BACnet/IP	BACnet/IP act.
	RS-485	Router	BIP / MSTP	Via Web-UI
		Operation	BACnet MS/TP	MS/TP active
LVIS-3ME _x	10/100Base-T	Operation	Modbus RTU	Via Web-UI
			CEA-852	IP852 active
			BACnet/IP	BACnet/IP act.
			OPC UA Server	Via Web-UI
			OPC XML-DA	Via Web-UI
			FTP/HTTP SSH/HTTPS	Via Web-UI
	RS-485	Router	BIP / MSTP	Via Web-UI
		Operation	BACnet MS/TP	MS/TP active
	FT-10	Configuration	CEA-709	FT-10 active
		Operation	CEA-709	FT-10 active

Table 8: Available interfaces

6.2 Configuring the IP-852 interface

Before the L-VIS device can be used on an IP-852 channel, the following things have to be configured in the setup menu of the device (see also section 7.3):

1. **CEA-709 Settings:** If not controlled by the IP jumper, select the IP852 interface mode on the CEA-709 settings page. The currently active interface is listed at the bottom of this page and the corresponding button is green, the button of the currently inactive interface is red.
2. **IP Settings:** Configure the IP configuration of the L-VIS device. In the setup menu of the device, configure the IP-Address, Net mask and Standard Gateway. Reset the device when the IP configuration and interface mode selection is finished.
3. **CEA-852 configuration:** After reboot, set the Escrow time, Aggregation time and the MD5 key on the CEA-852/RNI page of the setup menu. The CNIP port setting should normally be set to 1628 and the NAT mode should be set to automatic.
4. **Configuration Server:** Add the L-VIS device to an IP-852 channel. To do this, either enter the IP address of the desired configuration server on the CEA-852/RNI setup page or add the IP address of the L-VIS device to the channel list in a configuration server, e.g. on an L-IP. In order to add the device by entering the configuration server address, the server must have the auto-member option enabled. No additional reboot is required.
5. **Check configuration server:** To check if the L-VIS device was correctly added to the channel, enter the CEA-852/RNI page in the setup menu. The configuration server field should now display the IP address of the configuration server.

7 Operating L-VIS

7.1 Touch Screen

7.1.1 Operation

This section describes how to operate the touch display of the L-VIS device. Most operations should be fairly intuitive, so that users of the device do not need to read a manual before they are able to operate the UI elements of the device, like number, text, bar, or knob controls.

NOTE:	<i>As a project designer, make sure to keep navigation and operation of the UI as simple and intuitive as possible. Avoid hidden functions for important operations, like long press or swipe, which may not be obvious to someone not familiar with the UI.</i>
--------------	--

L-VIS devices use resistive or capacitive touch screens, depending on the model. Devices with aluminum frame use resistive touch screens, devices with glass front use capacitive touch technology. Independent of the type of touch screen, only one touch point at a time is allowed, that is, you cannot press two buttons at once.

The touch screen delivers raw data, which must be converted to LCD pixel positions before it can be used; therefore the touch screen needs calibration. All devices come pre-calibrated from the factory and should be accurate enough to operate the device out of the box. To operate a device using very small control elements or from an unusual viewing angle, it may be necessary for the user to re-calibrate the screen, using the built-in calibration function which is available in the setup menu (section commands). The procedure is explained in more detail in the next section.

The touch pressure is currently not measured and has no effect on the operation.

The touch display is able to detect three different input actions:

- Touch (one point at a time)
- Touch and hold (long press)
- Touch and move

The simple and intuitive **touch** operation is used to select menus, select controls and enter data on the input keypad. This is straight forward. A short touch on a control will also trigger *touch*, *release*, *short press*, and *selection* actions connected to it.

The **touch and hold** operation opens the navigation menu after a configurable time (0 to 5 seconds) when done on a free sport on the display. If the area of a normal input control is touched, the control enters input mode and all further input is processed by the control. If the control has an action object which is set to trigger on a *long press*, the action will be

executed by a touch and hold on the control. Make careful and sparse use of long press actions, as they are not obvious for end users operating the device without instructions.

The **touch and move** operation can be used to adjust values on a bar graph control or move the cursor in a trend control. If executed in a free space on the page, there are three touch gestures which the device can react to:

- **Touch and move up:** This will immediately bring up the navigation menu, so the user does not have to wait for the timeout, as is the case with the touch and hold action.
- **Touch and move right:** This switches to the next page connected to the currently selected menu item. If there is only one page connected, this gesture does nothing (it does not move forward to the next menu item).
- **Touch and move left:** As above, but switches to the previous page connected to the active menu item.

NOTE:	<i>Touch gestures may be disabled in the project configuration, in case the project uses a button-only navigation through the pages. Also note that while gestures may provide a simple and fast way to navigate, they may not be available for LWEB projects running in a web browser and are not obvious for someone not familiar with the UI. Never rely on a swipe gesture as the only way to reach certain screens.</i>
--------------	--

7.1.2 Calibration

In case the touch screen has to be re-calibrated, select the command *Calibrate Touch Screen* from the command dropdown list in the setup menu. This will show a black screen with a white cross at the top-left corner of the display. Touch the middle of the cross using a touch pen or other suitable device. Look at the display from the same viewing angle as the user would during normal operation of the device. Once the touch is detected, the cross moves to the next corner. The procedure repeats until all four corners were measured.

Following the measurements is a verification pass. A white filled circle appears at the same locations as the cross. Again, touch inside the circle to let the device verify the data from the measurement pass. If the data is OK, the screen lights up green, if verification fails, the screen lights up red. The last position for verification is a circle in the middle of the screen, which must be touched as well for the calibration process to complete.

If the verification pass was successful, the new calibration data is stored in non-volatile memory so that it is retained across a power cycle or reboot.

NOTE:	<i>If you are unsure about the viewing angle at which the device will be used, it is best to calibrate for a direct front view, at a right angle to the display surface. This is the most likely position and will also provide enough margins for error in all possible directions. For normal control sizes, the small errors introduced by differing viewing angles are not noticeable.</i>
--------------	--

7.2 Data Input

As shown in the tutorial, L-VIS uses configurable control elements to display and input data. Whether a control is selectable for data input depends on the data points connected to it as well as on the current access level of the user:

- **Data Points:** If at least one output data point is connected to a control, the control becomes an input control, meaning the user may select the control and

input new data, which is then assigned to all connected output data points, unless they are marked as *Constant Value*. If only input data points are connected, the control is not selectable and will only display the most recent value received from any of the connected input data points. Some controls, like trend log controls, may not be used to input new data but may still be selectable, to navigate through the recorded data.

- **Access Level:** If a control is an input control, it will only enter input mode if the current access level of the user is equal to or greater than the level defined for the control on the **Common Properties** page (option 'Access Level for Data Input'). To change the access level, a suitable PIN code must be written to the system register called *Pin Code Enter* (for normal login) or the new access level can be set directly by writing the register *Access Level*.

NOTE:	<i>It is valid to connect input and output data points to the same control at the same time. This is often done when the control should display the current state of a network value (for example the current light level) and also allow the user to modify this value, using the current value as the starting point. The control will ignore any values coming in from the input data points while the user inputs data, so that incoming data does not interfere with the users actions, but resume displaying the incoming values when input mode is left.</i>
--------------	---

If a control is selected, it enters a special input mode, in which all further touch actions are routed to and processed by the control. Depending on the control type, different data input methods are available:

- **Numeric Keypad:** For number controls and bar graph controls, a numeric keypad can be displayed when the input control is selected. New values can then be entered on this keypad, just like on a phone or a calculator.
- **Keyboard:** For text controls without a mapping table, a free text can be entered on a simple keyboard which provides the most important ASCII characters for basic strings (numbers, letters, and most of the special characters found on a US keyboard). Non-Latin scripts are not supported by this simple keyboard. The entered string can be assigned directly to a connected string data point.
- **Touch and move:** For bar graph and knob controls, the bar or knob may be directly moved to the desired position by touching and moving it to the desired position. While doing this, the control will send periodic updates to connected output data points which have the *Immediate* update option set, so the user can get immediate feedback (useful for light control). For details regarding knob control input, see Section 7.2.1.
- **Drop down selection:** For text controls and bitmap controls, a drop down list is displayed when the input control is selected. The user can select the desired item from the list.
- **Push button mode:** For text, bitmap, and vector controls, a push button mode can be activated. This mode assigns a new value to a control whenever the control is selected. The next value is determined by selecting the next or previous entry in the mapping table, so the user may cycle through the available values.
- **Alarm Acknowledge:** The alarm list control may draw a push button next to an alarm entry if the alarm is to be acknowledged. Pressing the button will send the acknowledgement to the alarm server and change the color of the button. Once the server received the acknowledgment, the button or the entire alarm entry may disappear from the list.

- **Schedule Configuration:** Due to the complex nature of schedule configuration data, the schedule control implements a specialized input method which is explained in detail in section 7.2.1.
- **Actions:** There is an indirect way of assigning values to output data points using the *Update Data Points* action. This is sometimes useful, for example to store a current value in a register data point when the user selects or touches a control. These actions are explained in more detail in section 9.9.

NOTE:

Input events are never duplicated and sent to more than one control at a time. Similar to a desktop window system, every event is only routed to the most suitable control for further processing. Controls on a page are searched from top to bottom. The first control accepting input will get the event. It can either use it, or ignore it and return it to the system for further routing. Controls which have touch-controlled actions connected to them will use the input events to trigger their actions instead of entering input mode and using the touch events for normal value input. This means that it is not possible to build a number control which opens a keypad for value input and at the same time triggers an on touch action. However, it is possible to connect more than one action to the same control and have the control trigger them all using the same input event.

The control decides when to leave input mode. This is also called losing the *Input Focus*, since the control returns input processing back to the page which is currently shown. Upon input focus loss, the control usually sends out the final value to any connected output data points and the page flip timer is restarted (if configured for the current page).

Controls usually return the focus when you touch an area outside the control or when you leave the device idle for a specified amount of time. Some input methods (like the keypad) have OK and Esc buttons for you to press, to either close and accept the new value or close and restore the previous value.

Leaving a control idle while in input mode will usually be interpreted by the control as a *cancel* action. The control will then leave input mode, return the focus to the page and restore the value it had when input mode was entered. The idle timeout can be set in the **Project Settings** dialog, together with a number of other timeout values.

7.2.1 Knob Control

Knob controls accepting input can be rotated by touch and move, similar to bar controls. While this should be intuitive and easy to use for anyone operating the touch panel, there are some details which are worth mentioning:

- **Entering input mode:** The control will enter input mode whenever a touch is detected inside a circular area including the button and any tick marks or labels around it, as configured on the property page of the knob control. Touches outside this circular area will not be consumed by the control and will therefore be routed to other controls below the knob, and finally to the page.
- **Starting input:** Once the knob entered input mode, it will wait for a touch position which is far enough from the knob center such that a well-defined knob rotation can be calculated. In case input mode was started by a touch close to the knob center, the knob will keep monitoring the touch position until the desired knob rotation can be determined. This will start actual input and the knob will follow the current touch position from this point on. Depending on the mode of input, the knob may initially turn to the exact angle determined by the current touch position and stay aligned with it, or remain at its current angle and move only relative to the movement of the touch position, to avoid sudden value changes. This happens when the option *Input by knob rotation only* is set.
- **Turning the knob:** Once the control actually started input mode, it will constantly sample the touch position and calculate a new knob rotation from it.

The knob can now be turned by moving the touch position around, including the knob center and any area outside the knob control. During this phase, the entire screen becomes the active area, such that the user may move the touch position well outside the knob. This helps to gain finer control of the knob rotation, especially for small knobs.

- **Over-Rotation:** In general, the knob cannot be rotated beyond its defined min/max positions. It will stop at the range end and not follow the current touch position anymore, until the touch position is moved back into valid range. However, when the user over-rotates the knob such that the touch position is moved beyond max and back to min (coming from max), it would lead to a sudden rotation change from max to min. To avoid this effect, the valid range is reduced while the button is at the range end, such that over-rotation of up to 50% of the full knob range is allowed without causing a sudden position change.
- **Snap into position:** Once the user releases the knob, it will either remain at its current position, or snap into place at the nearest tick mark before a final update with the current position is sent out to all connected output data points.

7.2.2 Simple Schedule Control

This section describes how to operate a schedule control which is set to simple view (extended UI mode not enabled). A schedule control displays the configuration of one or more local or remote scheduler units and their respective calendars. In the simple UI mode, the control consists of three basic areas:

- **Header:** The header displays the name and the effective period of the selected scheduler. If there is more than one scheduler configuration available (more than one scheduler data point is connected to the control), a small triangle is drawn next to the schedule name and touching the name opens a dropdown list of all available schedules.
- **Day List:** The left side of the control contains a list of days for which schedules are available. There is one entry for each day of the week, followed by the list of all exception days defined by the corresponding calendar.
- **Time Table:** The right side of the control shows the time table for the selected day. The time table starts with its own header line, which shows the selected day and also acts as a dropdown list of commands available for this day.

The various operations which can be performed on a schedule control in this mode are explained in more detail in the following sections. Many of the operations may be restricted to a certain access level by the project designer, so it may be required to enter a PIN code before all of the operations are available. For more information on access control setup for schedule controls refer to section 9.5.9.

NOTE:	<i>The update mode used by schedule controls is fixed to Focus Loss, meaning that the new schedule configuration will be written to the scheduler only at the time the control leaves edit mode. This avoids intermediate updates while the user is still modifying the schedule, which could result in temporarily inconsistent configurations.</i>
--------------	--

7.2.2.1 Select a Schedule

If more than one schedule data point is connected to the control, a small triangle is drawn to the left of the schedule name and touching the schedule name will open a dropdown list of all available schedules for selection. This action is not restricted by access level settings, so the dropdown box is always available to select one of the connected schedules.

7.2.2.2 Select a Day

Touching one of the day names in the left column will show the corresponding time table for that day in the right column. If the day is inactive, its name will be drawn in the same color as the background of the header lines instead of the normal text color. Selecting a day for display is not restricted by access level settings.

7.2.2.3 Set Exception Day Priority

If the day is an exception day, a number is displayed next to the day name. This number defines the priority of the exception schedule relative to other exception schedules which may become active on the same day. To set the priority, touch the number and enter the desired value on the keypad. This operation may be restricted by access level, in which case the control will not react to a touch of the priority field.

7.2.2.4 Change Time of Events

To change the time at which a certain value is scheduled, touch the hour or minute number in the time table and enter the desired value. The hour is entered in 24 hour format (0-23) or can be used to enter hour and minute in one operation (HHMM). For example touching the hour and entering 1254 will set the time to 12:54.

NOTE:	<i>Entries in the time table are always displayed in chronological order. If this order changes as a result of modifying the time of an entry, the entry will immediately move to its new position in the time table to maintain the correct order of events.</i>
--------------	---

This operation may be restricted by access level, in which case the control will not react to a touch of the hour or minute fields.

7.2.2.5 Select Value Preset

To select a different value for an entry in the time table, touch the value name to open a dropdown box. This box contains a list of available values, followed by some commands which will be explained later. Select the desired value from the list. This operation may be restricted by access level, in which case the control will not react to a touch of the value name.

7.2.2.6 Set Effective Period

If shown in the header line, the effective period of the schedule may be changed by touching the individual elements of the start and end date and setting new values for day, month and year using the keypad. To clear the start or end date (no restriction), select any of the fields (day, month, or year), press the Clr button in the keypad to delete the value and press OK to close the keypad. This operation may be restricted by access level, in which case the control will not react to a touch of the effective period.

7.2.2.7 Edit Values of a Preset

To change the physical values assigned to a value preset, touch the preset name to open the dropdown box. From the box, select the command *edit* to open a small window showing the current values of the preset. If the value preset consists of multiple individual values, each value is shown on a separate line with the value name to the left and the current value to the right. Touch the value to open the keypad and enter a new value. When done, close the window by touching the close symbol in the top right corner of the window.

NOTE: *Changing the values of a value preset will automatically affect all entries in all time tables which reference this value set. If for example the fan speed assigned to the value preset NORMAL is changed from 60% to 75%, all time table entries which set the output to NORMAL will use the new speed setting.*

This operation may be restricted by access level, in which case the edit command in the dropdown box is not selectable.

7.2.2.8 Define new Presets

If the scheduler controls only a single value, it is possible to create new value templates through the schedule control. Touch the preset name on the time table entry for which you want a new preset and select the command *value* from the dropdown box. Enter the desired value for the new preset using the keypad. A new preset will be created for the given value and assigned to the entry on which the operation was executed.

NOTE: *Do not confuse this operation with the operation of changing the values assigned to an existing value preset. This operation will actually create a new, independent value preset and assign it to the selected time table entry, not affecting any other entries. Only if a preset for the requested value already exists, the existing preset is used.*

This operation may be restricted by access level, in which case the value command in the dropdown box is not selectable.

7.2.2.9 Add and Remove Entries

To add a new entry to a time table, select the *insert* command on the last line (after the last entry in the list). A new entry will appear with the default time 23:00 and the first available value preset. Now edit the new entry as described earlier (change the time and the value preset). This operation may be restricted by access level, in which case the control does not react to a touch of the insert line.

NOTE: *The scheduler unit may report a restricted capacity for entries in the time table. If this limit is reached, the control will not allow any further insert operations.*

To remove an entry from the time table, touch the value preset name of the entry which is to be removed and select the *remove* command from the dropdown box. If this operation is restricted, the remove command will not be selectable.

7.2.2.10 Activate/Deactivate Days

To activate or deactivate individual days, select the desired day from the left column and touch the headline of the time table to open a dropdown menu. From this menu, select *activate* or *deactivate* (only one of the two commands will be shown). If this operation is restricted, the time table menu will not open.

7.2.2.11 Clear or Copy Time Tables

To clear all entries in a time table or copy them to a different day, select the day to clear or to copy from in the left column and touch the headline of the time table to open a dropdown menu. From this menu, use the *clear* or the various *copy* commands to perform the desired operations. If this operation is restricted, the time table menu will not open.

7.2.2.12 Edit Exception Days

To change the days on which a certain exception schedule is active, select the exception day in the left column and touch the headline of the time table to open a dropdown menu. From this menu, select the *edit* command to open a new window in which the definition of the active days may be modified. The window has two columns to show the type of entry and the data. The following entry types are possible.

- **Single Date:** This type of entry defines a single date pattern consisting of a day, month and year specification. Each of them may be specified or left unset (wildcard), so that this type of entry is suitable to define anything from a single day of a specific year (all parts given) up to every day of every month in every year (all parts set to wildcard).
- **Date Range:** This type of entry defines a number of successive days given a start date and an end date. Depending on the use of wildcards for the year or the month, the range may apply only once, once a year or every month.
- **Week and Day:** This type of entry defines a specific day in a specific week of a specific month, for example every second Friday in April. Again, using wildcards for the individual components allows specifying more than one specific day of the year (for example the first Monday of every month or any Monday in April).

To insert a new entry, either touch the *insert* line in the left column (append at the end of the list) or touch the entry at the position where the new entry should be inserted (the selected entry and all following entries will be moved down).

NOTE:	<i>Insert may not be possible if the calendar object reports that the limit of entries is reached or if the calendar does not allow insertion of new entries at all (for example embedded BACnet calendar objects may be fixed to exactly one entry).</i>
--------------	---

The new entry will be a single date entry. If the type should be changed, touch the type and select the desired type from the dropdown menu. To remove the entry, select the *remove* command from the same menu.

To edit the data of the entry, touch the individual components of the dates to open up a dropdown list with various choices. One of the choices will always be *value* to open a keypad and enter a value. Other choices may be available depending on the selected element. For most entries, the choice *any* will be available to set this component to the wildcard. For months, the choices *odd* and *even* may be available to specify all odd or all even months. For days, the option *last* may be available to specify the last day of the month.

To save the changes done to the exception day patterns, close the window using the close button in the top right corner of the window. This will update the calendar object with the new data. Note that this will also affect all other schedules running on the same calendar.

7.2.3 Extended Schedule Control

This section describes how to operate a schedule control which is set to extended UI mode, which is available on devices running firmware 4.2.0 or newer. In this mode, the control displays a calendar and event based preview of the schedule instead of a list of daily schedules. There are a number of different pages between which the user can navigate:

- **Weekly Overview:** This is the default page, which is displayed after a system start. It shows an overview of the effective schedule for one week, organized in seven columns, one for each day of the week.
- **Time Table:** This page shows the effective time table for a selected day. Starting at midnight, it lists the presets which will be scheduled. Each line consists of the following information: The time at which the preset becomes active, the time until this preset stays active, and the name of the scheduled preset.
- **Event List:** This page shows a list of all events applicable to a selected day. Note that this list does not exactly represent the presets that will be in effect on that day. Events may overlap and it depends on the event priority which preset will actually be scheduled.

- **Global Event List:** This page shows a listing of all events currently stored in the scheduler unit. It is similar to the event list, but it is not filtered by a certain date.

Each of the pages consists of a header line with navigation buttons and a heading, followed by the contents of the page. How these individual pages are used, is explained in the next sections.

7.2.3.1 Weekly Overview

This page displays one week worth of schedule preview on one page. The following areas of the page can be selected by the user to execute certain actions:

- **Single and double arrow buttons:** These buttons are used to navigate to the preview of a different week. The single arrow buttons switch to the next or previous week, whereas the double arrow buttons move by four weeks.
- **Plus Button:** This button is used to add a new event to the scheduler unit. A new dialog window appears which allows the user to define the desired event. This dialog is explained later on.
- **Pen Button:** This button opens the global event list.
- **Page Title:** Touching the page title will switch the preview to the current week.
- **Day Columns:** Touching one of the seven day columns can have different effects, depending on the touch position and type:
 - **Touch on a scheduled value:** Opens the event editor for the event which corresponds to the selected value. Note that multiple areas on the same day or on different days may correspond to the same event.
 - **Short touch on a free area:** Switches to a detailed view of that day. Depending on the configuration of the scheduler control, this can be either the time table, or the event list.
 - **Long touch on a free area:** Opens the event editor to add a new event at the selected time and day (depends on the touch position).
- **Day Column Headings:** Touching the heading of a certain day column switches to a detailed view of that day. Depending on the configuration of the control, this will be the event list or the time table. In any case, it will be the opposite from the page that is shown by a touch of the day column.

The schedule control can be configured to switch back to this page and show the preview of the current week, whenever the device enters idle mode (the display is dimmed after the user did not use the device for a while).

7.2.3.2 Time Table

This page displays a list of scheduled preset values for the selected day. The page header contains buttons to navigate back and forth by one day (single arrow) or one week (double arrow). The pen button opens the event list for the selected day, the button with the arrow pointing to the top left goes back to the weekly overview.

The information shown on this page represents the sequence of preset values which will be scheduled on that day as a result of individual events that are applicable. Each line contains a start time, an end time, and the name of the value preset which will be in effect during this period of the day. The list covers the entire day, starting at midnight.

7.2.3.3 Event List

This page displays a list of all events which are applicable to the selected day. Note that an event which is generally applicable to a certain day may not yield to a new value being scheduled, since the event may partially or fully overlap with other, higher priority events. Partially overlapped events may cause the start and/or end times of the actual time table to be different from the start and end times of the individual events.

The event list is normally used to modify, add, or delete events. The following areas of the page can be selected to execute various operations:

- **Single and double arrow buttons:** These buttons can be used to navigate back and forth by one day (single arrow) or one week (double arrow) without leaving the event list page.
- **Plus Button:** This button is used to add a new event. The event editor will be opened and the new event can be entered. Note that the currently selected day has no influence on the new event, since all data for the new event can be modified in the event editor window.
- **Back Button:** The button with the arrow pointing to the top left is used to go back to the previous page. This can be the time table or the weekly overview, depending on how the event list was reached.
- **Lines in the event list:** Each line of the event list may be selected anywhere on the line (not just the edit buttons), to open the event editor for the selected event. This way, existing events can be modified or deleted.

Each entry in the event list can show the following information:

- **Start and End Time:** The start and end time of the event.
- **Preset Name:** The name of the value preset which should be scheduled.
- **Priority:** Optionally, the priority of the event (depending on available space and the setup of the scheduler control).
- **Date Specification:** The dates on which this event will be active, as well as a name for this event, if available (not supported by BACnet at the moment).

Events are sorted by their start time. In case of overlapping events, the actual sequence of scheduled values may be different from the sequence in which the events are displayed.

NOTE:	<i>In case of overlapping events scheduled at the same priority but with different values, it cannot be guaranteed which of the two conflicting values will be scheduled.</i>
--------------	---

7.2.3.4 Global Event List

This page is almost the same as the normal event list. The only difference is that this page shows all events which are currently stored in the scheduler unit. Since this list is not related to a specific date, there are no navigation buttons. Only the buttons to add a new event and to navigate back to the previous page are available. The page can be reached directly from the weekly overview by pressing the edit button.

7.2.3.5 Event Editor

To modify existing events and add new events, a separate editor window is opened in front of the scheduler. The following settings can be adjusted in this editor:

- **Name:** A custom name for this event. If supported by the underlying scheduler unit, a name can be entered here. Note that even if this field is available, a length limit for the name may apply.
- **Value:** The value which should be scheduled when this event is active. Depending on the type of data points controlled by the scheduler unit, this field may be a dropdown list of all the available value presets or a field to enter an arbitrary number via the keypad. If supported by the scheduler, an additional edit button will be displayed to the right of the value field, to open the value template editor for modifying or adding/removing value templates.
- **Priority:** The priority of this event. Depending on the configuration of the schedule control, this can be a numeric priority value as used by the native implementation of schedules in this system, for example 0 to 15 for BACnet or 0 to 127 in case of CEA709 networks, or a dropdown list of discrete priority levels (low/normal/high).
- **Start:** The time of day at which the given value should become active.
- **End:** The time of day at which the value should be taken back. The value of the next lower event applicable at this time will apply next. If no other event applies, it depends on the underlying network technology what the next output value will be. To guarantee a well-defined behavior independent of network technology, a default event should be defined, which applies to all days and schedules a desired default value at the lowest priority (see note below).
- **Duration:** The time duration for which this event will be active. This is calculated from the start and end times. If the duration is set via this field, the end time of the event will be adjusted to match the desired total duration. If this would move the end time past midnight, the desired duration is archived by additionally shifting the start time forward.
- **Event Type:** The type of event defines how often it will be repeated. There are a number of different event types to choose from, for example one-time events, events that are active daily within a defined start and end date, events that are scheduled on a weekly, monthly, or yearly basis, events which are scheduled whenever a certain global calendar event applies, or events which are defined by special date patterns.
- **Event Details:** Depending on the event type selected, different input fields are shown here to enter the specific information required to define when exactly the event should apply.

In the bottom area of the event details are some push buttons to cancel the editor without making any changes, to close and save the changes or add the new event, as well as to delete the event currently open in the editor.

NOTE:	<i>If applicable, a default button will be shown. This button can be used to easily set the correct definitions for a default background event, which is scheduled at the lowest priority on every day. It is normally used to control the outputs of the scheduler whenever no other event applies. There can only be one default event per schedule. If present, the default event will always be listed as the first event in the event list and it will be marked as default.</i>
--------------	---

7.2.3.6 Value Template Editor

If the scheduler unit uses value templates and allows the user to change them, the event editor will display an edit button next to the value template field. Pressing this button will open the value template editor, where a list of all value templates is shown. Currently unused templates show a small X button via which the template may be deleted. At the end of the list, a new value template may be added. Touching the name of the template allows

the user to enter a new name via the keyboard. To close the template editor, press the close button in the top right corner of the window.

7.2.3.7 Calendar Pattern Editor

If an event references a global calendar pattern and the calendar allows the user to edit the pattern definition, an edit button will be shown next to the dropdown list from which a calendar pattern is selected. Pressing this button will open the calendar pattern editor. This is the same editor which is also used by the simple view of the scheduler control.

If the calendar allows adding of new calendar patterns, an additional + button will be displayed. Pressing this button will add a new calendar pattern, which can then be edited via the edit button.

To change the name of a calendar pattern, select the headline of the editor window, where the current name of the calendar pattern is displayed.

7.2.3.8 Technical Notes

When using the extended schedule control, data input is no longer based on different daily schedules and calendar patterns. Instead, the user defines individual events, which are independent of each other. Since each of these events may have its own name and priority and may overlap with other events, it must be implemented in the underlying scheduler unit by using so called exception schedules.

Each event will be stored as an independent exception schedule, whereas the fixed weekly schedules (one for each day of the week) are not used in most cases. However, if data is found in the weekly schedules of a device, it will be visible in the schedule control as weekly events at priority 255 or “Weekly” (when the simplified priority scheme is used).

Note that some devices may lack support for exception schedules or the number of possible exception schedules may be limited. In such cases, enter your events as normal weekly events using the “weekly” priority (255). These events will be stored in the weekly schedule of the device and will not allocate an exception schedule. However, since all of the weekly events have the same priority, they are not independent of each other. If they overlap on the same day, the data will be merged and some events may have their start or end time modified to resolve the overlap.

Alternatively, the simple UI mode can be used, or the user needs to remove older events which are no longer needed, to free some capacity of the device for adding new events.

NOTE:	<i>To make sure that the capacity for events in local CEA709 schedules is high enough, set the maximum number of daily schedules in the network settings dialog to a high value when using the extended UI mode. This number defines the maximum number of events which can be stored in the CEA709 LonMark scheduler unit, since each event will consume one entry in the table of daily schedules.</i>
--------------	---

Also note that a default event is implemented as an exception schedule as well, and that it will apply for all days. Since the weekly schedules have a fixed priority which is lower than the lowest priority of any exception schedule, such a default event will immediately mask out all data which might still be present in weekly schedules. If this is the case, the user will see the available weekly schedule data at priority 255, but since they will not apply, there will be no value names shown for these events until the user converts them to events using exception schedules. Therefore, do not use a default event when you are still using weekly schedules.

7.3 Setup Menu

All devices come with a built-in setup menu, which is used to set device specific parameters, such as the devices IP address, control network settings, or date and time zone. These settings are related to the installation of the device and are usually configured at installation time.

Once the device is installed and a project is downloaded to the device, the built-in setup menu may be hidden, so that it is no longer accessible to the users of the device. Alternatively, access to the setup pages may be protected by a PIN code. Both options are available in the **Project Settings** dialog, which can be accessed from the **File** menu in the configuration software.

7.3.1 Common Pages

The standard setup menu always contains the following pages:

- **TCP/IP:** On this page, the IP address, network mask, gateway, and optionally NTP server addresses for time synchronization can be configured. Alternatively, the device can be instructed to get the required settings from a DHCP server in the network (automatic IP configuration). Changes on this page will not be effective until the device is restarted.
- **Date/Time:** This page is used to set date and time, as well as the current time zone. Always adjust the time zone first and then set date and time to the local time. There are various ways to synchronize the time with external time sources. The current time source is defined by the loaded project and is displayed at the bottom of the page. If an external time source is used, only the time zone needs to be adjusted here.
- **Commands:** This page contains a dropdown list of the commands which can be executed. Some commands are available on all devices, while others are specific to a device family. Also on this page is a selector to change the effective unit system. The following commands are defined:

Command	Description
Send Service Pin Message	CEA-709 models only. Send out a service pin message.
Send I-Am Message	BACnet models only. Send out an I-Am message.
Un-configure Node	CEA-709 models only. Set the device back to un-configured state (forget network address).
Clear System Log	Clear the system log of all entries.
Lock Pages (Logout)	Consider all protected pages to be locked again, after the PIN code was entered and the pages were unlocked before. This basically sets the current access level back to zero.
Calibrate Touch Screen	Start the touch calibration routine.
Send Test E-Mail	Sends a short mail message to the recipient specified as the test receiver in the project settings.
Clear Data and Reset	Clear all persistent data (user data points forget their stored value).
Reset Device	Perform a clean shutdown and reboot. All persistent data will be saved before the device restarts.
Standby	Puts the device into standby mode.
Disable HTTP Server	Disables the web server on the device.
Enable HTTP Server	Enables the web server on the device (on by default).

- **Storage:** This page is available on devices which support external data storage like USB and SD card memory. The buttons on this page can be used to save and restore device backups, L-Vis project files, and to write current trend log data to the external memory. See section 15.9 for details.
- **Info:** This page contains some basic status information of the device, to check CPU load and temperature, available memory, and also the status of externally connected switches and temperature sensors (verify device installation).
- **About:** Show the about page (default title page).
- **Exit:** Exit the setup menu and return to the main menu.

7.3.2 CEA-709 Models

Devices supporting CEA-709 networks provide the following additional pages:

- **CEA-709:** On this page, the device can be given a CEA-709 address (domain/subnet/node) and can be set online. This is useful to connect to the device in standalone mode, before the device is actually commissioned in a network, for example because the project needs to be downloaded first to define the static interface of the device and there is no TCP/IP connectivity available.
- **CEA-852/RNI:** On this page, the IP-852 parameters or RNI parameters (depending on the mode of operation) can be specified. Supported parameters are the escrow and aggregation timeout, the IP-852 port number, the MD5 key, and the NAT support mode. Please refer to the documentation of your configuration server for more information about these parameters. The defaults should normally be OK, unless your channel uses MD5 encryption, in which case you need to enter the MD5 key and activate it by selecting *Enable with new key* -> from the dropdown list after the key was entered. Starting with firmware 4.0.2, it is also possible to enter the address of the configuration server on this page.

NOTE:	<i>The key is displayed as 8 groups of 4-digit hex values, which is a common representation of MD5 key values. The order to read the key is from left to right and top to bottom.</i>
--------------	---

To add the device to an IP852 channel, you can either enter the address of the desired configuration server on the device or add the devices IP address to the channel on your configuration server, which will then contact the device and set the channel-wide parameters such as the time server addresses automatically. If the configuration server address is entered on the device, the server should have the auto-member option turned on, so that the device may add itself to the channel automatically. In any case, the result is displayed at the bottom of the CEA-852 setup page, where the IP address of the current configuration server is displayed.

NOTE:	<i>For security reasons, enabling MD5 with a new key will store and activate the key and then reset the input fields to all 0000, so that the current key cannot be seen on the display. Once the key is stored that way, it may be enabled and disabled using the dropdown list, even though the input fields always show 0000. Consider these fields to be used for input of a new key only.</i>
--------------	---

7.3.3 BACnet Models

Devices supporting BACnet networks provide the following additional page:

- **BACnet:** On this page, the BACnet device ID, MSTP node address and baud rate (for MSTP mode only) and the BACnet/IP port number (for BACnet/IP mode only) can be set. The currently active interface is shown and can be selected using the buttons on the right side of the screen. The currently active interface is shown in green, the other one in red.

7.3.4 Custom Setup Menu

Starting with version 2.0 of the device firmware, the project may use standard controls on standard project pages to set all system parameters, like IP address, date, time zone and so on. To do this, create a suitable control and connect the system parameter which it should modify. This way, a project may provide a completely customized setup menu, matching the design of the other pages in the project, for example a localized Japanese setup menu may be provided, or a subset of the complete setup menu, such as specific commands from the command page, or the setup of date and time only.

NOTE:	<i>The example projects that are installed together with the configuration software include projects for the standard setup menu of the different LVIS models. These projects should provide a good starting point for a customized setup menu.</i>
--------------	---

8 Configuration Software

8.1 Main Window

The main window of the configuration software is shown in Figure 2. It is divided up into 3 main areas, as explained in section 3.3.3 of the tutorial.

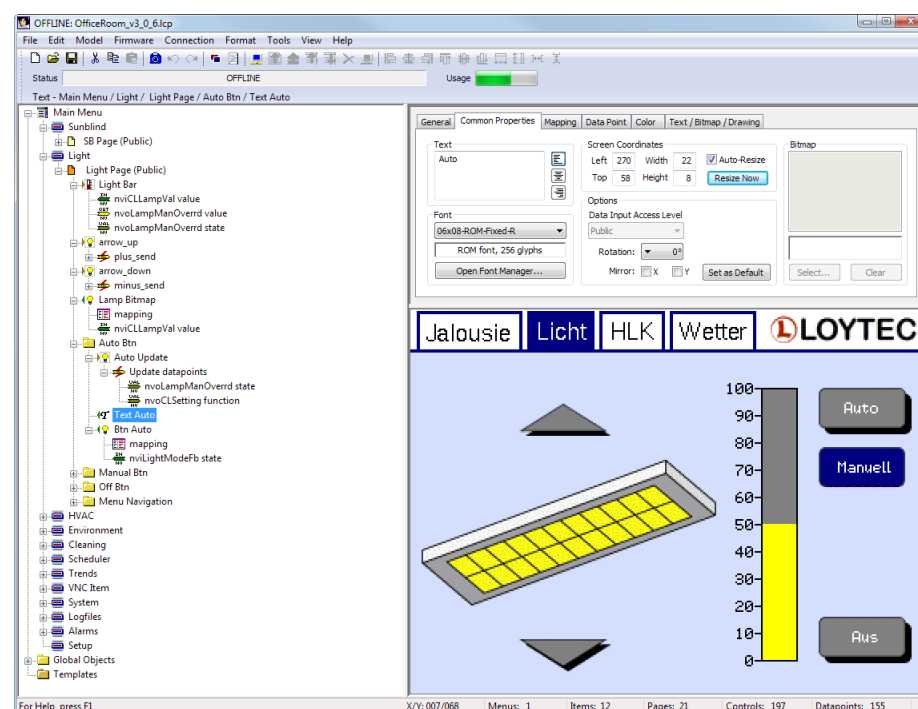


Figure 2: Main window of the L-VIS configuration software

The following sections describe the object tree and the LCD preview in more detail.

8.2 Object Tree

The object tree shows all objects which are part of the L-Vis project. The top level contains three nodes. A root menu, a folder for objects which are not visible and not related to any of the menu items or pages below the root menu, and a folder to organize template pages which are then instantiated on pages below the root menu.

The following operations are possible in the object tree:

- **Left click an object:** The object is selected. All previously selected objects are deselected. If the selected object is a child object of a control, the page on which the control is located will be shown and the control will be marked in the LCD preview.
- **Right click an object:** A context menu with commands related to the object is shown. This is used to add new child objects, cut, copy, paste, lock, hide, or execute other operations on objects in the tree.
- **Left button down and move:** When done on a selected object, all currently selected objects will be moved and can be dropped at a different location in the object tree (drag & drop). When done on an unselected object, the object will be selected and moved.
- **Cursor up/down:** Move the selection up and down.
- **Cursor left:** Fold the selected object, in case it is currently expanded.
- **Cursor right:** Expand the current object, in case it is currently folded.
- **Mouse wheel:** Scroll the tree view up and down.
- **Shift + left click:** Selects all controls on the current level between the currently selected object and the object which is clicked on while shift is pressed. Start and end of the selection range must be on the same hierarchy level.
- **Ctrl + left click:** Changes the selection status of the clicked object without affecting selection of other objects. If the clicked object was already selected, it will be deselected, and vice versa. This is used to select multiple objects. **Note:** It is not valid to select a child object of an already selected object. Whenever an object is added to the selection, all its child objects will automatically be deselected and child objects of an already selected object cannot be selected.
- **Shift + cursor keys:** Moves the selection to the next or previous object but does not change the selection status of other objects. This is a way to select a range of objects in the tree using the keyboard only.
- **Enter:** Changes keyboard focus to the LCD preview. The selected object can now be manipulated in the LCD preview.

8.3 LCD Preview

The LCD preview shows the selected menu or the page on which the selected object is located. In case of multi-selection across different pages, the page of the last selected object is shown.

In a multi-language project, the preview always shows the currently active language, which can be selected either from the *View / Language* menu, or the **Editor** property page. Selecting a different language immediately updates the preview.

The preview can also be used to select elements on the page and change their size or position. The following sections explain the LCD preview in more detail.

8.3.1 Selection Candidate

Moving the mouse cursor around in the preview window will indicate if a selectable object is currently located at the position of the mouse cursor. This is done by showing a small dotted frame around the selectable object.



In many cases, there will be a number of different selectable objects at the same position, so one of them has to be picked as the primary selection candidate. This is done by the following rules:

- If there are any controls underneath the cursor, the topmost control will be selected. This is the control which is located highest up in the object tree and which appears in front of any other overlapping controls on the page. Controls which are locked or which are part of a template instance are skipped during this evaluation and will therefore never be chosen as the primary selection.
- If there are no selectable controls below the cursor, but the cursor is within the bounding box of one or more collections, the inner-most collection is selected. This is the smallest possible collection, which is located the farthest down in the hierarchy of nested collections. Similar to controls, collections which are locked or which are part of a template instance are skipped and are never chosen as the primary selection candidate.

8.3.2 Object Selection

If a selection candidate is highlighted, it can be selected by a single click of the left mouse button. To select one of the other possible objects, perform a double click of the left button at the same position. This will move the selection to the next possible object. Repeat until the desired object is selected. As an alternative, it is of course always possible to select the desired object from the tree view.

NOTE: Locked objects are never considered for selection in the LCD preview. On pages with a lot of overlapping objects, it can help a lot when objects which are currently not of interest are locked into position via their context menu. This will get them out of the way during object selection and prevent accidental move or resize.

8.3.2.1 Multi-Select

Similar to the tree view, the **shift** key can be used to select multiple objects in the LCD preview. While the shift key is pressed, selecting an unselected object will not deselect any of the already selected objects. When clicking on an already selected object while shift is held down, the object will be removed from the selection.

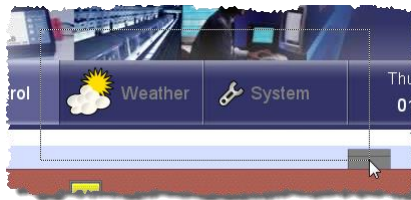
NOTE: It is valid to select some objects in the LCD preview and others in the tree view. Any change to object selection in one of the two views will be visible in the other view as well. Note that if multiple objects from different pages are selected, the tree view must be used to select the first object on a new page. The LCD view will indicate the location of all selected objects, even those which are located on other pages and are not visible at the moment.

Release the shift key to return to normal operation and allow for example the selected group of objects to be moved or resized on the screen. While the shift key is held down, object selection takes precedence over move or resizes operations.

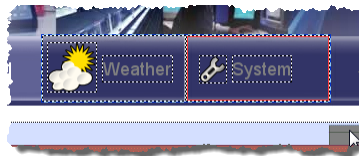
8.3.2.2 Selection Box

A selection box is an expandable temporary box used to select multiple objects which are located near each other on the page but not necessarily within the same collection. This is useful to perform an operation like move, lock, hide, cut, copy, or similar, on all objects within a certain area of the page. To select multiple objects via the selection box, follow the steps below:

- Click and hold the mouse button a short distance away from the objects you want to select. This will start a new selection box if there are no selectable objects at this location. In case there is a possible selection candidate at the desired starting point of the box, hold down the key **s** to disable normal selection and instead force a selection box to be started.
- Drag the mouse to expand the selection box over the objects you want to select. There are two different types of selection box:
 1. Clicking to the right-side and dragging to the left selects any elements within the selection box, including those that are only partially contained in the box.
 2. Clicking to the left-side of the objects and dragging right selects only those elements completely within the selection rectangle.



- Release the mouse button when all of the elements are either partially included (right-to-left selection) or fully included (left-to-right selection).



NOTE: Selection boxes can also be used in multi-select mode to change the selection of a group of objects in one step. Follow the same steps as above, but hold down the **shift** key. If a selection candidate is at the desired starting point of the multi-select box, hold down the **s** key in addition to shift, to override the selection candidate and spawn the multi-select box.

8.3.3 Selection Indicator

The LCD preview indicates each individual selected object by drawing a wide dotted frame around its outer boundaries. This frame is black and white, like the selection candidate frame, but its dot pattern is wider to distinguish the two frames from each other.



In case there are multiple selected objects, an additional frame around the entire group of objects is drawn. This frame has an even wider dot pattern and is light and dark blue instead of black and white. When the mouse cursor is anywhere within that group frame, the cursor shape will change into a move cursor and the group of selected objects can be moved. A right click on the group frame will show the multi-select context menu with commands to lock, unlock, show, hide, cut, or copy the selected objects.



NOTE: While it is possible to lock objects via the context menu in the LCD preview, it will be necessary to unlock them in the tree view, since they will no longer be selectable in the LCD view once they are locked.

8.3.4 Resize Frame

When activated via the tool bar or the edit menu (see Section 8.6.1), the LCD preview can indicate the active control area with a red frame.



This frame marks the position and size of the control as specified in the common properties of the control. It usually excludes any decorations around the control, like scales or scale labels. Since the control can only be resized at the border of this frame, it can help to see where it is for a particular control.

The indicator will only be shown for the last selected control in case of a multi-selection.

8.3.5 Guides

When an object is selected, the preview will show wide-dotted green lines in horizontal and vertical direction to help in alignment with other objects on the screen.



The guides are usually extending from the four corners of the selection indicator, but depending on the selected object, other guides may be shown. For example, most controls show guides along the edges of their container box, which is the bar of a bar control or the graph area of a trend control. Controls which contain lines of text, like text, number, or date controls, will show guides indicating the baselines of each line of text.

8.3.6 Snap to Edges

When activated via the tool bar or the edit menu (see Section 8.6.1), the guides shown for the selected object will snap to nearby edges of other objects to help in exact alignment. The effect is not very strong, but enough to correct a 2 or 3 pixel offset.

When the **shift** key is pressed and held down while moving, snap to edges is temporarily disabled. Note that the move has to be started first without holding shift. Otherwise the preview will be in multi-select mode and will not allow object move.

8.3.6.1 Adaptive Snap Strength

The snap effect will be stronger at positions where many already aligned edges are located, or where there are long edges, compared to areas with fewer edges. All edges of the moving object are evaluated individually for their best position. The edge with the highest score is then selected to control the snap.

This prevents the algorithm from being thrown off by a large number of objects on a page. It makes sure that areas where elements are already well aligned are automatically preferred for placement of a new object.

8.3.6.2 Back to Back Placement

The snap algorithm distinguishes between left and right, as well as top and bottom edges, to allow for back to back placement of similar objects without causing a one pixel overlap. This is archived by the following rules:

- Same edges align on top of each other. The left edge of the moving object aligns at the same pixel location as the left edge of a nearby object.
- Opposite edges align with one pixel offset. The left edge of the moving object aligns one pixel to the right of the right edge of a nearby object.

8.3.7 Snap to Grid

When activated via the tool bar or the edit menu (see Section 8.6.1), the LCD preview provides a snap to grid function. Grid size and offset are configured in the project settings (Section 11.4.3), since they are potentially different for each project. Note that there will be no grid drawn on the LCD preview, but objects will snap to the grid and the mouse cursor will be restricted to the grid while in vector drawing mode. The following rules apply:

- When moving individual edges of a control during resizing, the new edge position will align with the grid.
- When moving an entire control, or a group, the center point of the control or group will align with the grid.
- When snap to edges is active at the same time, the snap to edge algorithm will be run **after** the grid locking, based on the grid-locked position. This means that aligned edges take precedence over grid locking. In many cases, using both functions at the same time may not be desirable.
- When the **shift** key is pressed and held down while moving, snap to grid and snap to edges is temporarily disabled. Note that the move has to be started first without holding shift. Otherwise the preview will be in multi-select mode and will not allow object move.

NOTE:	<i>Aligning all four edges of a control with the grid does not necessarily yield a center point which is also located on a grid point. Therefore a subsequent move of the control may result in the edges being no longer aligned with the grid.</i>
--------------	--

8.3.8 Pan and Zoom

Using the mouse and the keyboard, the LCD preview can be panned and zoomed as needed. The following actions are supported:

- **Mouse Wheel:** Scroll up and down.
- **Shift + Mouse Wheel:** Scroll left and right.

- **Ctrl + Left Button:** Pan in all directions (move the mouse).
- **Ctrl + Mouse Wheel:** Zoom in and out, centered on the current position of the mouse cursor. Place the cursor at the location of interest, then zoom in.
- **Ctrl + Keypad + / -:** Zoom in and out.
- **Ctrl + 0:** Zoom to fit. The entire page will be visible.
- **Ctrl + 1:** Set a 1:1 zoom factor.
- **Ctrl + 2:** Set a 2:1 zoom factor.
- **Ctrl + 3:** Set a 3:1 zoom factor.
- **Cursor Keys:** If no control is selected, pan in all directions. Otherwise manipulate the selected controls (see Section 8.3.10).
- **Z + Left Button:** Zoom to window. Drag to define a zoom window. When releasing the mouse button, the preview will zoom in to show the selected area.

NOTE: *As can be seen from the above list, the control key is associated with panning and zooming. Remember to hold the control key down whenever moving around on a large page. While the key is pressed, the preview can be panned in all directions by dragging with the mouse, zoomed in and out via the mouse wheel, or set to fixed zoom levels by pressing keys 0 to 3.*

8.3.9 Edge Scrolling

The LCD preview window will automatically scroll the view whenever the mouse is moved near the edges of the view while the left mouse button is pressed. This helps when moving objects and editing the contents of vector controls. Scroll speed increases when the mouse cursor is moved further out.

8.3.10 Move and Resize

If a control is selected, the cursor keys and the mouse can be used to move the control and change its size. The same is true for a multi-selection, in which case the same change is applied to all selected objects individually. The following commands can be used:

- **Cursor Keys:** Move the selected object up, down, left, and right.
- **Shift+Up:** Reduce the height by 1 pixel.
- **Shift+Down:** Increase the height by 1 pixel.
- **Shift+Left:** Reduce the width by 1 pixel.
- **Shift+Right:** Increase the width by 1 pixel.
- **Left button inside active area:** Move the selected object (any direction).
- **Left button inside active area+h:** Move the selected object horizontally.
- **Left button inside active area+v:** Move the selected object vertically.
- **Left button along an edge of the active area:** Move the selected edge, either left/right (vertical edge) or up/down (horizontal edge).
- **Left button in a corner of the active area:** Resize the object.

- **Right button inside active area:** Open the context menu for more operations regarding the selected object.

NOTE: *Keep the cursor key pressed to make larger changes to the object. For each press and release of the cursor key, an undo step will be recorded. Pressing and holding the cursor key for a while will assign the entire movement to one undo step, which is easier to undo.*

8.3.11 Vector drawing

If the selected control is a line or polygon drawing, it can be edited in the LCD preview. To do this, the control must first be set to edit mode. Press the key **e** or use the context menu of the control to activate edit mode. Once activated, the control occupies the entire page and the drawing can be modified. Please refer to Section 9.5.11.1 for a detailed description of the vector control drawing mode.

8.4 Main Menus

Many of the items in the main windows menus do not require additional explanation, as they are commonly found in PC software. Also, a short description of every menu item can be seen in the bottom left of the main window when moving the mouse over the menu item in question.

All items specific to this configuration software are referenced from the individual sections in this manual which explain the items function, except for some of the more general functions, which are explained in the following sections.

8.4.1 Undo / Redo

Undo and redo of all changes done in the main program window is available via two buttons in the tool bar, the edit menu of the main window, or the usual accelerator key combinations *Ctrl+Z* and *Ctrl+Y*.



Changes to the data point configuration, images, sounds, or fonts are currently not covered by this system, so the project should be saved before major rework is done to these areas. For example, using undo to bring a deleted bitmap control back will not be able to also restore the images or data points used by the control, in case they were removed from the project in the meantime (while they were unused).

NOTE: *The undo history will be kept during a normal project save, so that it is possible to use undo to return to a state before the project was saved. However, if the option to automatically generate a corresponding LWEB project is active, a project save will invalidate the undo history due to the necessary intermediate project conversion.*

8.4.2 Find Objects

The *Edit* menu provides a *Find...* function which opens a dialog to locate specific objects in the project. This can be useful for example to get a list of all trend controls or all pages with a certain name. The items of the resulting list can be selected and the object will be selected in the main window for review. The find dialog may be kept open side by side with the main window, so that new search requests can be executed and the resulting item list can be used to jump to the found objects in the project.

The basic settings in the search dialog can be found on the top line:

- **Search Pattern:** This input field can be used to enter a search pattern to apply to the name and/or the object path. If the field is left empty, no restrictions are placed on the object name or location.

- **Include Path:** If this option is checked, the search pattern is not only applied to the object name but the entire object path (the location within the project) is searched.
- **Match Case:** This option causes the search pattern to be case sensitive.
- **Regular Expression:** Check this if the search pattern is a regular expression instead of a simple DOS pattern. In contrast to a simple pattern, which only accepts one star (*) to denote an arbitrary number of unknown characters, a regular expression is a lot more flexible and can be used to match almost any complex naming scheme. See section 15.8.3 for more information about regular expressions.

Below the basic settings, there are a number of areas to filter objects based on type or location. The following filter options are available:

- **Object Type:** In this area, one or more of the available object types may be selected to limit the search to these types. If the object type does not matter, use the button *Set All* to mark all possible object types as valid search results.
- **Control Type:** This area becomes available if the object type *Control* is selected. It is used to further narrow the search on specific control types. Again, use the *Set All* button to select all control types if the type is not known or does not matter.
- **Shared Objects:** This area is used to extend the search to objects which are not located at a specific position in the object tree but are shared resources, like images, sounds, or fonts.
- **Search Areas:** These options allow the user to limit the search to one or more of the top level objects in the tree. The main menu, global objects, and template area can be enabled and disabled separately using these options.

At the right side of the dialog, there are some buttons available to execute the search:

- **Like Selected:** This button is used to initialize the search options to the properties of the currently selected object. If you want to find objects similar to the one which is currently selected, press this button to have the options preset. Then modify them as needed before executing the search.
- **Find Objects:** This button executes the search.
- **Show Next:** This button selects the next available search result from the result list and jumps to the object in the main window.
- **Show Previous:** This button selects the previous object from the search results and shows it in the main window.
- **Close:** Closes the search dialog. The current search options are kept and will be set the same way when the dialog is opened the next time.

The lower half of the dialog contains the search result list. Select one of the found objects to see it in the main window.

8.4.3 Find Next

To make it easy to locate objects in the object tree which are similar to the currently selected object, the function *Find Next* from the *Edit* menu or the keyboard shortcut F3 can be used. Select the desired object and press F3 to skip forward to the next similar object.

An object is considered similar if it is of the same basic type (menu, menu item, page, control, data point ...) and has the same name. This allows for example to find all uses of a certain data point reference, all 'back' buttons, all 'navigation bar' folders and so on. If the project uses meaningful names for the objects, this function can be very convenient.

For more complex searches, use the full-featured search dialog (see Section 8.4.2).

8.4.4 Find by UID

This command from the *Edit* menu allows to search for an object by its UID. The object UID is shown on the **General Properties** page. This function is primarily intended for debug purposes, when a system log entry mentions an object by its UID, and the corresponding object should be located in the project.

8.4.5 Device Model and Firmware

When the configuration software is started in standalone mode and is not connected to a device, the *Model* menu can be used to select the model for which a project should be created, and the *Firmware* menu can be used to select the firmware version of the intended target device or software version of the LWEB-80x client. This enables the user to create or edit a project, which is intended to run on an older firmware version or with an older LWEB client software.

8.4.5.1 LVIS Devices

When started as a plug-in or connected via TCP/IP or CEA-709 in standalone mode, the menu items to select model and firmware version are grayed out, so that the firmware version cannot be arbitrarily changed by the user. The menu items of the correct model and firmware version, as reported by the device, are automatically checked. The real time preview of the LCD is automatically adjusted to reflect the behavior of the reported firmware version and the project is saved in a format readable by the firmware of the connected L-VIS device.

The configuration software is always backwards compatible to older firmware versions, so that it is possible to use newer configuration software with older devices. Also, the devices are backwards compatible to older project versions, so it is possible to use older configuration software with newer devices as well.

NOTE:	<i>When using older configuration software to connect to newer devices, the software will not recognize the firmware version reported by the device, since this firmware was not available at the time the configuration software was built. In this case, the software will output a warning and save the project in the most recent and supported version. This will be readable by the device, due to the devices backwards compatibility to older project files, but you may not be able to use all features, and the LCD preview may not accurately reflect the behavior of the device. It is therefore suggested to first upgrade the configuration software, and then upgrade the devices.</i>
--------------	---

The firmware of the L-VIS device can be upgraded as newer firmware versions become available. To find the current firmware version of your device, use the *Device Info...* menu item in the *View* menu, which will be enabled when the software is connected to a device or go to the About Page on your device.

While it is always possible to upgrade to a newer firmware and keep the device fully functional without changing its configuration data, it is often not possible to downgrade to an older firmware version, since the older firmware may not be able to read the data stored by a newer firmware. If a downgrade is necessary, it is strongly recommended to first clear the device by either selecting the command **Remove Configuration** from the *File* menu of the main window, or reset the device to factory defaults before conducting the downgrade.

To upgrade the device with a new firmware, select the *Upgrade Device...* menu item from the *Firmware* menu, which will be available when connected to a device.

NOTE:	<i>Do not attempt to downgrade the device to a firmware version older than the one shipped with the device. Old firmware may not be able to run on new hardware revisions, and may render the device unusable. Always check with support before downgrading a device, unless it is a simple roll-back to a previously installed firmware.</i>
--------------	---

8.4.5.2 LWEB Projects

In case of LVIS projects, the project format is automatically adjusted to match the detected device firmware before download. However, the configuration software has no way to know which version of LWEB-802/803 client software will be used to load a project saved for the LWEB-80x model. It is the responsibility of the user, to make sure the used LWEB client software is capable of reading the chosen project format, selected via the firmware version menu.

To avoid compatibility problems, it is recommended to load the LWEB client software from the default URL on the LOYTEC webserver (<http://www.loytec.com/lweb802/>). This way, the client will always be the latest release, which is tested to work with all features available in the most recent configuration software.

In cases where automatic update of the LWEB client version is not desired, or the devices running the client do not have access to the LOYTEC webserver, locally installed versions of the LWEB client software can be used. In this case, make sure to upgrade the local client software as needed, before deploying projects saved for new LWEB versions.

NOTE:	<i>For maximum compatibility to older client software, do not set the version of the project higher than necessary to enable all required features. Increasing the version beyond the required minimum without actually using any of the new features, can break backward compatibility to older LWEB client software, without adding any benefit for the project.</i>
--------------	--

8.5 Context Menus

Context menus are an important part of the workflow, since most operations related to the objects of the L-VIS project are accessible from the context menu of the object in question. The context menu is usually opened using a right mouse button click on the desired object in the tree view, but for objects visible in the LCD preview, the same context menu may also be opened by a right mouse button click on the preview of the desired object.

Commands accessible via the context menus are detailed in the following sections.

8.5.1 Add Objects

Most context menus contain a number of *Add...* commands to add new objects to the selected object. Only those objects which may be added to the selected object are listed in the context menu.

8.5.2 Enable/Disable Access Control

These commands are available in the context menu of menu objects and menu item objects. The command *Enable Access Control* will go through all pages below the selected menu or menu item and turn on the *Locked Page* flag for all pages which have a non-public access level set. The command *Disable Access Control* executes the inverse operation, turning off the *Locked Page* flag on all pages below the selected object. This function is useful for testing purposes.

8.5.3 Cut / Copy / Paste

Every context menu contains commands to cut, copy, paste, and delete the selected objects. Aside from using the context menu, objects in the tree view can also be copied using the *Edit* menu from the main window or the keyboard shortcuts shown next to the commands.

The commands **Cut** and **Copy** will place a copy of the selected objects with all their child objects into the Windows clipboard. The copy is therefore not only available for paste into the same project but may also be pasted into any other project.

NOTE:	<i>Especially with larger projects, creating a copy of an object including all its references to data points, sounds, fonts, images, and templates requires some processing time. In cases where the object should just be removed from the project without the need for a copy in the clipboard, it is usually faster to use Delete instead of Cut.</i>
--------------	--

8.5.3.1 Data References

If the copied object tree contains references to data points, these references (data point objects) are copied without modification. For example, the copy of an entire page will reference the same data points as the original page. If the new copy should reference other data points, use the command *Manage Data Points* from the context menu of the copied object and reassign the data point objects to other data points, as explained in section 8.5.5, or consider using templates.

If the copy is pasted into a different project, the referenced data sources may not be available at all. In this case, the required data sources are automatically created in the destination project, so that the references may be resolved. For example, it is possible to copy a page which operates on register data only, into a completely different project and the page will function the same way it did in the original project. Registers which exist (for example system registers) will be used directly, while others will be created as necessary.

8.5.3.2 Fonts

If the copied object uses custom fonts and the object is pasted into another project, these fonts may or may not be available, depending on the setup of the destination project. The software will search all loaded fonts in the destination project to find the required fonts (based on the font size, type and family as displayed in the font dropdown list). If a suitable font is found, it will be used automatically. If the required font is not loaded, the software will try to load and use the font from the copied object. If this fails, for example because the additional font would exceed the capacity of loadable fonts, an alternative font from the destination project must be assigned by the user.

8.5.3.3 Translations

When objects are copied from a multi-language project, all text and font data required to display the object in any of the available project languages is stored in the clipboard. During paste, languages of the destination project are matched to languages available in the source project by locale ID (language/region pair). It is not required for both projects to use the same languages, or define them in the same order. Data is taken over for as many languages as possible, while data for languages only existing in the destination project is initialized from the primary language of the source project. Text and font data for languages only existing in the source project is not pasted, but remains in the clipboard.

8.5.3.4 Images and Sounds

If the copied object uses images or sounds and the object is pasted into another project, the referenced items will be copied to the destination project automatically, if they are not available yet. This will ensure that the copied object stays fully functional.

8.5.3.5 Template References

If the copied object contains instances of templates and the object is pasted into another project, the configuration software will try to identify a suitable template in the destination project based on the number, type, and order of objects in the template. Template instances, for which there is no suitable template available, will cause the original template from the source project to be copied over to the destination project automatically.

NOTE: *It is possible to change details of the template like for example the position or other properties of the contained objects, without causing problems for the software to recognize the template as suitable for use. If a template is automatically created by a paste operation and the template is then changed slightly before another paste operation requesting the same template is executed, the template will not be created again. However, if significant changes were done to the template in the meantime, the second paste operation will no longer recognize the edited template as suitable (compared to the template from the source project) and will therefore create a fresh copy of the template from the source.*

8.5.4 XML Export / Import

Using the *Export to XML* command, objects in the tree view may be exported to an XML file. The exported object will include all child objects, but there will be no information about loaded user fonts or data points included in the XML output, so that importing the XML file into a different project may lead to unresolved references to fonts or data points if the project configuration of the destination project differs from the source project.

The *Import from XML* command is similar in function to a paste command. The difference is that it reads the object to paste from a given XML file instead of the windows clipboard.

NOTE: *The XML export and import functions are intended to be used for custom tools which may be able to build a complete project (in XML format) out of basic building blocks, which were prepared and exported to XML fragments. Based on information in a network database, a custom tool may read the individual XML building blocks which for example describe individual pages of the project and put them together into a complete project, which may be loaded into the configuration software and downloaded to the device. Note that it is also possible to save an entire project in XML format or export global parts of the configuration, like the font set, the project settings, or the data point configuration. See section 15.7 for details about this function.*

8.5.5 Manage Data Points

This command is available in the context menu of menus, items, pages and folders. It will open a new dialog, listing all data point objects below the selected object, which data points they reference and which type and direction they have. This dialog is useful to organize data point references, do batch renaming of data points or reassignment of data point objects to different data sources, as is often required when copying parts of a project, for example to add a page for a new room based on the page of an existing room, but referencing data points of the new room.

Depending on the selected objects in the data point list (multi-select is possible), some of the operations available using the buttons at the bottom of the window may be disabled.

8.5.5.1 Naming Rules

Many of the batch operations open a dialog to set the naming rules for the batch operation. The dialog provides three different levels of naming rule specification, from very simple to very complex and powerful. The three options are:

- *No Custom Naming Rule:* This is the most basic setting. It does not apply any special rules on the source name or the target name. All selected sources will be processed and the target name will be chosen to be as close to the source name as possible. For example the duplicate operation, when run with this naming rule, will create a data point Test1 from the original data point Test.
- *Search and Replace:* This setting is used to specify a simple search and replace algorithm. All data sources which contain the search string (given in the first input field) will be processed. The last occurrence of the search string in the original name will be replaced by the replacement string (given in the second input field) to form the new name. Most often used in conjunction with the

reassign operation, for example to automatically reassign all data points referencing xxx_Room001 with xxx_Room002 (replace 001 with 002).

- *Regular Expression:* This is the most powerful setting. The regular expression given in the first input field is matched against the original name. If the expression matches, the entry is processed, otherwise ignored. If processed, the matching groups (if present) are extracted and are used together with the name template given in the second input field to build the new name. The new name will be composed of the text in the name template, with all occurrences of \0 replaced by the contents of the first matching group, all \1 replaced by the contents of the second matching group and so on. For example, the regular expression `{(nvi)|(nvo)}{.*}_{[0-9]}` will match the original name **nviRoomTemp_2** and assign the substrings **nvi** to \0, **RoomTemp** to \1, and **2** to \2, so that the new name, when using a template of `\0\1-3` will become **nviRoomTemp-3**. Please refer to section 15.8.3 for a complete description of the regular expression syntax.

Below the naming rule settings is a preview window, which shows a preview of the resulting names and the number of matches in real time.

8.5.5.2 Rename Data Source

This operation allows batch renaming of data sources (the actual register, NV, or server object name). All data sources which are mentioned in the selected entries will be considered for renaming. If a particular source is actually renamed depends on the given naming rules, as described in the previous section.

NOTE:	<i>Renaming a data source does not reassign any references to that source, that is, all data point objects which referenced the original data source will reference the same renamed data source after this operation. To actually reassign data points to different data sources, use the 'reassign' or 'duplicate and reassign' operations explained below.</i>
--------------	---

8.5.5.3 Select Data Source

This operation is available if all selected entries reference the same data source. It allows using the data point manager window to select a replacement data point, which must be of compatible type (same type and direction as the original source).

8.5.5.4 Reassign Data Point

This operation allows reassigning a batch of entries to a new data source, based on the currently referenced data point. Use the naming rules to construct the desired name of the new data source from the name of the currently assigned data source.

This is most useful to remap all data points referencing data from one room to similar data of another room, where the data source name contains the room number for easy remapping (only the room number from the original name must be replaced by the number of the new room to construct the name of the new data source).

NOTE:	<i>In contrast to the 'duplicate and reassign' operation, this operation only attempts to reassign the data points to other existing and compatible data sources. New sources will not be created on the fly. Instead, the reassign operation fails if no compatible data source can be found using the new name.</i>
--------------	---

8.5.5.5 Duplicate and Reassign

This operation is similar to the normal reassign operation, only that the new name is not used to search for an existing data source but the new name is used to create a copy of the original data source and the selected item is reassigned to this copy.

The function keeps track of the copies it creates and ensures that each unique source is copied only once. For example, if more than one selected entry references the same original data source, one copy will be made of this source and all entries will be reassigned to this copy. As a result, data points which referenced the same source before the duplicate operation will reference the same source after the duplicate as well (only the references will be to the copy of the original source).

8.5.5.6 Create Static NV / Server Object

This operation allows creating local data objects from remote data objects. Depending on the network technology used by the device, it is possible to create static NVs which are complementary to referenced external NVs or to create local server objects which are complementary to an existing client mapping.

The operation works in the same way as duplicate and reassign, except that it creates new data objects based on existing remote sources, instead of existing local sources.

8.5.5.7 Remove Data Point

This operation removes the selected data point objects from the project. It is useful to remove a number of data point objects in one step, instead of going through the object tree in the tree view and removing each data point individually.

Removing data point objects may be necessary to be able to remove the referenced data point, since the data point keeps a reference count and cannot be deleted as long as it is referenced by a data point object in the project.

8.5.6 Find Instances

This menu item is available on templates only and opens the find dialog (see Section 8.4.2) to show all template instances of the selected template.

8.5.7 Expand / Fold Tree

The commands to expand and fold the object tree are available on menu items, pages, and folders. They can be used to expand or fold the object tree below the selected object.

8.5.8 Lock / Unlock Position

These entries are available on all visible objects on a page. They can be used to lock or unlock the screen position of the object, so that objects are not moved unintentionally while working on other objects on the same page. Also, a locked object does not show up as a candidate for selection and is not selected when clicking on it or spawning a selection box containing the object. See also Section 8.3.2 for details regarding object selection.

Note that the lock function applies to the LCD preview window only. Locked objects may still be selected in the tree view and moved to a different position in the tree once selected. The lock only prevents objects from being selected in the LCD preview and from changing their position or size. Position and size properties will be grayed out for a locked object.

Applying the lock command on a collection will lock the collection and all child objects of the collection. As long as the collection itself remains locked, the lock/unlock commands will be disabled for the individual objects below the locked collection.

NOTE:	<i>If only some of the objects in a collection are locked and the collection is moved, only the unlocked part of the collection will move, while all locked items will remain at their current position. This may change the bounding box of the collection as a result of the move.</i>
--------------	--

The lock state of individual objects in a template will not be taken over to instances of these objects. This allows the instance to be positioned via the LCD view, without some objects

getting stuck in their original position due to being position-locked. If needed, the entire instance can of course be locked into place after positioning.

8.5.9 Hide / Show, Hide Contents, Show Contents

These commands are available on pages, groups, and other visible items in the object tree. They can be used to hide and show the selected object and all of its child objects. Hidden objects have their name displayed in square brackets in the object tree.

A hidden object will not be visible on the device after project download, but it can be made visible during run time via a trigger command.

8.5.10 Add Items from Data Points

Available for menu objects, starts the automatic page create wizard described in section 15.4. In this mode, the selected data point folder will be searched for sub folders and a new menu item will be created for each sub folder. If there are data points located directly in the selected data point folder, they will be ignored. This command is useful to visualize an entire device folder or other high level data point folder which itself does not contain any data points.

8.5.11 Add Pages from Data Points

Similar to the above command, but available for menu items. In this mode, the selected data point folder will be scanned for data points and pages will be created to visualize them. If there are sub folders present, a new sub menu will be created to hold the menu items for the data point sub folders, similar to the operation above (add items from data points). The page wizard is described in more detail in section 15.4.

8.5.12 Update Instances

Available in the context menu of templates, causes all instances of the selected template to be updated. Note that updates are done automatically whenever an object outside the template is selected after the template was modified, so this command should normally not be required.

8.5.13 Update Symbol(s)

This command is available on template folders and templates which represent symbols in L-Studio. It triggers a re-load of the selected symbol or symbols below the selected folder, as if these symbols were changed and needed a refresh. Usually not needed, but can be helpful to check that a refresh of a symbol will not cause any problems.

8.5.14 Update Subproject

This command is available on menu items which represent an instance of a sub project in L-Studio. Similar to the symbol update command, re-loads the entire sub-project and updates the instance, as if the sub-project was changed and a reload was required.

8.5.15 Restore Defaults

This command is available in the context menu of template instances, and all objects which are part of a template instance. The command becomes available on objects for which there are local changes recorded. Using this command, the corresponding object will be restored to the state defined by the template, that is, all local changes related to the selected object will be discarded. Calling this command on the top level of a template instance will restore the entire instance to the default state.

8.6 Tool Bar

The tool bar in the main window contains some of the commands which are used more often. Pointing the mouse on one of the tool bar buttons will open a small bubble help with a short description of the buttons function. All functions of the tool bar are also accessible from the main menu. Similar to the menu items, most of the buttons will be mentioned in the sections where their functionality is described, except for some of the general functions which are listed in the following sections.

8.6.1 Preview Settings

There are three toggle buttons to control the most often needed aspects of the LCD preview.

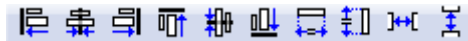


They turn the following features of the preview on and off (from left to right):

- **Resize Frame:** This button controls display of the resize frame (Section 8.3.4).
- **Snap to Grid:** The middle of the three buttons toggles the snap to grid function, as described in Section 8.3.7.
- **Snap to Edges:** The last button toggles the snap to edges function, as described in Section 8.3.6.

8.6.2 Layout Tools

To aid in the layout of controls on the page, there are a few buttons located in the tool bar which become available when two or more controls are selected in the LCD preview.



To select more than one control at the same time, select the first control, press and hold the shift key, and select additional controls.

NOTE:	<i>It is important which of the controls you select first. The control which was selected first will act as the reference for the layout functions, as explained below.</i>
--------------	---

Once the controls were selected, execute the desired layout functions by pressing the buttons in the tool bar or using the commands from the *Layout* menu of the main window.

8.6.2.1 Alignment Functions

The alignment functions will align the left, right, top, or bottom edge, or the horizontal or vertical center lines of all selected controls. The reference will be the first selected control.

8.6.2.2 Make Same Size

These functions make all selected controls the same width or height as the first selected control. This function is usually used together with controls of the same type.

8.6.2.3 Space Evenly

These functions require at least three controls to be selected. They modify the horizontal or vertical position of the selected controls such that all controls have the same amount of empty space between each other. This function automatically determines the two controls which have the greatest distance from each other and spreads the remaining controls out between them, such that each control remains as close to its original position as possible (the overall order of the controls is not changed, no matter in which sequence you selected the controls).

NOTE:

This function may cause the rightmost or bottommost control to move by a few pixels, if the original distance between first and last control did not allow the other controls to be evenly distributed between them. This may be the case because a control can only be placed at an integer pixel position, which means that the calculated space between the controls can either be 4 or 5 pixels, but not 4.5 pixels. If the total amount of originally available space is not a multiple of the calculated best fitting distance, the last control may need to be moved a little to accommodate for the rounding error.

8.7 Images

The LVIS-3E100 and LVIS-ME200 devices use a 256 color display with a standard VGA palette. All controls and images can therefore only use colors from the VGA palette, which is the same as the default Windows palette in 256 color mode. It is usually available for selection in image editors when the drawing mode is set to indexed color. Projects of this type are also referred to as VGA projects and are considered legacy since these devices are already EOL.

All other models support true color images. This type of project contains embedded JPG or PNG data which is used by the device to render the images. Images using PNG format can also contain transparency information in a so called alpha channel. True color images are displayed in the preview in up to 32-bit color depth (ARGB format). Note that currently available LVIS devices are limited to 18-bit color resolution and will therefore only display 262144 different colors (256k) on the LCD screen. Refer to Table 1 for details.

8.7.1 Image Manager

All images used in the project are managed in a separate dialog window, which can be opened either via the menu *Edit->Images...* or via the various buttons used to select an image for the selected object.

The image manager shows a small thumbnail for each of the loaded images, along with information about the image name, size, and usage count. Hovering over a thumbnail opens a tooltip with more detailed information about the image, like the source file from which the image was loaded, the image format, and if the image contains alpha channel data (transparency).

Below the image tiles is a group of buttons to manipulate the selected images or load new images. The following functions are available:

- **Add:** Opens a file dialog to select and load a new image. Another option to add new images is to just drop the file anywhere inside the image manager window.
- **Reload:** Causes the selected images to be reloaded from their last known source file (as shown in the tool tip). This can be useful to refresh a project when still working on an image in an external graphics program.
- **Resize:** This button is used to resize the selected images. For each image, a dialog is shown to enter the desired new size. The original image data is needed for this function. In case it is not embedded in the project, it will be reloaded from the last known location on disk (see also Section 8.7.2).
- **Copy:** Creates copies of the selected images. The duplicated images can then be resized, rotated, or flipped as needed.
- **Replace:** Replaces all selected images with one new image loaded from a file on disk. Remember to review the project for necessary changes in layout, if the replacement image has different dimensions.

- **Reassign:** Similar to replace, but the replacement is picked from one of the already loaded images.
- **Export:** Generate new image files on disk from all selected images in the project.
- **Remove:** Remove the selected (unused) images from the project. Any source data which is no longer needed is removed as well.
- **Rotation:** Specify the amount of clockwise rotation, compared to the original image.
- **Mirror:** Specify if the image is to be flipped along its horizontal or vertical axis.
- **Lock aspect ratio:** If set, this image will always be scaled proportionally, otherwise its aspect ratio may change since width and height are scaled independently. This option is important for automatic scaling of bitmap control contents. Note that automatic upscaling of pixel images is only done when this option is not set.
- **Remove Unused:** This button will remove all images which are currently not used in the project. This can also be done when the project is written to the device. If there are any unused images at this time, a dialog appears is shown to check if the unused images should be deleted before download or not.

8.7.2 Image Size and Auto-Scaling

In many cases, an image will be required in different sizes. For example a larger size for a push button, and a smaller one for an icon. There are different approaches to handle this:

1. Load two different images, maybe even slightly different in design. For each button or icon, select the desired image from the image manager. The bitmap control, which is set to auto-size by default, will adjust its size to the size of the selected image.
2. Load the image in the largest resolution required, and create smaller (down-sized) copies using the *Copy* and *Resize* buttons in the image manager. As above, select the desired image for each button or icon.
3. Load the image in the largest resolution required, or even better, load it in SVG format, and assign this image to all buttons and icons. Manually set each bitmap control to the desired size. The control will leave auto-size mode and the assigned image will be scaled automatically to fit the control.

Each of these approaches has its advantages and drawbacks. The first approach is suitable for cases where there is a large difference in image size, such that down-scaling the same image does not work well. In this case, there are essentially different images used for the different buttons and icons.

An advantage of the copy and resize approach could be that the controls stay in auto-size mode and will adjust themselves automatically to any change in size of the assigned image. It is therefore possible to resize an image using the image managers *Resize* button, and all controls (in auto-size mode) using this image will automatically adjust their size.

The third approach is probably the most intuitive one. Select and assign the image from the image manager and adjust the control to the desired size afterwards. The advantage is that each image is only listed once in the image manager, no matter how many size variations exist in the project. The disadvantage could be that the control leaves auto-size mode and will no longer adjust its size when the imported image is resized via the image manager. It is suggested to use templates, to ensure uniform control sizes and allow adjustments to the design later on, without manually resizing every single control.

NOTE:

All of the above can be combined as needed. For example, two separate images can be prepared in the image manager for the two sizes which are used most often, which ensures uniform control sizes and a clear design, while a few extra sizes can be done by resizing the control, which avoids the need to create and manage additional image variations just for these special cases.

8.7.3 Embedded Sources

When an image is loaded, its original file data is embedded in the project. This helps for operations which need the original image data, when the project is edited in a different environment than it was originally designed, and the original files are no longer available.

The same source data is embedded only once, even when it was imported a number of times in different sizes or rotations. When the last image generated from a specific source file is deleted, the embedded original data is removed as well.

8.7.4 Color Reduction

If a true color image is loaded, like for example a JPG or a 32-bit BMP or PNG file, the configuration software will automatically create a VGA color version using a Floyd-Steinberg dithering algorithm. When used in a VGA project, this color reduced version is used, otherwise the original data is used if already available in JPG or PNG format, or the original image data is converted to JPG or PNG, depending on the image properties.

In VGA projects, dithering will be noticeable when viewing the images from a close distance, which will be OK for photo-realistic color graphics, but might be unwanted for line-art graphics, which contain large areas of uniform color. In such areas, dithering to the exact color would not look as good as just picking one color from the VGA palette which is closest to the intended color. If this is required, it is best to design the graphics using the VGA palette and saving them in 256-color indexed format, like for example 8-bit BMP or GIF. It is important to make sure that the palette used by the graphic tool is in fact the VGA palette, otherwise the image will still need dithering.

8.7.5 Transparency

Transparency is handled in different ways, depending on the color mode of the project and the properties of the image. In VGA projects, all images are displayed using VGA colors only and transparency is defined by selecting a color index from the VGA palette. All pixels using this color index are considered transparent if transparent background is turned on. In true color projects, transparency is more complicated. Several cases exist:

- The image contains transparency information (alpha channel data). In this case, transparency is defined separately for each pixel in an additional set of data embedded in the image file (the alpha channel). All pixels of the image will be drawn according to their RGB color values from the main image data and the opacity values from the alpha channel, independent of the *transparent background* option.
- The image does not contain any transparency information and uses colors which are not part of the VGA color palette. In this case, all pixels are drawn fully opaque, because there is no alpha channel data available and the specification of a transparent color by VGA color index is not possible. This is the case for JPG or BMP files and **also for resized VGA images**.
- The image does not contain any transparency information, but uses only colors from the VGA palette. In this case, transparency can be defined by VGA color index in the same way as in VGA projects.

The LCD preview will show the image in the same way as the device, so the behavior of transparency can be checked there. Also, pointing the mouse over an image in the image manager will show a tool tip containing more information about the loaded image. In this

tool tip, it will be noted if the image contains alpha channel data and what the original image format was.

8.7.6 Animations

Animation is supported via animated GIF and true color MNG files. MNG animations are preprocessed by the configuration software when used in projects for VGA devices, such that both formats may be used independent of the target device model. Note that for true color devices, the MNG format is preferred as it avoids color dithering artifacts.

NOTE:	<i>Keep in mind that playing animations can result in a significant use of both memory and CPU time. Also, there is no special handling for animations which contain only one frame. Use a still image file format like JPG or PNG for images which are not true animations.</i>
--------------	--

8.7.7 Supported File Types

The following image file types are supported:

- Bitmaps (*.bmp)
- Icons (*.ico)
- Joint Photographic Experts Group / JPEG (*.jpg)
- Tagged Image File Format / TIFF (*.tif)
- Portable Network Graphics / PNG (*.png)
- Multiple Image Network Graphics / MNG (*.mng)
- Scalable Vector Graphics / SVG (*.svg)
- Graphics Interchange Format / GIF (*.gif)

The file open dialog automatically shows all supported graphic files it finds in the directory. Unsupported files are hidden, for example Visio, Corel-Draw, or Auto-CAD proprietary vector drawing files, which cannot be imported directly. Use the programs export function to save the image in SVG or PNG format for use in the L-Vis project.

NOTE:	<i>When using SVG format, make sure the image contains a viewBox attribute. This attribute is used to define the exact boundaries of the image, making sure that all versions of configuration software and LWEB-802, running in different browsers, will render the image in the same way. Although SVG files without a defined viewBox may be imported, they may render differently on different browsers or different versions of configuration software, since a bounding box will be calculated automatically based on the vector data of the image. Algorithms to do this may vary and yield different results.</i>
--------------	--

8.8 Fonts

Every device has a set of three basic fonts which cover the ISO-8859-1 (Latin-1) code set in three different sizes. To extend the available choices, the configuration software and the device support user loadable fonts in most common formats, including TrueType, Type1, BDF, PCF, OTF and others.

The L-VIS configuration software is shipped with a set of selected freeware fonts to cover Western (ISO-8859), CJK (Chinese, Japanese, and Korean) and also Unicode (ISO-10646) character sets. These should cover your immediate needs for a large number of languages.

In addition, any of the available Windows TrueType fonts may be used.

Fonts in PCF or FNT format are fixed size fonts, meaning that each of them is designed and optimized for a specific resolution. Most of these fonts are therefore available in a number of variants, starting at very small sizes, up to about 24 pixel resolution. When loading a font which is available in a vector format like TrueType or OpenType, you will have to specify additional parameters, like font face and size in points (1/72 inch). The configuration software will then add the font in the desired style and size for you.

NOTE:	<i>Vector fonts are rendered with anti-aliasing technique on all devices except LVIS-3E100 and LVIS-ME200, which are limited to the VGA color palette. This may lead to faded line segments when the line width falls below the LCD resolution at small font sizes. For this reason, some vector fonts contain optimized pixel-versions for small sizes, which are automatically used in place of a rendition of the vector data. If a font does not provide optimized data for small sizes, anti-aliasing may be turned off.</i>
--------------	---

8.8.1 Font Import

New Fonts are added to the project via the font import dialog. To open the dialog, either choose *Select new font...* from the font selector dropdown, or press the **Add** button in the font manager (see next section).

The font import dialog opens with a list of installed Windows fonts to choose from. Depending on the selected font, a number of different font faces or styles may be available. The standard face is usually called *Regular*, other faces which are frequently available include *Bold*, *Italic*, *Light*, *Condensed*, *Narrow*, *Black*, or any combinations. Upon selection of a specific face, a preview is shown in the box below the font settings.

Since a TrueType font can be rendered at any size, the desired import size must be selected before adding the new font to the project via the **OK** button. This is because an imported user font must always be fully specified, including font family, face, and size.

To import a font from a file rather than the operating system, press the **Open File...** button. Select the desired font file using the file open dialog. The font import dialog will then show the font properties detected in the file and allow to specify any options which are available. In case the selected file contains a collection of font faces, like a TTC font catalog, the desired face may be selected from the **Face** dropdown. In case it is a vector font, the desired size can be selected from the **Size** dropdown.

NOTE:	<i>For fonts imported via file, no preview is available. This is because the imported font may use a format not natively supported by Windows, for example PCF, so it cannot be displayed in a Windows UI element like the preview text box. Fonts selected from the list of installed operating system fonts can be used in a dialog, so a preview can be shown.</i>
--------------	---

8.8.2 Font Manager

All custom fonts are managed in a separate dialog, which can be opened either via the command *Fonts...* from the *Edit* menu, or via the button *Open Font Manager...* on the *Common Properties* page.

The dialog contains a list of currently loaded fonts with some extra information, like the type of font, size, number of characters contained in the font, the number of times the font is currently used in the project, font licensing information, and the source file from which the font was originally imported or where it was found the last time it was searched (see also Section 8.8.3).

Below the list are some buttons to manage the loaded fonts, add new fonts, and remove unused ones:

- **Add:** Opens the font import dialog described in the previous section. The new font will be added to the list of loaded fonts if not already loaded. The same font can be loaded multiple times in different sizes and variations. New fonts can also be added by dropping a font file anywhere inside the font manager window.
- **Reload:** This will try to reload all selected fonts from their currently known source files. This is of course only available for fonts imported from files.
- **Resize:** Available for vector fonts, allows changing the size of the font. The original source file may be required for this operation, unless it is already embedded in the project (see the option “*Do not embed font source*” below).
- **New Size:** Create a new font in a different size from an already loaded font.
- **Replace:** Replace the selected font with a new font loaded from a file. If this operation causes a change in font size or style, it is recommended to check the project for required adjustments.
- **Reassign:** Reassign all current uses of the selected font to a selected replacement font, so that the old font ends up unused.
- **Remove:** Remove the selected font from the project. Only possible with fonts that are currently unused.
- **Find...:** Opens the find dialog and shows all objects which are currently using the selected font.
- **Assign to Selected:** Assigns the selected font to all currently selected objects (the controls which were selected when the font manager was opened). Useful to immediately see the result in the LCD preview without closing the font manager.
- **Show Target Size:** Changes the display of the font size to show the reduced target size (only containing the characters currently in use). This is useful for the small L-Vis devices, which use prebuilt and optimized versions of the fonts. Large devices usually read the original font source file, so the reduced size has no meaning in this case.
- **Disable anti-aliasing:** In cases where a vector font is rendered in a small size, the line width may fall below the LCD resolution, causing lines to fade out. If the font does not automatically provide an optimized pixel rendering for small sizes, the font rendering can be changed from anti-aliased to monochrome, avoiding faded lines in expense of edge smoothness. The effect is immediately visible in the LCD preview.
- **Do not embed font source:** If possible, the original font source file will be embedded in the project file, such that the source is available for the client to render the project exactly as shown in the LCD preview. For most fonts, this will be OK, but font files can vary a lot in size, ranging from some 50kB to over 20MB. Font sources which are obviously too large will be excluded automatically, but for other fonts, the user may choose to not embed certain fonts to reduce the size of the project, especially for older devices (LVIS-3E100).
- **Use installed font:** Available for LWEB projects, indicates that the actual font data should not be embedded, since it is assumed that a suitable font is already installed on the client. All necessary font properties like type face family, style, weight, and stretch are saved with the project, to allow the LWEB client to pick a suitable font from the system it is running on. The difference compared to the *do not embed* option is that there will be no pre-rendered font data replacing the missing original font file. It is assumed that the font is installed on the client and

the project will be rendered as if the font was embedded. Anti-aliasing and other features requiring the original font data will not be disabled.

NOTE: *If the original font source is not embedded in the project and is not assumed to be installed on the client, the configuration software will analyze the project and pre-render all required glyphs for the client to use as a replacement for the original font data. Since the pre-rendered glyphs are bitmaps, no anti-aliasing will be available. Also, changes in text layout may occur, since the pre-rendered font data will be missing kerning tables and other meta-information required for optimal layout of specific character sequences. The LCD preview will reflect any differences in layout immediately when the options are changed.*

If available, the size of the original vector data will be shown in parenthesis following the path to the source file. If the font source is embedded, some operations like resize and new size can be executed without the need to locate the original file on the PC. In case the original file is not embedded, it will be searched on the PC where the configuration software is running. This mechanism is detailed in the next section.

NOTE: *Complex writing systems and localized controls cannot use pre-rendered fonts at all. Text shaping and layout for complex writing systems needs the original font data to work, while localized controls do not allow the configuration software to determine a complete set of glyphs required during run time to display localized content. It may work for one language, but not for another one, so make sure to **always embed fonts used for localized controls**.*

8.8.3 Font Embedding

When a custom font is used for a project, the original font source is usually embedded in the project. This type of use may be restricted, or require a suitable license from the font vendor. In case of TrueType or OpenType fonts, the font designer may have added certain information in the actual font file, to reinforce what he specified in his license agreement and govern how the font may be embedded in electronic documents. The font manager reads and displays this information, if available.

NOTE: *The configuration software reads and displays the available embedding flags from the font file **for informational purposes only**. It does not enforce a particular use of any of the loaded fonts based on this data, nor does it guarantee that this data was accurately prepared by the creator of the font source file. Embedding information from the font file **does not constitute a license**, and as such cannot replace a written license agreement. It is the responsibility of the project designer to comply with the terms and conditions of a written end user license agreement issued by the font vendor.*

The possible embedding modes and their general meaning, as defined by the OpenType standard, are listed below:

- **Unrestricted:** There is no limit how the font may be used when embedded in the project. It may be installed on another system, modified, used to edit the document and so on. Usually used in open source fonts.
- **Editable:** The font may be embedded and used to edit the document on other systems, where the font would not be available otherwise (not installed). This is the most common mode, particularly for Windows fonts licensed by Microsoft.
- **Preview:** The font may be embedded, but only used for (pre)view and printing, but not to modify the document (add new content using the embedded font), when the font is not available on the system on which the document is opened. This mode is used by some of the 3rd party fonts available on Windows.
- **Bitmaps only:** This is a rarely used mode, in which pre-rendered bitmap glyphs contained in the font file may be embedded, while it is not allowed to embed the vector data (for scalable glyphs).

- **Restricted:** This font may not be embedded in a document at all. To use such a font in a project, obtain a licensed version of the font from the font vendor. A licensed font file should not use this embedding mode.

NOTE:

If a restricted font is embedded in an LWEB project, the browser viewing the project may refuse to load the font, based on the data found in the font file. The configuration software does not alter the original file in any way, to trick the browser into using the font. The font embedding flags stay intact, so the browser can see them and make his own decisions. If a custom font does not show correctly in an LWEB client, check the licensing of the font. It may be a restricted font and the browser in question may enforce the license.

8.8.4 Font File Location

When a project contains a user defined font **imported from an external file**, it may be necessary to access to the original file when working with the project. As long as only the pre-installed fonts are used, access will be possible since these files are located in a known place on every installation of the configuration software. However, if fonts from an external source are used, they too must be available whenever the project is loaded and modified and the user has to make sure to deliver the font file together with the L-VIS project file, if the project is to be edited on a different PC, where the font may not be available in the same location.

Most operations can be completed without access to the original file, when the font is embedded in the project. If this is not the case, the original file has to be searched. To locate the original font file, the configuration software will automatically try the following locations in the given order until a suitable font is found:

- The list of source files embedded in the project.
- The full path from where the font was built the last time.
- The directory containing the project file.
- A **fonts** subdirectory of the directory containing the project.
- The fonts directory of the configuration software installation.
- The Windows system font directory.

If the font cannot be located in any of these places, the user is asked to provide an alternative font file. If the font can be found, the project is updated with the newly located font file to reflect any changes (a new version of the font may have been installed).

Based on the above search order, the easiest way to make sure that a project can be used on any PC is to either embed the source file in the project, or store all custom fonts which are used in the project in the same place where the project file itself is located.

8.8.5 Font Recoding

The L-VIS configuration software as well as the L-VIS device itself is Unicode enabled. All text entered in the configuration software and displayed on the device is encoded using UCS-2 as defined in the ISO-10646 Unicode standard to cover the entire basic multilingual plane (BMP). PCF and BFD fonts may use other encodings, such as ISO-8859, JIS-X0212, Big5, IBM, or Microsoft code pages (for example MS-CP1250). If such a font is loaded, it is automatically recoded to UCS-2 before it is used, since the L-VIS device itself does not recognize encodings other than UCS-2. The font recoding is done using the coding tables found in the directory *Encodings*, located in the program directory. New encodings may be added to make them available for the import of special fonts.

NOTE:

Fonts in other formats than BDF or PCF cannot be recoded using the installed encoding tables, but they usually contain an ISO-8859-1 or ISO-10646 code page which can be used as-is, or there are Adobe Type-1 glyph names to facilitate recoding to UCS-2. While there are practically no vector fonts which do not provide any of this information, it may be an issue when loading old Windows fonts for other than western languages. Such fonts may be converted to BDF format in order to allow recoding using the installed encoding tables.

8.9 Sounds

All devices except LVIS-3E100 and LVIS-ME200 support the playback of sounds and streaming of MP3 data (web radio). To start and stop sound playback, new actions were introduced. The play action is linked to a sound loaded from disk or defined by a URL. To manage the sounds defined in the project and link sounds to actions, a sound manager dialog is used, similar to the image manager dialog. The dialog can be opened from the *Edit* menu, via the command *Sounds...* or by pressing the *Select...* button on the action property page to select the sound to play back.

To add new sounds, use the **Add File** or **Add URL** buttons, depending on the type of sound you want to add. Sound files loaded from disk are usually in MP3 format, but the configuration software will also allow opening wave files and recode them to MP3 for use on the device. The preferred MP3 type is stereo, up to 128kbps, using a constant bit rate. While other coding variants may work as well, it is generally not useful to load 384kbps variable bit rate or other very high quality formats, since they would only use an unnecessary amount of space and processing time. If problems are experienced during playback, it is recommended to change the coding to a basic format and try again.

Similar to the image manager, the loaded sounds may be replaced by other sounds while keeping the linkage to already defined actions, sounds may be reloaded from disk if the disk file has changed since the last import, and unused sounds may be removed from the project.

There are also buttons to play back the loaded sound on the PC for preview and to export sounds contained in a project to normal files on disk.

8.10 Template-System

This section gives an overview of the template-system and shows the advantages over a completely manual configuration by copy and edit. It is intended as an introduction to the subject of templates and should facilitate the understanding of the technical description in Section 9.15.

8.10.1 Efficient Project Development

The concept of templates and instances provides a powerful tool for cost savings regarding service effort for creation and continued maintenance of customer projects. Especially for projects with recurring graphical elements, like room automation, the effort for creating the project is considerably reduced. References to data points used by the template can be variable, such that each instance can automatically work on a different set of data without any extra work to re-route the data point references manually. For continued maintenance, templates offer much more than a simple copy, because each instance is linked to the template it was created from, and automatically follows any subsequent changes done in the template, even after the instance was created.

Another advantage is that projects using templates are significantly better structured, reducing the effort for maintenance and extensions in the future. Project documentation can be done mostly by documenting the templates, without sacrificing the convenience to have the documentation readily available anywhere throughout the project.

Building a library consisting of templates for various applications can lead to consistent user interface designs across different projects, considerably increasing the recognition value of a company's products.

A typical user interface will consist of a header, footer, navigation area, and a main area containing elements to display information and input data. Using a template instance for each of these areas can guarantee a consistent placement of UI elements across pages, avoiding shifting effects due to inaccurate positioning of for example a navigation button. This leads to a more professional and intuitive interface, with a consistent layout across the individual pages of the project. Local deviations from the template contents are possible for each instance, allowing changes to text, color, position, or visibility, to name a few.

8.10.2 Improve Marketing

Templates not only speed up commissioning and maintenance of individual projects, they can also help in technical sales and distribution to create demo applications. For example, if a room automation project based on templates was done for a customer before, this same project can easily be adapted to match the requirements of a potential new customer, such that it can be used as a demo. Exchanging the entry graphics, the company logo in the header, and adjusting colors and fonts is usually enough to make the demo look like a customer-specific solution.

8.10.3 Corporate Design

Corporate design in a building means specific requirements from the customer regarding the appearance of visible elements in the entire building, including graphical user interfaces for room control. Such requirements are best met by using templates, since they provide the flexibility to deploy a certain look throughout the project easily, without editing a lot of individual components.

Any templates created in this process can be re-used for other projects of the same customer to maintain a consistent cross-project design.

8.10.4 Solving Complex Tasks

Templates can be nested inside each other, which means that a template can instantiate other templates to access their functionality, thereby automatically taking over any changes or improvements done to the instantiated child templates, for all instances of the parent template.

This mechanism can be used to create a type hierarchy, where simple elements are put together to form more complex parts of the user interface, which can in turn be used to create the content of entire pages. On the top level, only one template may need to be instantiated to provide the content for an entire page. Which template to use, would depend on whether the page should be used to control lights, sun blinds, or any other system. Such a page template would contain instances of a header, footer, and navigation template, as well as an appropriate content template (lighting, sun blinds, HVAC ...).

8.10.5 Improved Structure

As shown in the above section, the use of templates can greatly enhance the structure and therefore maintainability of a project. It guarantees that the project follows certain standards, no matter who is building the top level project based on the templates from a ready-made library. Templates can be organized in a folder hierarchy to categorize them into sub-systems. The tree icon of a template indicates if that template is currently instantiated anywhere in the project, such that unused template, which may have been copied over in a batch from a template library can be removed at the end. The context menu of a template instance provides a command to jump to the template associated with the instance.

Each instance maintains a list of all its data point references. When setting up the base paths for a template instance, this list can be consulted to make sure the instance is operating on the desired data set and all required data points could be found.

All templates and their instances are constantly monitored by the configuration software for possible problems, like non existing base paths, or missing data points. Any template or instance that failed one of these checks is marked with a red warning triangle to alert the project engineer.

8.11 Workflow

Even though there are a lot of things to configure, creating a new project really is not that complicated. If this is your first attempt to create a project, it might help to concentrate on one thing at a time to avoid unnecessary confusion caused by switching back and forth between objects and their different property pages.

A simple workflow is suggested below:

- First, build the desired object hierarchy using the tree view at the left. Use the context menus to add, cut, copy, and paste objects and use drag and drop inside the tree view to move existing objects around. It is also possible to have a second instance of the configuration software running and copy/paste objects between the projects. In the property view, keep the **General** property page open and enter names and descriptions for your objects as you create them. Do not get lost in detailed object configuration at this time!
- You should now have an object tree containing at least the hierarchy of menus, menu items, pages, and the basic controls which will be used to display and enter data. All objects should be clearly named. Now is a good time to go through the menus and configure their basic appearance. Select one menu at a time from the tree view and keep the **Common Properties** page open in the property view. For each selected menu, enter a menu title in the *Text* field and select the desired font.
- With the **Common Properties** still open, go through all menu item objects and enter each items text in the text field. Choosing a medium sized font, like the 12x16 ROM font, will make it easier to select the menu item on the touch screen. If the item text does not fit the width of the menu, don't worry about it for now. The required menu width will be set at a later step, when all menu items have their final text assigned. While you are at the common properties page, you may also load an icon for the menu item, which will be shown to the left of the item text.
- Go back once more to the menu objects of your project and open the **Menu/Page** property page. For each menu, set the *Container Width* such that the longest item text fits the menu. If your items use graphical icons as well, you may need to adjust the width reserved to the left of the item text such that the widest icon will fit. In case the menu contains more items than fit on the screen, make the scroll bar wider, so that the user may easily grab and drag it with the index finger. The default width of the scroll bar is designed to look good for menus which do not need a scroll bar.
- Now go through your pages and lay out the controls as desired. While a page is selected, go to the **Common Properties** page to select a background bitmap for the page, or go to the **Color** page to select a background color. That's all you should need to configure for the page itself at this time. Note that each page for which you set a full-page background image will require additional memory (RAM). To place small logos or page headers, use a bitmap control instead and

select a suitable background color for the rest of the page. This will require less memory and be more efficient when switching pages.

- As you lay out the controls, it is a good idea to also set their properties to modify their appearance, since this will ultimately change the size of the controls and may have an impact on their placement on the page. For text and bitmap controls, you should already have the data points connected and the mapping tables filled in, bar and trend controls should have their final settings of scale, tick marks and value range. The individual property pages are discussed in more detail below.
- Add actions, alarm generators, mathematical objects, and other global objects you require to put functionality into your project.
- Fine-tune colors and transparency on the **Color** page and the visibility of objects (layer stack up) by using drag and drop in the tree view.

This is of course not a complete list of things to do, but should give you an idea where to start from a blank project.

If objects or complete pages are copied from one project to another project, the copied objects may contain references to data points or loaded fonts which do not yet exist in the target project. If possible, any missing data points or fonts will be created automatically in the destination project. If this is not possible, for example if an object references a data point from a foreign technology (copy from a BACnet project into a CEA-709 project), these references will be lost. See Section 8.5.3 for details about copy/paste.

NOTE:	<i>For best results, make sure to review Sections 8.2 and 8.3 for details about important mouse and keyboard functions, as well as Section 8.5 about commands in the context menu.</i>
--------------	--

8.12 Common Property Pages

The individual property pages visible in the property view will change according to the currently selected object. Most property pages are directly related to a certain type of object and will be discussed together with the object itself. Some pages are used to set common properties and remain visible all the time. These common pages are explained below:

8.12.1 General

This is the page which should be open while creating new objects in the tree view. As you create new objects, give them a suitable name here, to clearly identify them in the tree view.

NOTE:	<i>The object name specified here has nothing to do with the appearance of the object on the device, if it is a visible object at all. The name on this page is just the name of the object in the object hierarchy. The name should be chosen such that looking at the tree view is enough to understand what an object is used for.</i>
--------------	---

The large area below the name field can be used freely to enter any description which might be needed for other people to understand what this object does, why it is here, how it works, and so on. This can be used to store the projects documentation together with the project, improving reusability and maintainability, in case the project is taken over by another person later on.

The object description is shown in a tool tip when hovering over the item in the object tree.

Additional information shown on this page is the unique ID of the object as well as the estimated memory usage on the target device.

NOTE:	<i>The unique ID of the object is used for identification. If the object records data which is stored in a file (trend log or data log), the file name will include this ID in order to match the data files to the respective controls. The UID is also used to reference other pages in the project (action Show Page).</i>
--------------	---

8.12.2 Common Properties

The **Text** field is used to configure the textual part of an object. This may be just ordinary text or a format string, instructing the object how to format its text output. For menus and items, this field holds the menu or item title as it appears on the device, for controls which display text or a numbers (text, numeric, and date controls) this field contains a format string mixed with additional static text. Since these controls may need a larger amount of text to be entered, a separate property page called **Editor** is available to easily enter and review larger chunks of text.

The **alignment buttons** allow to adjust the alignment of the text inside the display box (left, middle, or right aligned). The **font window** is used to select the font for the text. If other than the built-in fonts are needed, additional fonts can be loaded into the project using the font manager, which can be opened via the **Open Font Manager...** button. The font manager is covered in more detail in section 8.8.1.

Text is used by various objects in different ways, as shown below:

Object Type	Usage of the Text field
Menu	Directly used for the menu title.
Menu Item	Directly used for the menu item text.
Text Control	<p>Format string which may contain one %s placeholder, which will be replaced by a string taken from a mapping table or a connected string data point. For static texts, use a text control and fill in the static text in the text field, without using a %s placeholder. The %s placeholder conforms to standard C rules and may include additional format modifiers between the % and the s:</p> <p><number> A number following the % sign forces a field with for the string of at least the given amount of characters. If the string is shorter, it is padded with white space at the left (right justified within the given field).</p> <p>- A dash preceding the field width means the string should be left-justified inside the field (padded to the right).</p> <p>.<number> A decimal point followed by a number is interpreted as the maximum length of the resulting string.</p>
Numeric Control	<p>If locale dependent formatting is not used, it is a format string which may contain one placeholder for a floating point value, for example %f, %g, or %e. To display a hexadecimal value, the placeholder %x may be used together with the option <i>Convert Data Point Value to Integer</i> which is available on the Numeric Control property page (detailed below).</p> <p>There are a number of format modifiers available for the expert user to tailor the output of the numeric control. Novice users may use the format string wizard property page to construct a suitable format string without knowledge of the details.</p> <p>In case of locale dependent formatting, the C-style format string is replaced by a simple placeholder tag #{value} or #{value:<number>} as in #{value:120}, at the position where the formatted value should appear. The optional value defines the minimum field width of the formatted number in pixels, for alignment purposes. All format related options, including the padding, are set on the number control property page.</p> <p>In addition, number controls recognize a place holder tag of the form #{unit} or #{unit:<number>} like in #{unit:40} and replace it with the effective engineering unit of the connected data point. The optional number defines a minimum width reserved for the unit, to allow table layout of numbers using different units.</p>
Date Control	<p>If locale dependent formatting is not used, it is a date/time format string which may contain a number of different placeholders to display date and time in textual form. The format string conforms to the ANSI C strftime() function and is detailed in the appendix of this document. On a separate property page, a format string wizard is available for the user to create a correct date/time format string without knowledge of the details. Expert users may also look up the available placeholders in a C programmers manual for the strftime() function.</p> <p>In case of locale dependent formatting, the format specifier changes to a different syntax used by the locale-aware date formatting routines. This format can be managed entirely through the UI of the Date/Time Control property page, so the user does not need to be familiar with its syntax. However, it could also be edited manually.</p>
Alarm List	When localized custom date format is selected, the text field is used to enter the custom date format pattern. Otherwise it is unused.

Table 9: Usage of the common text field for visible objects

Also available on the common properties page are fields to enter the screen **position and size** of the selected object, if the object is a control. However, position and size of controls are usually modified directly in the LCD preview using the mouse or the cursor keys. For more information about position and size of controls please see step 5 of the tutorial.

The checkbox **Auto-Resize** is normally turned on, so that controls which support it will resize themselves automatically to accommodate the largest content they need to display according to the text field and the entries in the mapping table. The checkbox may be turned off for some controls to prevent automatic resizing and set the desired size manually. Resizing a control manually in the LCD preview or by changing the width and height properties automatically turns off the Auto-Resize option.

NOTE: *Manual size for text and number controls is most often used when the final length of the displayed text is not known at configuration time, or the selection area of an input control should be larger than the displayed text. Examples for situations in which the final text length cannot be determined at configuration time include number and date controls using **localized formatting**, and text controls receiving content from the network.*

Next to the screen coordinate section is an area where a bitmap can be loaded. A number of graphic formats are supported, for example BMP, JPG, PNG, TIFF, SVG and others. Pressing the button **Select...** opens the image manager dialog, where all images of the project are shown. In this dialog, new images can be loaded, unused images can be removed, and existing ones can be replaced by others or resized, rotated, and flipped as necessary. Graphics import is discussed in detail in section 8.7.

Depending on the object type, the image is used according to the following table:

Object Type	Bitmap is used for...
Menu Item	Icon which is shown to the left of the menu item text.
Page	<p>Page background. The loaded bitmap should match the screen resolution of the device, for example 320x240 pixels. Smaller background pictures are placed in the top left corner of the display. For example a 320x20 bitmap would appear as a page header, while an 80x240 bitmap would appear as a vertical bar at the left.</p> <p><i>On devices supporting true color, the best format for page background images is JPG. If the original image does not fit the screen resolution, it may be resized in the image manager dialog.</i></p> <p>To place smaller static bitmaps anywhere on a page, use a bitmap control and assign the bitmap to the control instead of the page, then move the control to the position where the graphic should be located.</p>
Bitmap Control	<p>Bitmap to show when the control is created. If the control does not have a mapping table attached to it, this is the only image which is shown by the control. It is then similar to a text control containing static text. Bitmap controls may be used in this way to place logos or other graphic elements anywhere on the screen.</p> <p><i>On devices supporting true color, the best format for icons is PNG or SVG, since these formats can support alpha blending, which results in a smooth transition between the icon and the background along the icons edges.</i></p> <p>If a mapping table and a data point are attached to the control, the images from the mapping table may replace the image specified here, when new values are received or selected by the user.</p>
Bar Control	The bitmap is used to draw a moving handle or indicator, at the center point of the border line between the empty and the filled area of the bar (at the edge of the moving bar). Use the bar controls properties to modify the drawing position for the bitmap.
Knob Control	The bitmap is used as a faceplate for the knob control. The control is automatically fixed to the size of the faceplate image. Using the knob control properties, position and radius of the knob need to be adjusted to match the loaded faceplate.
Webcam	The bitmap is displayed centered in the control window when video playback is turned off.

Table 10: Usage of the common bitmap for visible objects

Below the screen coordinates is an area for additional options. The **Data Input Access Level** defines the required minimum access level for data input via the selected object (available for input controls). The access level must be equal to or higher than the level specified here, otherwise the user will see the control but will not be able to select it and input new data.

Simple controls like text, numbers, bitmaps, and vector controls can also be rotated and flipped in X and Y direction using the options on this page. Note that this will affect the entire control. For example, rotating a bitmap control will show all bitmaps from the mapping table rotated accordingly. To rotate individual images, use the options in the image manager.

The **Line Break** option is available for controls which output one or more lines of text, like text, number, or date controls. If the option is set, the control will insert additional line breaks at appropriate positions, whenever a line is too long to fit the control container. If support for complex writing systems is available and turned on, line breaks will be inserted according to the language being used. Even East-Asian languages, which do not contain any white space characters, will be handled correctly. If complex writing system support is unavailable or disabled, new line breaks will only be inserted at white space positions.

NOTE:	<i>Any line breaks which are already contained in the original output are retained. The line break option only inserts additional line breaks as needed. Also, a control using this option will align the text block at the top edge of the control, instead of vertically centered.</i>
--------------	---

The button called **Set as Default** may be used to save the design of the currently selected visible object as the default design for new objects of the same type. If a number of similar objects is to be created, configure the first object, press the default button and then create the other objects.

NOTE:	<i>The set defaults are saved together with the project, so that the same defaults apply the next time this project is loaded. To change the defaults for an object, change the design as needed and press the default button again to update the stored default settings. Note that defaults using custom fonts or images will result in additional references to these objects. Fonts and images can only be deleted from the project when they are no longer referenced by any other object, including default settings for a control.</i>
--------------	---

8.12.3 Editor

The editor property page is used to change control text and font settings. While this can also be done on the common property page for the current project language, the editor page allows to edit text and font for multiple languages at the same time. In addition, it provides a larger text input area to work with.

The list of project languages at the right of the property page can be used as follows:

- **Single select one language:** The selected language is set as the current project language. The LCD preview and the contents of the property pages adjust to the selected language. Any changes made to the text or font settings apply to the selected language only.
- **Select more than one language:** Selecting a range of languages using the shift key as modifier, or individual languages using the control key (standard multi-select method in list or tree views), the current project language for preview remains unchanged. However, any modifications to the text or font settings will apply to all selected languages. If current font or text settings are not the same for all currently selected languages, the font dropdown or text input area will be empty to indicate differing settings.

The copy/paste buttons to the right of the font selection can be used to copy a font selection from one language to one or more other languages without remembering the

selection of the source language and re-selecting the same font for the destination languages.

NOTE: *For date and number controls with locale-aware formatting, the text field contents may be the same for all languages (as in `$(value)`). However, the control may need different fonts to display the formatted content in different languages. Also, if extra text like a value label is added to the value placeholder, it will be different for each language.*

8.12.4 Color

The color property page is used to configure color and transparency for all visible objects. To avoid different color property pages for every object, the colors are set through this common page. Colors which are similar in use appear on the same location, so that multi-assignment of for example a text or a background color to controls of different types is possible. Not all objects support all colors. When an object is selected, it will report its color configuration on this page and set all unsupported colors to *unset* (black cross). To change a color, click on the corresponding color button and select the new color from the dialog.

NOTE: *Most colors are named such that it is clear to which element of an object the color applies, like Text, Background, Selection and similar. Depending on the selected object, the naming of the colors will change accordingly. How the available colors are used by a particular object can be found in the objects documentation.*

The right side of the color page allows setting the current color scheme for multiple objects at once. Applying a color scheme to multiple object means that all colors which are *set* in the color scheme will be changed in the object, if the object supports this particular color, while all colors which are *unset* in the color scheme will remain unchanged.

NOTE: *This allows changing individual colors only, for example you may change the background of all controls on a page without changing any of the other colors*

There are two modes of operation available right now:

- **Apply to all sub-objects:** This mode will apply the color scheme to all sub-objects of the currently selected object. This is used to control the color design of the whole project (root menu selected) or individual pages (a page selected).
- **Apply to all objects of this type:** This mode will search all objects of the same type and with the same parent as the currently selected object, for example if you select an item of a menu and use this mode, you can change the colors of all items in THIS menu, without affecting other menus. If you select a text control on a page, you can use this mode to change the color of all text controls on THIS page, and so on.

NOTE: *If the device firmware is older than 4.2.0, all color settings done on this page will be limited to the 256 color VGA palette, even for RGB capable devices. Starting with firmware 4.2.0 on RGB devices, colors can be selected from an RGB color selection dialog which also includes an input field to set the transparency of the color. This field is available as a numeric value only, where 0 means transparent and 255 opaque. Note that the color setting of fully transparent black (all values 0) has a special meaning. It indicates that this value is unspecified and will be displayed as a black cross.*

For RGB projects, individual colors can be made transparent by using the *Unset* button in the RGB color selector. This is the preferred way to handle transparency in RGB projects, as opposed to the VGA backward compatible way of setting a background color and turning on the *Transparent background* flag to make all pixels using this color transparent (color to alpha function to emulate transparency, as used in VGA projects).

9 Object Description

9.1 Introduction

Every L-VIS project is built using a number of different objects. Each of the objects has a relatively simple function. To build a complex project, the simple objects are put together and organized in an object tree, which is shown in the tree view of the configuration software.

Interactions between the objects are defined by their location in this object tree, for example, a menu item object will always display a menu item on the device, but it depends on the location of the object in the tree in which menu and at which position in the menu the item appears.

There are objects to represent menus, menu items, pages, various types of controls to display and input data, there are objects to perform mathematical operations on data, or generate alarms. The total number of different object types is kept to a minimum, while available objects are designed to be as generic as possible. This allows flexible and creative use of the available objects as the building blocks of more complex applications.

When building the object tree, there are certain limitations in how objects can be connected to each other. Most of the time, this will be clear from the context, for example it is clear that you cannot connect a page object to a menu, since a menu would not know what to do with a page. A menu is used to manage a list of menu items from which the user may choose. Therefore, a menu object will only accept menu item objects as child objects. A page object can then be attached to a menu item object, causing the item to show the attached page, when it is selected from the menu.

NOTE:	<i>Such limits are enforced by the configuration software automatically, so you cannot drop or paste objects into places where they have no meaning.</i>
--------------	--

Some objects may be connected to a number of other objects, for example a data point may be connected to all kinds of controls, but a trend control only accepts input data points, not output data points. Mapping table objects may be connected to text, bitmap, number, and bar controls, but not trend or date controls, and so on. Using the context menu of an existing object to create a new object will automatically show only the kind of objects which can be attached to the existing object.

The individual objects are explained in more detail in the following sections.

9.2 Menu

A menu object is used to manage a collection of menu item objects. On the device, a menu is a window which opens on the left side of the display and shows the items which are

contained in the menu. The menu can handle more items than fit on the screen. If this is the case, a scroll bar is provided for the user to move the contents of the menu up and down to see the hidden items.

NOTE: *Every project always has a menu object as the top level object. This is called the root menu. It must always be there and it cannot be deleted. However, you may adjust its properties just like any other menu.*

Apart from the root menu, the only other places where menu objects may be created are as child objects of menu items. If a menu object is connected to a menu item, selecting the item will show the connected menu. This is called a sub-menu. To navigate back from the sub-menu to the parent menu, the sub-menu must contain a return item, which is a menu item object which has no further objects connected to it (no page and no sub-menu to show).

9.2.1 Menu Properties

A menu consists of a frame and an area containing the menu items. The frame includes the menu title and the scroll bar, whereas the item container is the inside space of the menu, where the menu items are shown. The assignment of colors on the **Color** property page to parts of the menu can be seen in the following table.

Color Name	Element to which the color applies
Text	Menu title text.
Frame	Title background and the frame around the scroll bar.
Scrollbar	Scrollbar, color of the filled bar.
Scrollbar Backgr.	Scrollbar container background.
Background	Background of the menu item area. The background of a menu cannot be set to be transparent.

Table 11: Color assignment for menus

On the **Menu/Page** property page, the width of the menu item area and the scroll bar can be entered. The value is specified in LCD pixel (the full screen width being 320 pixels). The *Bitmap Width* parameter specifies how much space for icons to the left of the menu item text should be reserved, if an item uses a bitmap. This is used to have a vertically aligned text column, even when the icons of individual items do not have the same width.

NOTE: *Even though the bitmap width parameter changes the appearance of menu items, it is a property of the menu because it is the same for all items which are shown in this menu. Moving an item to a different menu will show the item according to the bitmap with rule of the new menu.*

9.3 Menu Item

Menu item objects are used to build the contents of a menu. They are shown in the menu in the order in which they appear in the object tree. On the device, each menu item becomes a selectable area of the menu, which the user may select to open the page or the sub-menu associated with the item.

Each item object can hold zero or more page objects and zero or one menu object, as shown in the object hierarchy diagram. It depends on the connected objects, what will happen when the user selects the menu item.

Pages	Menu	Function
None	None	This item is a return item . When the user selects it, the current menu will close and the parent menu will open. Usually added as the last item of a sub-menu.
1	None	A standard item . Upon selection, the menu will close and the page will be shown.
> 1	None	A multi-page item . Upon selection, the menu will close and the page which was up when the item was active the last time will be shown (the first page, if this is the first time the item is selected). The page flip action touch gesture, and page timeout applies.
None	Yes	A sub-menu item . Upon selection, the connected sub-menu will be shown for further navigation.
1 or more	Yes	A combined item . The connected page(s) will be shown in the background, while the connected menu will open in the foreground for further navigation. This type of item can be confusing for the user and should be used with care.

Table 12: Function modes of a menu item

In the **root menu**, a return item is used in a special way. If a return item is found in the root menu, the device will use it to **connect the system setup sub-menu** to it, making it the **setup menu item**. This is done so that the user may provide a menu item for the built-in setup menu and configure the item through the normal configuration software, just like any other menu item.

NOTE: *If the project does not provide a return item as the last item of the root menu, and the built-in setup menu was not disabled in the project settings, a **default setup menu item** will be created automatically. However, this item will most likely not fit the design of the other items in your root menu, so it is a good idea to always provide this item in your project.*

9.3.1 Menu Item Properties

On the **Common Properties** page, a menu item can be assigned a text as well as an icon bitmap. Obviously, at least one of them should be set to make a usable menu item. If both icon and text are set, the icon will appear on the left side, the text on the right side. If the icon is smaller than the reserved bitmap width of the menu, the icon will be centered inside the bitmap width.

On the **Color** property page, the following colors can be set for the menu item:

Color Name	Element to which the color applies
Text	Menu item text.
Selection Frame	Frame drawn around the selected item.
Icon Bitmap	Color of a monochrome bitmap.
Icon Background	Color of the bitmaps background. This will be the color which is shown transparent in the items bitmap, if the bitmap has no alpha channel or the project is not true color. The background of a menu item is always set to transparent mode.

Table 13: Color assignment for menu items

9.4 Page

Page objects are simple objects to organize the control elements which are visible on the device. When a page object is connected to a menu item, and the item is selected from the menu, the page and all the controls on it will be shown on the display. Previously visible controls will be hidden, but will continue to exist on the device (so that a trend control may

continue to record data, for example). Note that entering or leaving a page can trigger actions connected to that page. See Section 9.9 about action objects for details.

NOTE: *When editing a symbol in L-Studio, the top level object will be a page instead of a menu. This page serves as the drawing surface to design the symbol, and will be used to build a suitable template page when the symbol is instantiated in other symbols or projects.*

9.4.1 Page Properties

On the **Common Properties** page, a bitmap for the page background may be specified. Such a bitmap should match the dimensions of the page, for example 320x240 pixels on an LVIS-3E100. Please also see the information in section 8.12.2 before using a background bitmap. For a single-color background, select the desired color on the **Color** property page.

All other properties of a page object are set on the **Menu / Page** property page.

On this page, the following settings are available:

- *View Timeout:* If this timeout is not zero, and there is no user input for the specified amount of time, the system automatically switches to the next page (provided there is more than one page connected to the same menu item). After the last page was shown, the procedure is restarted from the first page.
- *Page Access:* This defines the minimum access level required to view the page if it is marked as locked/protected (see below).
- *Locked Page:* If this flag is set, the page may only be displayed if the current access level is equal to or higher than the level specified for this page (see above). If the access level is not sufficient, a suitable PIN code must be entered.
- *Invisible Page:* Allows hiding the page on the device, which means that it will not be accessible for the user until this flag is removed. However, the controls on a hidden page work as usual, for example a trend control placed on such a page will still be recording trend data. A hidden page may also be used as the target of a show page action, so that the page may only be visited through this action and not directly via the menu.
- *Default Page:* Mark this page as the default page of the device. The default page is displayed after a configurable amount of idle time and when the device is started.
- *Login Page:* Mark this page as the login page of the device. The login page is displayed when the user switches to a protected page for which the current access level is too low. On the login page, there should be at least one numeric input field to enter a PIN code (write the value to the *PIN Code Enter* register). It is useful to place other information on this page as well, for example the current access level and the minimum access level required to view the protected page. A button to execute the *back one page* action is also useful in case the user does not have the required PIN code and wants to go back to where he came from.
- *About Page:* Mark this page as the about page of the device. This page is shown when the user selects the *About* item in the setup menu or when the device switches to the default page after the idle timeout and there is no default page set.

9.5 Controls

All objects which can be displayed on a page are called **controls**. This is because they are usually used to provide to user with an interface to control a data point value or trigger

actions. Controls are also used to display the current value of a data point and sometimes even display just static content, to enhance the visual appearance of a page.

NOTE: *There are other objects which are used to control or process data point values, which are not documented in this section, for example mathematical objects or alarm generators. These objects are not control objects, because they do not provide a user interface. They have no position and size coordinates or other visual properties.*

Depending on the data point objects connected to the control, the control will either show static content only, show dynamic content based on values which are received from connected input data points, or become selectable and accept new values from the user, which are then written to the connected output data points. Controls which accept user input are called input controls.

The following sections describe each of the available controls and their properties.

9.5.1 Text Control

Text controls are used to display one or more lines of text. The text is entered on the **Common Properties** page in the *Text* area, or the separate full-size **Editor** property page, as explained in the section about the common property pages. Dynamic text from a mapping table or from a connected string data point may be inserted at the %s placeholder, otherwise the text remains static.

On the **Color** property page, the following colors can be set for the control:

Color Name	Element to which the color applies
Text	Text color. This color may be overridden by color specifications from a connected mapping table.
Selection Frame	Frame drawn around the text, if the control is selected. Also used to draw the grid of the drop-down list, from which a new text element is chosen (input controls only).
Background	Color of the controls background. The background of a text control may be set to transparent using the check box next to the background color button.

Table 14: Color assignment for text controls

If the text control is an input control, it may be selected by a touch on the controls text field. A dropdown list will appear, containing all entries of the mapping table which is connected to the control. From this list, the user may select a new entry. The value which is associated with this entry in the mapping table will then be assigned to all connected non-constant output data points.

Instead of opening a dropdown list, text controls may also be used as push buttons, where each press of the button will change the value. This push button mode can be enabled on the **Text / Bitmap / Drawing** property page, which is explained below, since it works the same for text and bitmap controls.

For the L-WEB model, the text control may be set to web browser mode using the option *Web Browser Mode* at the bottom of the property page. In this mode, the L-WEB application will not just display the contents of the text control, but instead try to interpret them as web content according to the following rules:

- **URL:** If the contents have the form of a URL, like <http://www.host.com>, the specified URL will be followed and displayed in a browser window of the size and position given by the text control.
- **<html>:** If the contents start with an <html> tag, they will be parsed as HTML source and displayed in the browser. Useful if a complete HTML page was created in a web design tool and pasted into the text control.

- **<body>**: If the contents do not look like a URL and do not start with an `<html>` tag, they will be used as the body of an empty web page, that is, the L-WEB application will generate the required HTML headers around it, including the `<body>` tag, and fill the empty body with the contents of the text control. This mode can be used to display HTML formatted text without having to provide the full source code of the HTML page.

If web browser mode is enabled, additional options are available to change the way the web content is displayed.

9.5.2 Bitmap Control

Bitmap controls are very similar to text controls, only that they display graphic elements instead of text. Otherwise the same behavior applies for input controls.

A static bitmap may be set on the **Common Properties** page, which is shown when no mapping table is attached to the control or no value was set, so that the control does not know which entry from the mapping table to choose.

The bitmap control will automatically adjust its size to fit the loaded bitmaps, as long as the **Auto-Resize** option is set. If there is no bitmap to display, the control draws a rectangle in the selected frame color instead.

If the control size is changed, the **Auto-Resize** option is turned off, and all image content is automatically adjusted to fit the new control size. Depending on the image, auto-resize of the image content is handled in different ways:

- Images created from imported SVG data will be scaled to any size. If the image has the option **Lock aspect ratio** set, it will be scaled to fit the new control size while maintaining the original aspect ratio. If unset, it will be distorted and scaled to the exact width and height of the control.
- Images from other sources will only be scaled to sizes smaller than the original, if the **Lock aspect ratio** option is **set**. Images which are allowed to be distorted will always be scaled to the exact control size, even if the result is an up-scaled pixel image.

If an image is smaller than the control, either due to an aspect ratio limit or due to not up-scaling a pixel image, it will be displayed centered inside the controls image area.

For best results with manually sized bitmap controls, try to use SVG source data if possible.

NOTE:	<i>Since images are usually not language dependent, the images of a bitmap control cannot be different for individual languages of a project. Other than text and font selection for a text, number, or date control, images remain the same for all languages.</i>
--------------	---

The following colors may be set for bitmap controls on the **Color** property page:

Color Name	Element to which the color applies
Bitmap	Color of the frame which is drawn if no bitmap was assigned, or color in which a monochrome bitmap is drawn. May be overridden by colors from the mapping table. For color bitmaps, this color definition is ignored.
Selection Frame	Frame drawn around the text, if the control is selected. Also used to draw the grid of the drop-down list, from which a new text element is chosen (input controls only).
Background	Color of the controls background, if no bitmap is shown. Otherwise this is the color which should be considered the background of the graphic. If the transparent check box is set, all pixels using this color will be transparent. In true color projects, transparency is usually controlled by the alpha channel of the image.

Table 15: Color assignment for bitmap controls

NOTE:	<i>For best results, use SVG format for icons. This will generate PNG format for your images on the device and embed transparency information as alpha channel. Using this image in true color projects, transparency will be fully automatic based on the alpha channel data. When used in VGA color projects, the image will be converted to VGA palette, all transparent pixels will be set to color index 248 (a duplicate of middle-grey), and the background setting will be color 248 transparent, so that transparency will be fully automatic in this case as well. Also, LWEB clients will make direct use of the SVG data and display the icons in maximum physical screen resolution at all zoom levels.</i>
--------------	--

9.5.3 Push Button

Push buttons are not provided as a specialized control object type, but are implemented by a special mode of operation, which can be activated for text controls and for bitmap controls, to create push buttons showing text or graphic elements using all the features of these controls. Push button mode is activated on the **Text / Bitmap / Drawing** property page using the *Enable Push Button Mode* check box. Once enabled, the mode of operation is selected on the right via three radio buttons. The options are:

- **Send current value:** In this mode, each press of the button will cause all connected output data points to be updated with the current value of the control. The current value is defined by the last value received from any of the connected input data points.
- **Select and send next value:** This mode will instruct the control to find the entry in the mapping table which corresponds to the current value of the control and then select the next entry to determine the new value which is sent out.
- **Select and send previous value:** Similar to the above mode, but the previous entry in the mapping table is selected to determine the next value.

Below the radio buttons is a check box, which, when checked, causes the control to change direction when either the end of the mapping table is reached, that is, it switches back and forth between mode 2 and 3, walking through all available entries in the mapping table (up and down).

NOTE: Obviously, mode 2 and 3 only make sense if a mapping table is attached to the control and the table contains at least two entries. A typical use is a table with two entries and the control in mode 2, **with direction change enabled**. This provides a simple ON/OFF toggle button. To build a push button which only sends out a fixed value when selected, no mapping table is required. Set up the control using static properties and connect the output data point which should be updated with a certain value. Mark the data point to be Constant Value and enter the desired value as the Default Value of the data point.

Push button mode is very similar to normal input mode, except that the control chooses the value to send automatically and that it automatically leaves input mode as soon as the push button is released, whereas normal controls remain in the input mode until another object is touched or the input idle time is reached.

NOTE: As a consequence, a push button mode sends out two updates on a data point which has its update mode set to **Immediate**. The first update is sent immediately when the control chooses the next value, the second update is sent when the button is released and leaves input mode. To prevent this extra update, set the update mode to **Focus Loss**.

9.5.4 Numeric Control

These controls are used to display values as numbers. Most properties can be set on the **Common Properties** page, as described earlier in this document. Similar to text controls, the color of the number, the background, and the selection frame can be set on the **Color** property page, while text can alternatively be edited on the full-size **Editor** property page.

The **Numeric Control** property page provides access to the specialized properties of numeric controls, like the allowed value range, some options, and the actual number format configuration in case locale dependent formatting is used.

For standard number controls, which are not locale aware, a separate property page containing a wizard to generate a suitable C-style format string is available, in case you don't want to enter it yourself on the common properties page.

In the area *Display and Input Range*, the properties concerning the value range for input and display are defined. These values indicate the value range and resolution expected for the control, and are used to estimate the maximum required output width when **auto-size** is enabled. If input is allowed, the control will limit any input to the indicated value range. Received values will be shown unmodified, even if they are out of range.

- **Value Range:** The values entered in these two fields (min –max) define the limits for input of new values. The user cannot input any value below min or above max. Values received from input data points are always displayed, even if they are outside the allowed input value range. However, to estimate the most likely space requirements for the output in case of auto-size, the given value range and resolution data is used to deduce some test-values whose output is evaluated to find the widest output. Since this will not be accurate in all cases, especially for more complex locale dependent formats, it is advisable to set a fixed size for such controls.
- **Input Resolution:** This value defines the amount by which the value is increased or decreased when the + or – buttons are used in the key pad. The resolution value is also used together with the value range to generate test-values for output width estimation, in case auto-size is enabled.
- **Acceleration:** This setting is only used by the older LVIS-3ECTB model, which has a jog dial instead of a touch screen. It defines how much the dial speed increases the amount of value change, starting at the given input resolution for low speeds and increasing gradually as the jog dial is turned faster.

The *Flags* area contains the following additional options:

- **Locale Dependent Formatting:** This option activates locale aware formatting of numbers. The C-style format string is replaced by a simple placeholder tag *#{value}* at the position where the formatted value should appear. All properties regarding the way the value is formatted are set in the *Number Format* area on the right hand side of the property page. Any changes are immediately visible in the LCD preview. See section 9.5.4.1 below for details.
- **Convert Data Point Value to Integer:** This check box is available when C-style formatting is used. It needs to be checked when you want to use an integer format specifier in the format string, like %x or %d. If the format string wizard is used to generate the format string, this check box is set by the wizard as required.
- **Hexadecimal Input Keypad:** This check box can be used together with a %x format to provide a hexadecimal input keypad matching the way the value is displayed. It would be confusing to enter a decimal value and then see it converted into hexadecimal on the control, so the display format and the input keypad should use the same number system.
- **Password Input Keypad:** This option may be used to enter a PIN code. The keypad starts out empty instead of showing the current control value and for each digit entered it only shows a star (*).

9.5.4.1 Locale-Aware Format

When **locale dependent formatting** is used, the *Number Format* options on the right hand side of the property page become available. They are used to set a style and a number system to be used, as well as some additional options to refine the output. The following basic styles are available:

- **Decimal:** Standard decimal number output. Depending on the options, the output can either be fixed point or floating point (see options below).
- **Scientific:** Scientific format, using exponential notation. Depending on the options, the output can either be normalized (always one digit in front of the decimal point), or display a fixed number of fractional digits.
- **Percent:** The given value is interpreted as a factor and shown in percent. The value itself will normally be between 0 (0%) and 1 (100%), but other values are of course possible and displayed accordingly ($1.5 = 150\%$, $-0.3 = -30\%$).

If the project is designed for LVIS-3ME devices and the option **Enable all settings** is activated in the extended project settings, some additional styles become available, which are currently not supported by LWEB clients:

- **Compact Short:** Useful to display large numbers. The value is shown in orders of magnitude, using a configurable number of significant digits. A short locale specific suffix indicating the magnitude is appended, for example *K* for 10^3 , or *M* for 10^6 . A value of 20513867 could be shown as *20.5M*
- **Compact Long:** Similar to the short form above, but the magnitude is appended in its long form, for example *thousand* or *million*.
- **Ordinal:** The value is displayed as an ordinal number, for example 4 could be shown as *4th*, while 3 would be shown as *3rd* (assuming an English locale).
- **Spell-Out:** The value is written out in words.
- **Currency:** The value is shown with the currency symbol of the current locale. Position of the currency symbol relative to the number, as well as any separators are locale dependent as well.

- **Currency ISO:** Similar to the above, but instead of the currency symbol, the three letter ISO code is used (*USD* instead of \$).

Next to the style selector is a field to enter a desired **padding** of the output in pixels. Padding is applied to the left side of the number, such that the specified minimum width is reached. In case the output is already larger than specified, the width is extended to fit the entire output.

Below the style selector is the selector for the **number system** to use. The default setting uses the number system defined by the current locale, so that switching the locale will also possibly change the way the number is displayed. This is the recommended setting for most cases. However, for certain applications, it may be desirable to set a fixed number system which does not change when the locale changes, or is different from the default number system of the current locale.

NOTE:

*The default list of available number systems contains many decimal systems like Western, Arabic, Devanagari, or Thai digits. If the project is designed for LVIS devices and the option **Enable all settings** is activated in the extended project settings, additional systems like Roman, Hebrew, Chinese, or Japanese numerals become available. These systems are algorithmic, and currently not supported by LWEB clients. Also note that some algorithmic systems limit the ability to display certain values, for example, selecting Roman or Hebrew numerals will restrict displayable values to integers only. If a selected number system is unable to display the current value, the control may fall back to western decimal digits.*

Also note that the selected font must provide the glyphs required to display values in the desired number system. A western only font will most likely not contain the required characters to display a number in Thai digits or Japanese numerals. If the locale will be changed during run time and the number system is taken from the locale, it is important that the selected font provides glyphs for all expected languages, or that different suitable fonts are used for each language setting in multi-language projects.

For multi-language projects, be advised that a fixed number system will remain fixed for all languages, since it overrides whatever the current system locale defines. This may or may not be desired. If a fixed number system should only be used for a specific language, the current **Language** system register can be used to show a special copy of the control using the fixed number system, while for other languages the general control is shown, which uses the number system defined by the currently selected language (use visible on condition triggers).

Following the number system selector are the options to refine the output. Not all options are applicable to all styles, so some of them may be unavailable depending on the selected style. Always set the style first, before adjusting the options.

- **Specify significant digits:** This option toggles between definition of *integer and fractional digits*, and definition of the *range of significant digits*. Defining a minimum number of desired integer digits and a fixed number of fractional digits leads to an output where the decimal point is always at the same location. This makes the output suitable for fixed layouts like tabular display of multiple values, but it limits the range of displayable values. Specifying a range for the number of significant digits to display leads to an output where the decimal point changes position or may disappear completely, depending on the value displayed. This makes it unsuitable for fixed layouts, but it provides a very flexible way to display just about any number, no matter if it is a large number like 20136 or a very small number like 0.00124.
- **Use grouping:** Depending on the locale, large numbers may be grouped by certain separator characters for better readability. In many western locales, groups of 10^3 are common, but certain locales may use groups of 10^4 instead.

- **Always add sign character:** This option requests that positive numbers should be prefixed with a + character explicitly, while the default positive number prefix is usually not set (no prefix added).
- **Append data point unit string:** This option will include a suitable place holder tag which is replaced by the effective engineering unit of the connected data point during run time.
- **Unit Width:** Desired padding for the unit string, to facilitate column alignment even when the unit string changes due to selection of a different unit system at run time. The given number of pixels will be reserved for the unit string as a minimum width. Should the unit string be wider, the width will be adjusted to fit the string.

Local aware number controls expect the control text to contain a `${value}` placeholder, which will be replaced by the formatted value. Since each number control only handles one value, there should be only one value placeholder. In case there are multiple occurrences of `${value}` in the control text, each occurrence will be replaced by the same formatted value.

9.5.4.2 C-Style Format

If locale aware formatting is not supported by the target device, or the option is turned off for the control, the standard C-style format is used. A separate *Format String Wizard* property page is available to easily create a correct format string by setting a few options:

- **Style:** The basic style of the value display can be selected from this list. Each style has an example to show how it would look like. The styles *dynamic 1* and *dynamic 2* automatically switch between the floating point format and one of the two scientific formats (1 or 2), based on the value and the selected precision.
- **Field Width:** This value defines the *minimum* desired number of characters to output, including sign, leading zero, decimal point and any other characters which are part of the displayed number. If the number needs more characters to display than specified here, the field width will be extended as needed, so that no characters are cut off.
- **Precision:** This value specifies the desired number of digits following the decimal point. The displayed value is rounded to fit the desired precision. For example, the value 2.86 is displayed as 2.9 if a precision of 1 is used and as 3 if a precision of 0 is used.
- **Zero Padding:** This option causes the converted number to be padded with leading zeros rather than blanks, if the number is shorter than the minimum specified field width. This option has no effect, if the option *left justified* is used.
- **Left Justified:** This option causes padding characters to be inserted to the right of the converted number, so that the number appears left justified within the specified field width. The option has no effect if the field width is equal to or smaller than the space occupied by the converted value.
- **Reserve space for sign:** This option adds a blank character in front of positive values, to act as a place holder for the sign character, which is output as a – if the number becomes negative.
- **Always add sign character:** This option is an alternative to the preceding option and outputs a + in front of positive values, instead of a blank. If both options are checked, the positive sign character is printed.

- **Append data point unit string:** This option will include a suitable place holder tag which is replaced by the effective engineering unit of the connected data point during run time.
- **Unit Width:** Desired padding for the unit string, to facilitate column alignment even when the unit string changes due to selection of a different unit system at run time. The given number of pixels will be reserved for the unit string as a minimum width. Should the unit string be wider, the width will be adjusted to fit the string.

Press the button **Generate Format** to automatically generate the format according to the selected options. The format string will be set as the *Text* property on the **Common Properties** page and the result will be visible immediately on the LCD preview.

NOTE:	<i>It is possible to add static text around the value defined by the format string. If you generate the format string using the wizard, first generate the format string and then switch to the common properties page or the editor page and add the static text before and after the format specifier in the text field. Any text can be entered, with the exception of the percent sign, since this character starts a format specifier. The exact format of this string is described in section 15.8.1 of this manual. To output a percent sign, enter %% in the format string.</i>
--------------	--

9.5.5 Bar Control

The bar control is one of the more complex and versatile control types. In combination with other controls, they can be used for a large number of applications which are not immediately obvious, such as a moving sunblind or an arbitrary area on the screen which can be filled with a variable color (useful to mark open windows in a building, for example).

Bar controls can be used in one of three different modes. The most common one, which also allows the control to be used as input control, is the single bar mode. To create bar charts from a number of connected data points, the stacked or side-by-side modes are available. In these modes, no data input is possible.

The basic application of a bar control in single bar mode uses a scale on one side of the bar, including tick marks. If the value represents a temperature, a small dot is sometimes included at the lower end of the bar, to make it look like a thermometer. In this configuration, the bar control consists of the following elements:

- **Scale:** The numbers on one side of the bar, which are evenly spaced out and calculated to be easy to read. While resizing the bar in the LCD preview, the scale is constantly updated to show the most suitable scale for the given value range and size.

NOTE:	<i>On devices supporting localization, scale labels will be formatted according to the current locale, with the number system set to western digits.</i>
--------------	--

- **Tick Marks:** These are the small indicators next to the bar. They are calculated automatically to be evenly spaced out while representing a round step value between two tick marks. If the scale is disabled, the tick marks may be much closer to each other.
- **Dot:** At the lower end of the bar, a dot symbolizes a thermometer.
- **Frame:** The actual bar which displays the current value has a one pixel wide frame around it.
- **Bar:** Inside the frame is the bar which actually shows the current value. With increasing values, the bar moves up or to the right.

- **Knob:** A bitmap may be used to display a moving knob or handle, which is normally shown centered around the current value, but can also be shifted by a constant offset in X and Y direction.

Position and size of a bar control, as defined on the common properties page, correspond to the frame of the bar control, as described above. It does not correspond to the selection frame drawn around the whole bar control.

NOTE: *To move a bar control in the LCD preview, you must grab it somewhere inside the bar frame, not outside, like the scale or tick marks. Use any of the four corners or the bar's frame to resize the bar, similar to text and number controls.*

9.5.5.1 Properties

On the **Common Properties** page, a font may be selected to be used for the scale and value labels. Also on this page, a static bitmap may be loaded. This bitmap will be displayed at a configurable position relative to the mid-point of the current bar end (the 'fill height' of the bar control), so that the bitmap moves up and down (or left and right) to show the current value.

Color configuration is done as usual on the **Color** property page. The following colors are supported by bar controls:

Color Name	Element to which the color applies
Labels	Numbers of the scale, as well as default color for value labels, if no label color is set in the connected data point.
Lines	Color for separator lines between stacked bars.
Knob Bitmap	Color used to draw a monochrome handle bitmap. Not used if the loaded bitmap is a color bitmap (set line color 2 in this case).
Knob Backgr.	Color of the handle bitmap which should be transparent (background color of the handle bitmap), only used in VGA projects or for color bitmaps which use VGA colors only and do not provide any transparency data.
Selection Frame	Frame drawn around the entire control, if the control is selected.
Frame	Tick marks and the bar frame, as described above.
Filled Bar	Color for the bar representing the value. This color may be overridden by color information provided by the connected data point.
Container	Empty space inside the bar frame, which is currently not filled by the bar color.
Background	Color of the background outside the bar frame. If the transparent check box is set, all pixels using this color will be transparent.

Table 16: Color assignment for bar controls

NOTE: *To change the bar color of an individual bar in bar chart mode, select the corresponding data point and set the **Background** color. To override the value label color for an individual bar, select the data point and set the **Label** color.*

All other properties are set on the **Bar Graph** property page:

- **Value Range:** The most important property on this page is the *Value Range* of the bar. It must be adjusted to match the actual value range you want to visualize. If the bar is used to input values, this range also defines what the user may enter. Values outside the given value range cannot be entered, even using the key pad, which is optionally available for input bar controls. On the other hand, if the actual bar height exceeds the desired value range, the extra values may be entered because the bar may be dragged all the way up to the actual bar top, no matter what the desired value range is.

NOTE:

Set the desired value range, before you determine size and position of the bar, especially if tick marks and/or a scale will be drawn. This is important since the scale which best fits the current size will be determined and drawn as you resize the control. Depending on the value range, the space requirement of the scale and the optimum size of the bar will vary.

- **Knob Offset:** The X and Y values entered here are added to the coordinates of the current center point at the top end of the moving bar (the border line between the filled and empty area of the bar). The resulting position will be used as the center point to draw a bitmap, if specified on the common properties page or through a mapping table.
- **Tick Mark Length:** The length of the tick marks in pixel.

9.5.5.2 Operation Modes

Below the basic properties are a few drop-down selectors to change the operating mode of the bar control. The first selector is the **Bar Graph Mode**, which selects one of the three major modes of operation:

- **Single:** This is the classic mode for a bar control. It displays a single bar, independent of the number of connected data points. If more than one input data point is connected, the bar will represent the value which was received last, no matter from which data point it was received. This type of bar control supports value input, as well as a knob bitmap.
- **Side-By-Side:** This mode generates a horizontal bar chart from the connected input data points. The control draws one bar per data point from left to right, with a configurable spacing between the bars, either in the order the data points are connected to the control, or sorted by value (see extended options below). No data input is possible in this mode.
- **Stacked:** This mode generates a vertical stacked bar chart from the connected input data points. In order to support a mix of positive and negative input values, the control maintains two separate stacks. One stack contains all positive values and grows bottom up, the other one contains all negative values and grows top down. In this mode, separator lines can be added between the stacked bars. The order of bars on the stack can be defined by the order the data points are connected, or sorted by value (see extended options below). No data input is possible in this mode.

Following the basic mode of operation is the selector for the **Bar Color Mode**. This setting controls, how colors are applied when drawing a bar. There are three modes available to choose from:

- **None:** Available for single bar mode, disables all drawing of the bar and the container. Used for bar controls which are used as input overlays, possibly only showing a knob bitmap. This setting has no effect for multi bar modes.
- **Single Color:** The entire bar is drawn in one color. The color is derived from a number of different settings, which are considered in the following order:
 1. If a color map object mapping the *background* color is connected to the data point, the current data point value is translated into a color using this map, and the resulting color is used as the bar fill color.
 2. If no color map is connected to the data point, but a static *background* color is set, it is used as the bar fill color.
 3. If the data point has no mapping table and no background color (X), the bar fill color from the controls color settings is used.

- **Gradient:** The bar is filled with multiple colors, or a color gradient, instead of a single color. For this option to have any effect, a color map object mapping the *background* color must be connected to the data point. The color stops defined in that map will be used to draw the bar. Note that in case the color map uses a discrete mapping, the bar will show discrete color steps as well. For a smooth color gradient, activate the gradient option of the color map object.

The last selector is related to the display of value labels for the individual bars. There are various options to choose from, sometimes depending on the mode of operation:

- **None:** No value labels are added.
- **Center:** A value label is added to each bar, centered inside the bar. If the bar height is less than the text height, the label is not shown.
- **Top:** A value label is added outside the moving end of each bar. If the remaining space between the bar end and the control border is less than the text height, the label will be aligned with the control border and overlap with the bar to a certain extent. For bars growing downwards the moving end is the bottom of the bar. This mode is not available for stacked bars.
- **Base:** A value label is placed outside of the bar, near the bar baseline (the fixed end, as opposed to the moving one). This results in a label placed below the bar for bars growing upwards, and above the bar for bars growing downwards. Again, this option is not available for stacked bars.

9.5.5.3 Flags

Next to the basic bar settings, a number of flags are available to control which components are drawn, and how they are drawn. The following flags can be selected:

- **Horizontal Bar:** The bar control is turned by 90 degrees and the bar moves from left to right (lower values left) instead of up and down.
- **Tick Marks:** Enable or disable drawing of the tick marks next to the bar.
- **Scale Labels:** Enable or disable drawing of the scale labels next to the bar. Note that leaving out the scale labels usually allows for a narrower placement of the tick marks, since less space is required between the tick marks.
- **Flip Side:** This option draws scale and tick marks (if enabled) at the opposite side of the bar (right or bottom instead left or top).
- **Thermometer Style:** This is the option to draw a dot at the lower end of the bar, to symbolize a thermometer. It is usually used together with a vertical bar which displays a temperature. It is only available in single bar mode.
- **Numeric Input Panel:** This option instructs the bar control to open a numeric input panel if the control is an input control and the user touches the bar. The input panel is placed on the screen such that it does not overlap the bar, but as close to the bar as possible. This allows the user to enter an exact value on the input panel or drag the bar as usual. For bar chart modes which do not support value input, this option is not available.
- **Bar baseline at zero:** This option adds a line at value zero and uses this line as the base (fixed end) of the bars. Consider to set this option if the low end of the bars value range is a negative value. In applications where positive as well as negative values are expected for stacked bars, this option should always be set to allow the control to split the values into two separate stacks, one growing from the zero baseline up, and the other one down. If the bar base is not zero and stacking is

enabled, the visual result will be very hard to understand and probably not very meaningful in most cases.

- **Separator lines:** Available for stacked mode, draws additional separator lines between the stacked bars. This can be useful if the bars use dynamic colors and it cannot be guaranteed that two adjacent bars always have colors which are easy to tell apart.
- **Show data point units:** This option is available when value labels are enabled and can be used to add the unit of the data point to the label.

NOTE:	<i>As you may have noticed, not all of the elements of the bar control can actually be disabled. In particular, there is no flag to disable the bar frame. If the bar graph is used in an application where the frame is not wanted, it can still be made invisible. Select the same color for the bar frame as you selected for the background and check the option 'Transparent background'. The frame will now be invisible.</i>
--------------	---

9.5.5.4 Extended Settings

To the right of the flags are a few extended settings, applicable to the multi bar modes.

The **Bar Spacing** can be set for side-by-side mode to control how much space between the individual bars is left empty.

The **Sort** option is available for side-by-side and stacked mode, and defines the order in which the bars are drawn. General drawing of bars depends on the mode of operation. Side-by-side bars are drawn from left to right or top to bottom. Stacked bars are drawn on top of each other, starting with the first bar at the baseline. In case of mixed positive and negative values, two separate stacks are built.

The following choices are available:

- **Position:** Bars are drawn in the order the data points are connected to the control.
- **Descending value:** Bars are sorted by value and drawn starting with the highest value first.
- **Ascending value:** Bars are sorted by value and drawn starting with the lowest value first.

NOTE:	<i>For performance sensitive applications where a lot of updates are expected, the order by position is significantly more efficient than the sorted variants. This is because a fixed order allows the control to refresh only the one bar which needs updating, leaving the rest of the control unchanged. In sorted mode, a change in value of a single bar may change the position of all bars, requiring a complete redraw of the entire contents.</i>
--------------	---

9.5.5.5 Application Hints

A powerful way to use bar controls is provided by colors and transparency. By disabling all the decoration around the actual bar, it is possible to have a rectangular area on the screen, which can be filled with a color or made transparent, depending on the value of a data point. This provides an easy way to selectively hide certain areas of the screen.

Example 1: Define a solid color for the bar and the transparent background color for the container and the frame. Now you can prepare a page background, showing a sunblind which is completely rolled down. Place the bar in front of this image and you are able to cover up parts of the sunblind with the solid bar color, making it look like the sunblind is going up and down. If you switch color assignments, you can use the moving bar to selectively *uncover* parts of the background.

Example 2: In front of the above window showing the moving sunblind, a completely transparent bar control may be placed (both bar color and container color are set to the transparent background and drawing of the bar can be disabled using the 'Invisible Bar' option to increase efficiency). By connecting an output data point to this control, it becomes selectable and accepts user input. Now the user may touch any position on the window, which will result in a value update on the output data point. This value may now be used as a position indicator and the sunblind may be moved to the indicated position automatically, without the user keeping a down or up button pressed all the time.

NOTE: *The static sunblind image in the background may be replaced by a bitmap control with a mapping table to show the current rotation of the sunblind.*

Example 3: A combined display of a current value and a set point can be created by using two bar controls. One is a classic bar control to which the input data point is connected, showing the current value. Next to this bar, a second bar can be placed. Use the layout tools (see section 8.6.1) to make this bar the same with, same height and same baseline position as the first bar. Also, use the same value range, but disable all decorations and make the bar invisible. Load an indicator bitmap, for example an arrow pointing towards the other bar control, using the common properties page. Connect the output data point to the indicator bar to allow defining the desired set point by dragging the indicator bitmap (the arrow) up and down.

Instead of a static bitmap on the common properties page, it is also possible to connect a mapping table containing different bitmaps based on the value. This would allow for example the indicator arrow to change shape or color based on the set value. The bitmap may also be an animated GIF or MNG.

9.5.6 Trend Control

Trend controls record and display data point values over time. In case the connect data point is itself a trend recording or a data point from an existing trend recording is selected, the control does not collect trend data by itself but instead reads the data from the specified trend log data point.

This type of control does not allow input of new values and therefore cannot handle output data points. Only input data points may be connected to a trend control, except the device firmware supports the data point direction change option. In this case, output data points may be connected, but the configuration software will automatically set the direction change option to feed the current value of the output data point into the trend control as an input value.

On the device, the control shows a graph with the value on the vertical axis and time on the horizontal axis. The time axis shows either absolute time or the amount of time which has passed since now, where the rightmost end of the axis represents the present time, with older data to the left (for example 10 minutes ago, 20 minutes ago, and so on). As time passes, the recorded data moves to the left and new data is added to the right. The smaller the time span on the time axis, the faster the curve moves to the left.

The scale labels and tick marks, if enabled, are automatically calculated and laid out. If the page background already provides a graphic representation of the scale and the control is only used to fill in the data, the scale and tick marks may be disabled. The best way to lay out the control is to first enable all required decorations (scale, ticks, and grid) and then change the size and position of the control to fit the scale.

NOTE:	<i>On devices supporting localization, scale labels will be formatted according to the current locale, with the number system set to western digits.</i>
--------------	--

9.5.6.1 Internal Trends

Whenever a normal data point is connected to the trend control, the control itself will record the data received from the data point and store it in the file system when the option *Preserve Data* is set. In this mode, there are a number of settings regarding data recording and storage which are not applicable to external trends, explained in the next section. It is possible to mix internal and external trending in one control.

When recording data locally, the control is able to generate three data recordings and display three curves per connected data point: the minimum value, maximum value, and average value. To do this, the control scans all incoming value updates between two shift operations. From this data, it calculates the minimum value, the maximum value, and a time weighted average value and adds this data to the appropriate data recording.

NOTE:	<i>The average value calculated during each shift interval is the time weighted average of all input values received during that time frame. For example, if the input has a value of 0 for the duration of 9 seconds and a value of 20 for the duration of 1 second, the time weighted average value for these 10 seconds will be 2, because $(0*9s + 20*1s) / 10s = 2$. In contrast, the time independent arithmetic average value would be 10 $(0+20)/2$.</i>
--------------	--

9.5.6.2 External Trends

Since version 4.3.0 of the device firmware, it is possible to use the trend control to display trend data recorded by an external device or trend data source. In this case, the trend control itself does not record any data on its own. Instead, it fetches records from a trend service and displays this data. This is the recommended way to use trend controls. In many applications, trending will and should be done on an external controller device anyways. LVIS or LWEB should act as a client only, fetching the recorded data from the controller to display it in the trend control.

Trend services like these are represented by a special type of data point called trend point. Such a data point may represent a local trend (recorded on the same device, but not by the trend control itself), or data located on a different device on the network.

When such a trend data point is attached to a trend control, the control will scan the trend and display the first curve mentioned in the directory of the trend service. Note that most trend data points represent the recording of only one curve. In case of a trend data point which is known to contain more than one data recording, the trend will show the trended data points as child points. Selecting and attaching one or more of the child points to the trend control will show the corresponding curves. This is currently available for local trends and for remote trends accessed via OPC from a LINUX device whose configuration was imported.

9.5.6.3 Colors

Regarding color, the trend control is one of the more complex controls to configure, since it supports a large number of colors. The following colors may be set for trend controls on the **Color** property page:

Color Name	Element to which the color applies
Labels	Numbers of the value scale and the time scale.
Horizontal Grid	Horizontal grid lines (if the grid is enabled).
Vertical Grid	Vertical primary grid lines (if the grid is enabled).
Second. Grid	Vertical secondary grid lines (if the grid is enabled).
Buttons	Foreground color for button symbols.
Button Backgr.	Background color for button symbols.
Selection Frame	Frame drawn around the entire control, if the control is selected.
Axis	Tick marks and axis for value and time.
Container	Background of the graph area, which is the rectangular area spawned by the value and time axis. Also used for the background of the cursor-box showing time stamp and recorded values.
Background	Color of the background outside the graph. If the transparent check box is set, all pixels using this color will be transparent.

Table 17: Color assignment for trend controls

To set the color of the individual curves, select a connected data point or trend point and switch to the **Color** property page. You can now set the color of all curves which may be generated by this data point:

Color Name	Element to which the color applies
Minimum Value	Color for the minimum curve (if enabled).
Average Value	Color for the average curve (if enabled).
Maximum Value	Color for the maximum curve (if enabled).
Out-Of-Range	Color to indicate that the current value of a curve is out of range. If the value axis is adjusted to fit, the real value will be shown, that is, a limited value axis does not affect the actual data which is recorded.
Background	Color used for area fill, if enabled for this trend curve.

Table 18: Color assignment for trend curves

Apart from the colors and the font for the scale, which can be set on the **Common Properties** page, all other properties are set on the **Trend Control** property page. This page is divided in different sections, which are described below.

9.5.6.4 Time (X) Axis

On the right side, the **Time Axis** is configured. Note that some of these options only apply for data points trended locally by the control while other options apply also for display of external trend data.

- **Time Span:** Select a suitable time span for the initial zoom setting, given in time per page. For example, if you set this to one hour per page, the graph will show the values of the last hour together on one page. For internal trends, this setting will also define the interval at which trend data is recorded. For example, if the control is 200 pixels wide, a time span of 100 minutes will cause the control to record new data once every 30 seconds since one pixel represents a time span of 30 seconds in this case. For externally recorded data, this setting only defines the default zoom level of the time axis.
- **History Buffer:** To keep data which is not visible on the first page because it is older than the given time span, set the desired amount of history buffer. This value is given in pages, that is, if you set the time span to one day and configure one history page, the control will store the values of the last two days. Older data

will be overwritten by new data. For external trend data, this setting has no effect since no data is recorded locally.

NOTE: *For trend controls which are recording trend data locally, the memory requirements can become significant if you use a larger number of such controls and set a high number of history pages. Check the project statistics for more information. In case the control is used to display the contents of a trend data point (local or remote), it only fetches data from the trend service and does not record any data by itself. The control will keep a small amount of data records cached in local memory for faster operation, but the configured size of the history buffer has no influence in this case.*

- **Shift Interval:** The control normally uses the smallest usable shift interval, which is the time it takes for the graph to move by one pixel. The graph cannot be shifted earlier than this, because it needs to move by at least one pixel (can't move by half a pixel). If this interval seems unnecessarily small (each shift will require some processing time), or trend data is recorded locally and you want the control to average data over a longer time frame without changing the default time span, you may set a minimum time here. The control will then use the next higher shift interval which results in an integral shift of the graph (in pixels).
- **Absolute Time:** Draw absolute date and time on the X axis. Especially if the grid is enabled, this will require more processing time since the entire grid and the X-axis scale need to move, but usually absolute time is the preferred setting.
- **Localized Two Line Format:** Available if the **Absolute Time** option is set, and the device supports localization. Draws absolute time on the X axis, but uses two lines and automatically selects suitable labels for each line, depending on the current time span. The shorter intervals, indicated by the tick marks, are labelled in the first line, with coarse labels on the second line. For example, if the tick marks indicate days, the day number would be shown on the first line and the month name on the second line. Labels are localized, showing dates in a suitable and commonly used format for the current locale.
- **Tick Marks:** Enables drawing of tick marks for the time axis.
- **Scale Labels:** Enables drawing of scale labels. Note that scale labels should always be enabled for controls which have the *Absolute Time* option set.
- **Time Format:** Format string used to display non-localized time labels on the time axis. See section 15.8.2 for a complete explanation of this format. The default is usually a good choice. Time labels are displayed only when there is more than one label for one day. Otherwise, time labels will not be used.
- **Date Format:** Format string used to display non-localized date labels on the time axis. Date labels are shown at midnight of each day which has a label drawn. If the user zooms out further, date labels will be shown only for every other day, every start of the week, or every month, as space allows.

9.5.6.5 Value (Y) Axis

On the top left, the **Value Axis** can be configured. The following options are available:

- **Range 1:** Value range for the primary value axis drawn on the left end of the trend graph. This is the default range for the standard zoom setting and may be changed during runtime when the option **Enable Y-Scaling** is set.
- **Range 2:** Value range for the secondary value axis. If enabled, the secondary scale is drawn on the right end of the graph. In case the grid is enabled, the final scale range is adjusted such that a common horizontal grid between the primary and the secondary axis can be drawn.

- **Secondary Scale:** Activates the secondary Y axis. To assign the connected data points to one of the two value axis, select the data point from the tree view and set the desired **radio button** which is shown next to the value range input fields.
- **Enable Y-Scaling:** Enables navigation buttons for the value axis. This option also changes the width of the control box to accommodate wider Y-axis labels, which may be required when zooming or changing the value range of the Y axis.
- **Tick Marks:** Enables drawing of tick marks for the value axis.
- **Scale Labels:** Enables drawing of scale labels. Note that scale labels should always be enabled for controls which have the *Y-Scaling* option set.

9.5.6.6 Value Options

At the bottom left of the property page, it is possible to set options regarding trend curve display. Select the desired data point first to change these options for a particular curve of the trend control. The left dropdown is used to set the desired interpolation method. The following options are available:

- **Staircase:** The current value will be drawn horizontally until the next recorded value is available, causing the trend graph to show a vertical step to the new value. This setting is suitable to record binary or multi state values like window state or room occupancy.
- **Linear:** Selecting linear interpolation will draw straight lines between the recorded values, which is suitable for temperatures or other analog values which do not represent a small set of discrete values.
- **Dots:** The recorded values are displayed as dots, with a diameter corresponding to the selected line width. If multiple dots share the same horizontal position, they are grouped together and the first, last, highest, and lowest point are drawn.
- **Limit Line:** Only available for data points, but not for trend points. The control will fetch the current value of the attached data point and draw this value as a horizontal line across the entire graph. This can be used to add limit lines to the trend graph. No historical data is recorded by the trend control for limit lines.

The dropdown to the right controls area fill and stacking for the selected curve:

- **Standard:** No filling or stacking. The curve will be drawn with baseline zero.
- **Area Fill:** The curve will be drawn with baseline zero, but the area between the zero line and the curve will be filled with the fill color set for the data point.
- **Stacked:** The curve will be stacked on top of the previous curve assigned to the same value axis. This setting also implies area fill, filling the area between the baseline curve and this curve with the fill color of this data point. See the note below for more information on stacking.

The input field **Label** can be used to assign a label to each of the attached data points. This label will be shown in front of the current value in the cursor box.

NOTE: *Stacking is done separately for each of the two possible value axis (primary/secondary). Also, limit lines build their own stacks, again separately for both value axis. In the list of connected data points, it is allowed to mix curves and limit lines assigned to different Y axis as needed to archive the desired Z order. Foreign curves (not on the same axis or not of the same type) will be skipped automatically. It is possible to build multiple stacks on one axis. To start a new stack, set the option to **Area Fill** for the first curve of the new stack, and continue with **Stacked** for the following curves.*

9.5.6.7 Graph Flags

In the center of the property page are some additional **Flags** available, depending on the selected object. With the trend control selected, the following options can be set:

- **Draw Buttons:** This option enables constant display of the controls navigation buttons. If this option is turned off, the navigation buttons are only shown when the user selects the control to navigate. This is usually the desired behavior.
- **Enable Cursor:** This option causes the control to be selectable. When selected, navigation buttons and a cursor appear, which allow navigating through the recorded data (see section 9.5.6.10 regarding trend control operation).
- **Preserve Data:** When enabled, causes the control to store any locally recorded data in non-volatile memory, in order to preserve the data across a reboot or power failure. Saving is done in regular intervals, when enough data was collected, or when the unit is shut down in a regular way and thus has a chance to save all unsaved data before shutdown (not possible during a power failure). For external trends, this option has no effect.
- **Grid:** This option can be used to enable a grid which is automatically laid out to fit the current tick mark spacing and allow easier readout of the displayed data. Available are the choices **OFF** (no grid), **Dotted** (broken lines), or **Solid** (solid grid lines, since firmware 6.2). When using the dotted line style, it is advisable to set the final control size after enabling the grid and choose a size which yields a good looking result for the grid. Enabling the grid together with a secondary value axis may change the effective value range of the secondary axis such that the tick mark spacing is compatible to the spacing on the primary axis and matches the position of the horizontal grid lines.

NOTE: *Even if the **preserve data** flag is set, a trend control may still discard the stored data when it reboots. This is the case when the controls properties were changed such that the new configuration does not match the stored data anymore, for example because new data points were added to the control or data points were removed, changing the type and amount of data recorded. Recorded trend data is also removed from the device, if there is no control matching the recorded data, which is determined by the UID of the control. That is, if you remove a trend control from the project and reboot the device, the data which belonged to the removed control will be deleted from the system.*

With a locally recorded data point selected, the following additional options are available:

- **Draw Minimum:** Enable recording and display of the minimum curve for the selected data point.
- **Draw Average:** Enable recording and display of the average curve for the selected data point.
- **Draw Maximum:** Enable recording and display of the maximum curve for the selected data point.

The **Line Width** setting has two different functions, depending on which object is currently selected in the tree view:

- **Trend Control:** When the trend control is selected, the default **Line Width** for the entire control can be set using the drop-down box. Trend curves which do not specify their own line width will be using the controls default width.
- **Data Point:** When a child data point of the trend control is selected, and the firmware version is set to 6.2 or newer, the drop-down box can be used to set a specific line width for the selected trend recording. To use the default width of the control, select the first entry **Default**. Otherwise select the desired width from the list of available line width settings (1-15).

9.5.6.8 Time Shift

Trend curves may be shifted on the time axis, to compensate for time zone offsets in remote trends, or to correctly display trended historic filter data. Time shift is configured on the **Data Point** property page, because shift settings are stored in the data point object, similar to the color settings for the trend curve. Moving the trend point from one control to another one via drag and drop therefore keeps color and shift settings.

Time shift is configured via an input field specifying a shift value, a unit dropdown to select the unit for the time shift (seconds, hours, days, weeks, months, or even years), and a dropdown to specify the desired direction of the shift (left or right).

9.5.6.9 Trend Data Access

For every trend control which stores recorded data in non-volatile memory, a file is created on the file system of the device which holds the data in binary form. This file is not intended to be used by the end user. Instead, the recorded data is made available in CSV file format, in order to ease processing on the PC.

The files may be downloaded from the device using an FTP connection or via SCP protocol on the SSH port, if the device supports secure connections. Valid user names for the transfer are either admin or operator. The password must be specified as set on the Web-UI of the device. The default password for the admin user is *loytec4u* on devices with recent firmware and *admin* on devices with older firmware. There are two locations to access the CSV data. One location is intended for access by a human user, the other one is intended for automated tools:

- **/data/trend:** This is the location intended for access by a human user. The file name consists of the object name of the control (as seen on the tree view) and the unique ID of the control, as displayed on the **General** property page. This allows the user to easily identify the data file belonging to a certain control on the device.
- **/data/uid/trend:** This is the location intended for access by automated tools. The filename consists only of the unique ID of the control. This allows an automated tool to easily construct a list of files to access, independent of the name of the control, which may not be known and which may change.

Example: A project contains a trend control with **UID 0x122114B1** and name “**Trend 1**”.

The data recorded by this control will be accessible in CSV format under the following file names on the device:

`/data/trend/Trend_1_122114B1.csv`

`/data/uid/trend/122114B1.csv`

Spaces in the object name are replaced by underscores to avoid file names containing spaces, which may be a problem for some FTP clients.

NOTE: *Assign good names to your trend control objects, to make it easy to identify the data files and relate them to the trends which are visible on the device. Consider to add the name of each trend to the page where the trend is shown, so that the user is able to clearly identify each control and its data file.*

If the trend control object name contains non-ASCII characters, the resulting file name will be encoded in ISO-10646 (Unicode) and will be transferred to the client in UTF-8 encoding, according to the FTP protocol extension RFC2640. Clients who support this extension can then display these file names correctly. Most web-browser based FTP clients either detect the UTF-8 encoding automatically or provide an option where the user may set the desired encoding. To maximize compatibility to existing FTP clients, it is desirable to use only ASCII characters for the name of trend control objects.

An easy way to manually access the files on the device is to use the windows explorer or a web browser and type the following into the address/URL field:

```
ftp:<user>@<address>/
```

Replace <user> with either operator or admin and <address> with the IP address or DNS name of the device, for example:

```
ftp:admin@192.168.1.20/
```

You will then be asked for the password and the file system contents will be shown. You can now navigate to the required files and download them to the local hard disk of your PC for further processing.

NOTE: *Since the trend data files are actually symbolic links to a device driver, you cannot directly open the files on the device. Also, the file size may be shown as 0, because it is not immediately known (the data is generated on the fly, while you are downloading it). You may only download the files. Some FTP client software is known to have problems processing symbolic links to files. They tend to display these files as being directories but any attempt to open this directory will fail, since it really is a file. Examples for such programs are FileZilla and Leech-FTP. If you encounter this problem with your FTP client, use another client or fall back to the windows explorer or the command line FTP client (which is known to work as well). If your device supports secure connections, programs like WinSCP may also be used as an alternative to FTP clients. The SCP protocol uses only one port and is therefore easier to handle in environments where NAT routers and firewalls are present.*

9.5.6.10 Operating Trend Controls

Trend controls do not allow data input, but they can be made selectable. If the control is selectable, touching it on the screen will display a vertical line at the touch point and open a cursor box, showing the exact data which was recorded for the selected point in time.

NOTE: *In case the navigation buttons are hidden, the first touch will show the navigation buttons only. A second touch is necessary to show the data cursor.*

The cursor box will show the absolute date and time of the current cursor position, as well as the current value at that time for each of the curves displayed. Each value is printed in the same color as the corresponding curve, to relate the displayed numbers with the curves. If a curve has a label configured, the label text is shown in front of the value.

Use the check boxes in front of each value to enable or disable individual trend curves.

NOTE:	<i>The values shown in the cursor box are always the most recent known values at the cursor position as recorded in the trend log. Changing the interpolation mode changes the way the trend curve is drawn, based on the available data records, but it does not change the values displayed in the cursor box.</i>
--------------	--

The vertical cursor line may be moved by dragging it around or touching another location of the curve. The data shown in the cursor box will be updated accordingly. Using the + and - buttons on the time axis will zoom in and out the current cursor position if the cursor is shown. Otherwise it will zoom in and out the center of the current page.

If the **Y-Scaling** option is set, navigation buttons are shown for the Y-axis as well. They allow the value range to be shifted up and down as well as zooming in and out. The center button has three states:

- **Empty Circle:** The Y-axis is at its default settings, as defined in the project.
- **Circle with Dot:** The Y-axis is in auto-zoom mode. This state is entered when the empty circle button is pressed. Pressing the button again turns auto-zoom off and returns the Y-axis to its default state (empty circle).
- **Filled Circle:** The Y-axis is in a custom state, where its value range was shifted and/or manually zoomed. This state is entered whenever the user manually changes the Y-axis via the up/down or +/- buttons. Touching the button in this state resets the Y-axis to its default configuration (empty circle).

NOTE:	<i>Y-Axis shifting and zooming can be problematic in controls which use a secondary Y axis and horizontal grid lines, because this setup limits the way the two scales can be laid out. Depending on the value ranges used, zooming in or out on the secondary axis may not result in any visible change at first, since the new internal zoom setting does not allow for a different scale layout yet. In this case, keep zooming until the axis switches to a different scale. If such effects are noticeable, consider removing the grid to allow an independent layout of the two Y axis scales for a better zooming experience.</i>
--------------	--

If there is more than one page of data, the user can flip through the pages by moving the cursor to the rightmost or leftmost end, or touching the appropriate navigation buttons.

The control will return to normal operation when another control is selected or the user touches any other location outside the selected trend control.

NOTE:	<i>Leaving navigation mode will return the time axis to the current time. Zoom setting of the time axis and the current setting of the value axis are not changed. Using the navigation buttons located at the center of each axis, showing a small circle, will return both zoom setting and position of the corresponding axis to the project defaults.</i>
--------------	---

9.5.7 Date Control

The date control is a specialized control to display date and time. In most cases, the internal system time data point will be connected to the date control. However, it is possible to assign any scalar data type to the date control. In that case, the received value will be interpreted as the number of seconds since midnight of the first of January 1970 UTC time.

On LVIS-3E1xx devices running firmware 4.3.0 or newer, it is possible to connect an NV data point of type SNVT_time_stamp to the date control. The control can parse the NV data and display the corresponding date and time.

There are different modes of display available for a date control, some are text based while others show a graphical analog clock. Even though one date control can only use one mode at a time, two or more date controls may be used together to display the time as an analog clock and the date as text on the same page.

A font for text mode can be selected on the **Common Properties** page. All other settings can be done on the **Date/Time Control** property page. The basic configuration is done in the *Options* section. The first option is a style selector, offering the following styles:

- **Text:** A textual display of date and time, based on a C-style format string. If selected, a separate *Format Wizard* page will be opened to help generating a valid format string for the control. The format string can also be entered directly on the **Common Properties** or **Editor** page. The exact format to use is described in section 15.8.2 of this manual. Note that this mode is not locale aware. It is limited to the Gregorian calendar and will always result in English names for month and week of day. For locale aware formatting, use the *Locale* style.
- **Simple:** A simple analog clock. Clock hands are displayed as thin straight lines. Suitable for small analog clocks in page headers.
- **Complex:** A different style for the analog clock, using more complex shapes for the clock hands. Suitable for larger analog clocks.
- **Modern:** A new variation of the analog clock, using straight lines of variable line width and rounded caps. This style is only available for LVIS-3ME devices and LWEB-802/803. Suitable for any size of analog clock (adaptive design).
- **Locale:** Text based style available in projects supporting localization, replaces the older C-style formatting. If selected, the format options become available to define the desired output via the property page. See section below for a description of locale aware date formatting.

Below the style selector are some options for further configuration. Depending on the selected style, some of these options may not be available. The following options are related to analog clock styles only:

- **Seconds:** Turns the display of the second hand on or off.
- **Circle:** Adds or removes a circle around the clock face.
- **Labels:** Adds labels to the analog clock.
- **Place labels inside:** Added labels are placed on the inside (on the clock face) as opposed to the outside (around the outer diameter).

The following options apply to all styles:

- **Show UTC:** Turns off the date controls normal behavior of adding the current local UTC offset to the value of the timestamp before display. When connecting the system time register, this option has the effect of showing UTC time instead of local time, since the received time stamp is shown 1:1 on the control. Can be useful to display local time from a remote device, where the remote UTC time and the remote time offset are received separately and added at the control. In this case, the device-local UTC offset must not be added before display of the already calculated local time of the remote device. Also useful for timer applications, such that a value of 0 results in a time of 00:00:00 independent of the devices local time zone offset.
- **Periodic update:** Activates periodic updates of the date control, independent of any data point updates. Without this option, a date control is updated every time a new value is received through one of the connected data points. This can lead to choppy display of analog clocks if updates are not received on a regular basis or at low frequency. Enabling this option makes sure the control is updated once every second, auto-advancing the displayed time by one second each. Incoming

data point updates are only used to synchronize the time displayed on the control with the time received from the data point and do not have any direct influence on the controls update rate.

The last option is actually related to the selected date pattern in a locale aware control and is discussed in the appropriate section below.

On the **Color** page, the following colors may be set:

Color Name	Element to which the color applies
Text	Text color of the clock in text mode.
Tick Marks	Hour tick marks in graphic style.
Clock Hands	Hour and minute hands.
Second Hand	Second hand (if enabled).
Selection Frame	Frame drawn around the entire control, if the control is selected.
Frame	Circle around the clock face (if enabled).
Container	Background of the clock face.
Background	Color of the background outside the clock face or behind the text.

Table 19: Color assignment for date controls

Output data points may be connected to a date control, in which case the date control becomes a date input control. The user may then input a date and time using the keypad. The number of seconds which passed since the first of January 1970 and the given date/time will be written to the connected output data points. This value is suitable to write to the system time set register, to update the current system time on the device. However, note that due to limits of the underlying operating system, the system time of the device may not be set to a date before 1/1/1988.

9.5.7.1 Locale-Aware Format Basics

If supported by the selected model, dates can be formatted according to a certain locale and calendar system. To activate this mode, select the style **Locale**. When activated, the controls text is reset to a default locale-aware format pattern, so make sure to set the style before adding any additional text to the format. There is of course the possibility to undo the style change to restore the control to the previous state, so no information is lost.

Locale-aware format patterns can be configured entirely via the UI of the property page, but can also be edited directly in the text field, which may be useful for copy/paste operations. To allow multi-line date controls, for example showing the date on the first line and time on the second line, more than one date pattern can be defined in one date control. Each pattern starts with a \$ sign and is enclosed in curly braces, like in `${yMMMMeeee}`. Patterns can be embedded at any point in the control text and will be replaced by the appropriate output.

The UI to edit these patterns consists of a list of available date fields from which fields can be added to the current pattern, an area where the current pattern can be selected, new patterns can be added, and existing ones can be deleted, as well as an area listing the fields of the selected pattern with options to add and remove fields, change their order, and select a variant for each field.

For each pattern, the option **Use auto formatter** can be used to select between a *raw date pattern*, where date fields are formatted exactly the way they were specified in the pattern, and a *pattern template*, where only a list of desired date components is given, and an auto-formatter constructs a suitable format based on the current locale. The desired fields are put in correct order, and field separators or text is added automatically as required. Using the auto-formatter is the default, since it is easier to use. If a pattern is a raw date pattern, it contains a hash as its first pattern character, like in `#{#r-MM-dd}` for an ISO8601 date.

9.5.7.2 Available Pattern Fields

On the left side of the property page is the list of available date pattern fields to choose from. Since some of these fields are only available for the auto formatter and others only for raw date patterns, set the desired mode first before adding fields to the pattern. If the mode is changed later on, an unsupported field in the list will be ignored when using the auto-formatter and may truncate the output in case of a raw pattern.

Some fields are available in standard form and as a variant designated with **(Stand-Alone)**. The most important distinction between the standard form and stand-alone form is a grammatical distinction, for languages that require it. For example, many languages require that a month name without an associated day number (i.e. an independent form) be in the basic nominative form, while a month name with an associated day number (as in a complete date format) should be in a different grammatical form: genitive, partitive, etc.

Here is a list of available date pattern fields with a short explanation and the pattern field symbol representing the field in the date pattern string:

- **Calendar Year:** The current calendar year (numeric), as defined by the selected calendar system. This can be for example 2017 for a Gregorian calendar or 29 for the same year in the Japanese calendar. Pattern symbol **y**.
- **Quarter:** Quarter number / name. There is also a **Stand-Alone** variant of this field, which should be used when only the quarter is shown. Pattern symbol **Q** or **q** (Stand-Alone).
- **Month:** Month number / name. There is also a **Stand-Alone** variant of this field, which should be used when only the month is shown. Pattern symbol **M** or **L** (Stand-Alone).
- **Day:** Day of month (numeric). Pattern symbol **d**.
- **Day of Week:** Day of week number / name. If shown as a number, the first day of the week is shown as day 1. In cases where the selected locale does not specify weekday numbers, the formatter may fall back to the abbreviated name. Pattern symbol **e** or **c** (Stand-Alone).
- **Hour (default):** Only available when the **auto-formatter** is used. Selects the preferred hour format for the current locale. Pattern symbol **j**.
- **Hour w/o Period:** Same as the above, but leaves out the day period, if it would otherwise be used by the current locale (10 instead of 10 pm). Pattern symbol **J**.
- **Hour (alternative):** Similar to *Hour (default)*, but selects an alternative format, which is not based on the preferred attribute but the order in which formats are listed (the first entry is used instead of the preferred one). Pattern symbol **C**.
- **Hour [1-12]:** Hour from 1-12. Midnight is 12am, Midday is 12pm. Symbol **h**.
- **Hour [0-23]:** Hour from 0-23. Midnight is 00:00. Symbol **H**.
- **Hour [0-11]:** Hour from 0-11. Midnight is 0am, Midday is 0pm. Symbol **K**.
- **Hour [1-24]:** Hour from 1-24. Midnight is 24:00. Symbol **k**.
- **Minute:** Minute (numeric). Pattern symbol **m**.
- **Second:** Second (numeric). Pattern symbol **s**.

- **Day of Week in Month:** Day of week in month (numeric), indicates the n^{th} occurrence of the current weekday in the current month, as in *2nd Wednesday in July*, or *3rd Monday in December*. Pattern symbol **F**.
- **Day of Year:** Day of the current year (numeric), starting with day 1. Symbol **D**.
- **Modified Julian Day:** This numeric value can be thought of as a single number that encompasses all the date-related fields. Rarely used, symbol **g**.
- **Extended Year:** A single number designating the year of this calendar system, encompassing all supra-year fields (for example the era). In contrast, the standard year will start over at the beginning of a new era. Rarely used, symbol **u**.
- **Related Gregorian Year:** For non-Gregorian calendars, this corresponds to the extended Gregorian year in which the calendar's year begins. Related Gregorian years are often displayed, for example, when formatting dates in the Japanese calendar, as in *2012(平成24)年1月15日*. The related Gregorian year is usually displayed using western digits, regardless of what numbering systems may be used for other parts of the formatted date. If the calendar's year is linked to the solar year (perhaps using leap months), then for that calendar the related Gregorian year will always be at a fixed offset from the normal calendar year. For the Gregorian calendar, it's the same as the standard calendar year. Symbol **r**.
- **Era Name:** Era name for the current date, for example *AD*, as in *2017 AD* of the Gregorian calendar, or *平成*, as in *平成29* (year 2017 in the Japanese calendar). Note that the auto-formatter will automatically add the era name if necessary to disambiguate the calendar year, even if the pattern template only requests the calendar year and not the era explicitly. Pattern symbol **G**.
- **Quoted text:** Available for raw date patterns only, allows to insert any static text between date fields, for example dashes, commas, colons, or white spaces. When inserted via the UI, the given text is automatically enclosed in single quotes, such that characters which would otherwise be interpreted as field specifiers can be used. When editing the pattern directly, it is recommended to quote static text in the same way to avoid conflicts with field specifiers. When using the auto formatter, field separation characters are inserted at the appropriate locations automatically. Characters which are not field specifiers are ignored in this case.

9.5.7.3 Extended Patterns

For LVIS-3ME projects, there are some additional date pattern fields available, when the option **Enable all settings** is checked in the extended project settings. These fields are not supported by current LWEB clients, and therefore break compatibility of the project with LWEB if used. However, they are expected to be available in a future version:

- **Year in “Week of Year”:** Calendar year where the year transition occurs on a week boundary. Can be used in conjunction with *Week of Year*, as in *week 52 / 2017*. Since the last few days of the last week of the year may already be part of the next year, using the standard calendar year for this purpose will result in wrong output (week 52 / 2018, which is a completely different date). Symbol **Y**.
- **Week of Month:** Week number of the current month (numeric), starting with week 1 for the week containing the first day of the month, usually ending with week 5 for the last few days of the month. Pattern symbol **W**.
- **Week of Year:** Week of the year (numeric). Note that the week number may be locale dependent. While many locales use the ISO8601 standard for week enumeration, some countries like the US are still using different week number definitions which may result in different week numbers compared to the ISO norm. To avoid confusion, it is recommended to avoid week numbers entirely.

When used in conjunction with a year number, make sure to use **Year in “Week of Year”** instead of the standard calendar year. Pattern symbol **w**.

- **Cyclic Year Name:** Calendars such as the Chinese lunar calendar (and related calendars) and the Hindu calendars use 60-year cycles of year names. If the calendar does not provide cyclic year name data, or if the year value to be formatted is out of the range of years for which cyclic name data is provided, then numeric formatting is used (standard calendar year). Symbol **U**.

9.5.7.4 Pattern Management

When a new date control is added, it comes with one pre-defined pattern. This default pattern uses the **auto-formatter** and requests the **year, month, day, hour, and minute** fields. Using the US English locale, this results in an output like *Aug 8, 2017, 3:02*, while the same date control will output 平成 29 年 8 月 8 日 15:02 in the Japanese locale with Japanese calendar. Starting from this default, the list of requested date fields can be modified using the controls above the field list. New fields are simply added by double-click on the desired field from the list of available date pattern fields, or selection of the field and click on the + button above the list of currently selected fields after selecting the desired field width (see below).

NOTE:	<i>If the auto-formatter is deactivated, the field Hour (default) has to be removed, as it is only valid in auto-formatted patterns. Direct formatting requires manual selection of the desired hour format (12h or 24h system, 0 or 1 based).</i>
--------------	---

If an additional pattern is required, it can be added via the + button next to the pattern select drop-down at the very top of the property page. The new pattern will be appended to the end of the current control text on the same line. Use the **Editor** property-page to add newlines or other static text as required. The pattern select drop-down can be used to select the pattern to be edited. The drop-down will always show the patterns in the order in which they appear in the control text.

New patterns can also be added at the desired location directly in the text editor, patterns can be moved to different locations or removed entirely. The UI will parse the text and update the property page to reflect the current control contents. If the pattern field syntax is unknown, empty patterns can be placed manually as `${}` (auto-format), or `#{#}` (raw format), and the desired fields can be added via the UI.

9.5.7.5 Field Width

For each of the available date fields, multiple field widths or display variants are available to choose from. For purely numeric fields, there is usually a selection of **numeric** or **2-digit** and sometimes **3-digit** available. The standard numeric variant usually results in the minimum number of digits, while 2-digit will pad the number to always be 2 digits wide (useful for time fields). A special case is the year field, where 2-digit cuts off the higher order digits and outputs the two least significant digits only (*17* instead of *2017*).

If a field can be displayed in textual form, there are a number of different lengths available, like **abbreviated**, **wide**, **narrow**, or **short**. Some of these selections may not be available for certain fields, or may result in the same output, depending on the data provided by the selected locale. In general, **wide** is the most verbose form, for example a complete day of week name like *Thursday*, **abbreviated** results in the most common abbreviation like *Thu*, **short** is an even shorter and probably uncommon version like *Th*, and **narrow** is a really short form, which would be *T* in this example.

To change the field width of a date field after it was already added to the pattern, select the desired field from the list of fields contained in the pattern and choose the appropriate display variant from the drop-down box above the list. The field will be updated and the result can be seen immediately in the LCD preview.

Use the buttons next to the field with selector to add and remove fields and move the selected field to a different position in the list. Note that the field position is not relevant when the auto-formatter is used, as the list only specifies which date components the formatted output should contain. Field ordering and decoration is done automatically.

In the pattern itself, the field width or display variant to use for a certain field is defined by repeated occurrence of the field specifier. For example, a wide day-of-week name is specified as **eeee** while the abbreviated variant is denoted **eee**. Finally, **ee** specifies the 2-digit numeric format and a single **e** stands for the minimum digit numeric format. Since narrow and short were introduced in later versions of the pattern syntax, they use longer sequences of 5 and 6 characters even though they represent shorter output. However, since the field width can be selected via the UI, it is not necessary to be familiar with this syntax, unless the pattern will be entered manually.

9.5.7.6 Padding

Each pattern in a date control can be padded individually to a minimum width in pixels. Use the **Padding to** field to enter the desired minimum width. The two buttons next to the width specify if the padding should be applied at the left or the right side. Choosing the left align button will pad at the right side, such that the date string is left aligned, choosing the right align button will apply the padding to the left side. In the pattern string, the alignment appears as a positive or negative padding value. Positive values are applied at the left end (right aligned content), negative values are applied at the right end (left aligned content). The pattern is separated from the padding value by a **:** character, similar to the way number controls use padding. Example for a 200 pixel left-side padding: `${yMMMMeeee:200}`.

9.5.8 Data Log Control

Data log controls are used to record data based on events, as opposed to trend log controls, which record data continuously over time. The result of a data log is a list of data records which were recorded at specific points in time. Each data record is composed out of one or more visible elements to display the recorded data, like text, number, bitmap or date controls.

NOTE:	<i>This control is not supported by LWEB-802/803 clients. In many cases, it can be substituted by an alarm log, displayed in an alarm list control. If possible, do not use this control.</i>
--------------	---

The individual data records are created in the folder called *Data Fields* which is part of every data log control. To add a new log record, select *Add Record* from the context menu of the Data Fields folder. Then add the visible elements required to compose your data record to the new folder and attach the data point you want to record. As you lay out the visible elements to build a nice looking data record, the control adjusts its line height to match the area required by your data record.

You will notice that each added data record already comes with a trigger object which has similar properties than a math object. Connect the data points you want to monitor to determine when to store this data record and enter the required formula on the property page of the trigger object. Then select a trigger mode from the available choices. Refer to section 9.14 for a complete description of trigger objects.

NOTE:	<i>If the trigger mode is set to any value update, there is no additional condition to enter in the formula field, since the trigger will fire as soon as any of the connected input data points receives an update.</i>
--------------	--

It is possible to add as many data record layouts as needed and define a trigger condition for each of the data records individually. To work on the layout of a specific data record, first select the record in the tree view to make it visible in the LCD preview. You can then continue to work on the record in the LCD view.

Properties specific to the data log control are configured on the **Data Log Control** page. The options currently allow to define if the control should save the recorded data in persistent memory, how often it should save the data and the total number of records to save (when this number is reached, older data is overwritten with newer data). It is also possible to enable or disable the navigation bar at the bottom of the control and set the size of the buttons.

As with trend log controls, the recorded data is available in CSV format for download via FTP. See the section about trend data access for detailed information about the retrieval of this data from the device.

The following colors may be set for data log controls on the **Color** property page:

Color Name	Element to which the color applies
Highlight	Highlight color of 3D frames.
Shadow	Shadow color of 3D frames.
Horizontal Grid	Grid lines between data records.
Buttons	Button graphics.
Selection Frame	Frame drawn around the entire control, if the control is selected.
Background	Color of the background. If the transparent check box is set, all pixels using this color will be transparent.

Table 20: Color assignment for data log controls

9.5.9 Schedule Control

The schedule control is a specialized control to serve as a schedule configuration editor for local and remote scheduler objects. Starting with version 4.2.0 of the device firmware, there are two different UI modes available for this control.

9.5.9.1 Standard Mode

In standard mode, the control displays a list of daily schedules in a very compact format, suitable for small screens. However, for advanced configurations it may be difficult for the user to determine the resulting overall time table for a certain date, since the individually displayed daily schedules must be combined, based on date patterns and priority, to determine the final events happening on a certain day. In cases where only the basic weekday schedules and a very small number of exception schedules are used, the standard UI mode may be appropriate.

NOTE:	<i>In this mode, it is possible to connect more than one schedule data point to the same control. In this case, the title line will allow the user to select the schedule which he wants to view or edit.</i>
--------------	---

Please see section 7.2.1 for details about operating a schedule control in standard mode on the running L-VIS device.

9.5.9.2 Extended Mode

In extended mode, the control provides a calendar and event based view of the schedule data. There are different pages of information, like for example a weekly overview, a time table for a certain date, a list of scheduled events for a certain date, or a list of all events currently stored in the schedule. This mode is suitable for complex schedule configurations, possibly using LWEB-900 to set up a master schedule.

The user will always see exactly which values are scheduled on a certain date, since the control combines all the available daily schedules with their date patterns and priorities to determine the resulting effective schedule. Also, changing the schedule is based on events,

similar to appointments in a calendar, with an effective date (most often repetitive), a start and end time, a value, and a priority.

NOTE: *In this mode, only one schedule data point can be connected to the control. This is because the UI does not provide a way to display the name of the currently selected schedule or switch between multiple schedules in the same control. To view and edit multiple schedules, please use one dedicated control for each schedule. If there are multiple data points connected to the control, the extended user mode option will be deactivated and cannot be activated until the excess data points are removed from the control.*

Please see section 7.2.3 for detailed information about how to operate a schedule control in extended UI mode on the running device.

9.5.9.3 Font Settings

When operating the control in standard UI mode, it is usually required to load a user defined font. The smallest built-in font is too small to operate the control on the LCD and the medium size built-in font is too wide to be useful for many applications.

Therefore the configuration software automatically searches the list of loaded fonts for a suitable font and, if none was found, automatically loads a font which should be a good choice to get started. A specific font may be selected or loaded any time on the **Common Properties** page.

NOTE: *In extended UI mode, the control will try to keep all touchable areas large enough to be usable, even if the selected font is small. In this mode, it is therefore possible to use the smallest built-in font to save space on the screen and enable the user to see as much information as possible, especially on the smaller L-Vis devices.*

9.5.9.4 Schedule Control Properties

The properties to set on the **Schedule Control** property page define the basic appearance of the control and control access to various operations which modify configuration data. If a local scheduler data point is connected to the control, its configuration will be shown in the preview as well. If more than one local scheduler point is connected, select the desired scheduler point in the tree view to see a preview of the corresponding configuration data.

From a dropdown list, one of several date formats can be selected:

- **ISO-8601:** This option selects the ISO-8601 specification to display dates, which is a four digit year, a two digit month, and a two digit day, all separated by hyphens (YYYY-MM-DD).
- **DIN-1355:** This option selects the German DIN-1355 specification to display dates. In this format, the order is day, month, and year, separated by decimal points (DD.MM.YYYY).
- **US Format:** This option selects the US date format, which uses the order month, day, and year, separated by slashes (MM/DD/YYYY).
- **Locale:** Available on devices which support localization. Date and time is formatted according to the current locale. This is the recommended setting for multi-language projects, unless a unified language independent format like ISO-8601 is to be used for all project languages.

The following options concerning the controls appearance can be set:

- **Width of scroll bars:** This number specifies the width of the scroll bars, if enabled. The width should be chosen such that the bars are easily selectable by the user (touch and drag) but not consume too much space.

- **ISO-8601 week:** This option instructs the control to adhere to the ISO-8601 specification to display the days of the week in the left column. ISO-8601 defines Monday to be the first day of the week, so the day list will start with Monday. In countries which do not use the ISO norm, turn this option off so that the day list starts out with Sunday as the first day of the week.
- **Compact header:** This option causes the control to generate a one line header, placing the scheduler name left aligned and the effective period (if enabled) right aligned. If the font is too large or the scheduler names are too long, there may not be enough space and the effective period must either be disabled or shown on a separate line.
- **Enable extended user interface:** This option is available if the device firmware is 4.2.0 or newer and there is only one data point connected to the control. Activating this option sets the control to extended user mode.

The following options are available when the control is in standard mode:

- **Scroll daylist:** This option enables a scroll bar for the list of days (left column). If the number of total days is known and the control is sized such that all days are visible, the scroll bar may be disabled to save space in horizontal direction.
- **Scroll time table:** This option enables a scroll bar for the time table (right column). Since the number of entries in this table is usually not known, it is a good idea to keep this option checked. However, special applications may benefit from the possibility of removing the scroll bar and gain even more space.
- **Don't show effective period:** This option can be used to hide the effective period altogether. Useful to either save space or to make sure that the effective period is never modified through this control.

The following options are available when the control is in extended UI mode:

- **Hide priority in event list:** This option will hide the priority in the event view to simplify the view and gain more space for the value and date pattern names.
- **12 hour time format (am/pm):** Changes the display of time from 24 hour to 12 hour format. Not used when localized date format is selected.
- **Use am/pm as prefix:** Available when 12 hour mode is enabled, tells the control to use the strings for AM and PM as a prefix instead of a suffix. This option is required for some languages to yield correct 12 hour time representation.
- **Use year/month/day suffix:** This option is available when the date format is set to ISO-8601. It changes the date format to add the localized strings for year, month, and day as suffix to each date component, replacing the hyphens. This option is required for some languages to yield commonly used date representations, for example Japanese: 2011 年 12 月 14 日
- **Use hour/minute suffix:** This option is available when the date format is set to ISO-8601. It changes the time format to add the localized strings for hour and minute as a suffix. Used for some languages to yield commonly used time representations, for example Japanese: 5 時 24 分
- **Use event colors for value:** This option changes the way the event colors are used to draw the weekly overview. Please see section 9.5.9.5 about the two different modes.

- **Follow current date:** This option causes the control to switch back to the weekly overview and show the current week, after the device was idle for some time. The idle time is controlled via the setting '*Dim backlight after...*' in the project settings, since after this time, the device dims down the backlight and goes into idle mode. Also, this option will cause the control to follow the current date while the device is idle, meaning that the weekly overview page will be updated to show the current day whenever the date changes (for example at midnight).
- **Use simplified priority scheme:** This option hides the numerical priority values from the user and instead provides a drop-down list of four fixed levels to choose from (weekly/low/normal/high). The mapping is optimized to work together with schedules generated by LWEB-900, which also uses distinct priority levels for the entered events. Even in a standalone system, it may make sense to use this option to make the UI easier to read for users which are not familiar with the numerical priority scheme of the underlying network technology. When this option is turned off, the technology specific priority numbers will be used directly, so the user needs to know how they work. Usually, lower numbers mean higher priority, and the allowable range is different for each network technology (0 to 15 for BACnet, 1 to 127 for CEA709, 255 as a special priority to indicate use of the weekly schedule).
- **Prefer event list over time table:** This option changes the navigation on the weekly overview page. Normally, touching a day column switches to the time table of the selected day, whereas touching the day column header shows the event list. Turning this option on, means that the user prefers to go directly to the event list. Therefore, touching the day column will show the event list, whereas touching the day column header will show the time table.

The following colors may be set for schedule controls on the **Color** property page:

Color Name	Element to which the color applies
Text	Color for all text areas of the control.
Highlight	Highlight color of 3D frames.
Shadow	Shadow color of 3D frames.
Horizontal Grid	Grid line color.
Scrollbar	Scroll bar and scroll arrow graphics, only available when the control is used in standard UI mode.
Frame	Color for various flat (non-3D) frames, only available when the control is used in extended UI mode.
Buttons	Foreground color of button symbols.
Button Backgr.	Background color of buttons.
Event 1	First color to use for graphical display of events (extended mode).
Event 2	Second color to use for graphical display of events (extended mode).
Event 3	Third color to use for graphical display of events (extended mode).
Selection Frame	Frame drawn around the entire control, if the control is selected. Not used when the control is in extended mode.
Header	Background color of the header lines and color for deactivated schedules (day names in the left column) and menu items. Only available in standard UI mode.
Dialog Backgr.	Background color for the event dialog. Choose a color on which your text color is readable and which is different from the container color. Usually a color slightly darker than the container color works best. Only available in extended UI mode.
Container	Background of the day list and time table.
Background	Color of the remaining background (only a one pixel wide frame around the control, where the selection frame will be drawn when the control is selected). If the transparent check box is set, all pixels using this color will be transparent.

Table 21: Color assignment for schedule controls

9.5.9.5 Event Colors

The extended UI mode uses up to three different colors to display individual events graphically in the weekly overview. There are two different ways these colors can be used, depending on the setting of the option *'Use event colors for value'*. The two coloring modes are explained below.

When the option *'Use event colors for value'* is enabled, the three available event colors are assigned to the available value presets in a round robin fashion. The first value will use event color 1, the second value will use event color 2, and the third value is drawn in event color 3. If there are more values available, the color assignment will start at color 1 again (value 4 uses color 1 and so on). This mode is especially suitable for schedules where events do not normally overlap each other or overlapping events typically also use different values. Also, this mode works best if the number of distinct values is small, for example occupied/unoccupied/override or other small enumerations. It may cause confusion in schedules where arbitrary discrete values are entered, like for example a temperature set-point. In such a system, adding a new value which is lower than another already used value, the color assignments will change. In this case, the option should be turned off to use the standard color mode (see below).

When the option is not selected, the colors are used such that events of different priority use different colors. All events are internally sorted by priority and then drawn in groups of same priority events. For each new group, a new event color is used. If more than three different priority levels are used on the same day, the event colors will be re-used in a round robin fashion. This mode is especially suitable in situations where a large number of

different values are used and frequently changed, for example when a temperature set-point or other discrete value is scheduled. The color selection in this mode will cause all values scheduled at standard priority to be drawn in event color 1, with colors 2 and 3 only used for values scheduled at the same day at higher priorities. In case these events overlap the lower priority events, the overlapping will be clearly visible because the color of the higher priority event is different. For schedules which use a small number of fixed output values, for example occupied/unoccupied/override, this mode may cause confusion because 'unoccupied' may be drawn in different colors on different days, depending on the priority, when the user would expect the same color to be used consistently for the individual enumerations.

9.5.9.6 Access Levels

The **Access Control** property page is used to define access levels for various operations which may need to be restricted because they modify schedule configuration data. These restrictions apply in addition to the general required access level for the control, as set on the **Common** property page. The following operations can be protected individually:

- *Change time of existing events:* This operation allows the user to change the time (hour and minute) at which a value preset is scheduled. The entry in the time table must already be there, since this operation only allows modifying the time for existing entries.
- *Change preset selection:* This operation allows the user to change the value preset which is scheduled at a certain time. The entry in the time table must already exist, the user may select a different preset from the list of available presets, for example change the air condition from economy to comfort.
- *Edit values of existing presets:* This operation allows changing the physical values which are assigned to existing value presets. The user may for example change the light level assigned to the bright setting from 80 to 100 percent.
- *Define new presets:* This operation allows the user to define new value presets, if the scheduler unit controls only one value.
- *Add and remove entries:* This operation allows the user to add and remove entries in the time table. Also bound to this operation are the related operations to clear entire days, copy them to other days, as well as enabling and disabling individual days.
- *Change effective period:* This operation allows the user to change the effective period of the schedule, thereby enabling and disabling not only single days but the entire scheduler unit.
- *Edit day patterns of exception days:* This operation allows write access to the contents of the calendar which is connected to the scheduler unit. The user may add, remove, and edit entries in the calendar which define the days on which a certain exception schedule is active. Also linked to this access right is the setting of the exception day priority.

To change the minimum required access level to perform one of these operations, select the operation from the list and select the desired access level from the dropdown box below the list. Keep in mind that the minimum access level specified for the entire control on the **Common** property page must be satisfied in addition to the levels specified here.

9.5.10 Alarm-List Control

The alarm list control is a specialized control to display a list of alarms from an alarm server or alarm client data point. It is possible to connect more than one alarm data point to

the same control, in which case the control will display a list of all alarms from all connected points in chronological order.

The control always displays the newest alarm at the top with older alarms following further down. At the bottom of the control is a navigation bar which displays the number of pages the list currently uses and allows to navigate through the pages.

Alarm entries which may be acknowledged are shown with a small ACK button at the right end of the alarm entry. Pressing this button will send an alarm acknowledgment back to the reporting alarm server and change the color of the button. Once the server received the acknowledgment, it may send an updated alarm status and the button or the entire entry may disappear, depending on the configuration of the control.

9.5.10.1 List Entry Setup

The default information displayed for each alarm list entry is the date and time the alarm was first reported, the type of alarm, the current state of the alarm and, in the second line, an identification of the alarm source as well as a description and possibly the value which caused the alarm. The layout and display of the various information fields can be freely configured using the two lists on the **List Entry Setup** property page.

The list at the very left contains all available information fields and, at the end of the list, a special item to insert a line break as well as one item to insert custom text. The buttons between the two lists are used to add, remove, or change the order of items in the right list. The list to the right defines the final layout of the entries in the alarm control. While changing the display list, the effect can be seen immediately on the LCD preview.

Depending on the underlying network technology and the fact if the alarm was reported by a local or by a remote alarm server, not all of this information may be available, so that entries may be missing some of the information. Some information items, like the acknowledge time and source or the clear time become available during the life cycle of an alarm. If not yet applicable, such fields may be shown as dashes.

9.5.10.2 Date Format

The control supports a number of different date formats, used to display time stamps. The following choices are available from a dropdown list:

- **ISO-8601 date format:** This option selects the ISO-8601 specification to display dates, which is a four digit year, a two digit month, and a two digit day, all separated by hyphens (YYYY-MM-DD).
- **DIN-1355 date format:** This option selects the German DIN-1355 specification to display dates. In this format, the order is day, month, and year, separated by decimal points (DD.MM.YYYY).
- **US date format:** This option selects the US date format, which uses the order month, day, and year, separated by slashes (MM/DD/YYYY).
- **Localized long:** This option is available on devices supporting localization. It selects a relatively verbose format, including abbreviated weekday name, month name, numeric day of month, year, and time, as applicable for the current locale.
- **Localized short:** As above, but with reduced information. Usually includes an abbreviated month name, day of month, as well as year and time.
- **Localized numeric:** Further reduced format, purely numeric. Includes day, month, year, and time in the appropriate order, using appropriate separator characters.

- **Localized custom:** The format can be freely specified using the text input field on the *Common Properties* or *Editor* property pages. See Section 9.5.7.2 for detailed information about the date pattern syntax used to specify a format pattern.

9.5.10.3 Options

The following options can be set on the **Alarm Log Control** property page:

- **Read only (no ACK buttons):** This option, when checked, instructs the control never to show ACK buttons. It is intended for users who want to create a ‘view only’ alarm list, with no possibility to acknowledge alarms. While a similar effect may also be obtained by setting the controls access level (for example to level 15), the result is different. Using access control, the ACK buttons will still be visible, so there will be less room to show other information. Also, a user with a high enough access level will still be able to acknowledge alarms in this case.
- **Enable Navigation Bar:** This option enables the navigation bar. This will usually be enabled to allow the user to navigate to older entries of the list, which do not fit on the first page.
- **12 hour time format (am/pm):** Changes the display of time from 24 hour to 12 hour format.
- **Use am/pm as prefix:** Available when 12 hour mode is enabled, tells the control to use the strings for AM and PM as a prefix instead of a suffix. This option is required for some languages to yield correct 12 hour time representation.
- **Use year/month/day suffix:** This option is available when the date format is set to ISO-8601. It changes the date format to add the localized strings for year, month, and day as suffix to each date component, replacing the hyphens. This option is required for some languages to yield commonly used date representations, for example Japanese: 2011 年 12 月 14 日
- **Use hour/minute suffix:** This option is available when the date format is set to ISO-8601. It changes the time format to add the localized strings for hour and minute as a suffix. Used for some languages to yield commonly used time representations, for example Japanese: 5 時 24 分 23 秒
- **Keep Alarm History:** If this checkbox is set, the control will keep alarms even if they are already cleared and also store the list in non-volatile memory, so that it is not lost when the device restarts. In this mode of operation, the control can function as an alarm log, showing older, already processed alarms in addition to the currently pending alarms.
- **Max. Age of History Entries:** This input field specifies the number of days the occurrence of an alarm must lie in the past, before the entry is removed from the list. This applies only to alarms which are already cleared and are not pending acknowledgment (either not acknowledgeable or already acknowledged).
- **Max. Number of History Entries:** This input field specifies the desired maximum number of entries in the alarm list. If there are inactive alarms in the list which exceed this number, they may be removed by the control at the next chance (usually when a new alarm comes in) to make room for new entries. However, like the age limit, this limit does not apply to currently pending alarms, which are always visible as long as there is enough memory available on the device.
- **Height of Navigation Bar:** This input field specifies the desired height of the navigation bar, in pixel. Higher bars mean bigger navigation buttons.

- **Minimum width for date/time:** Set the minimum width for formatted time stamps. Especially useful for localized date formats, since these may vary in length. Output is not truncated in case it is wider than the specified width. Shorter output is padded with empty space.
- **Floating point value field precision:** This input field specifies the number of significant digits to print for alarm values.

9.5.10.4 Colors

The following colors may be set for alarm list controls on the **Color** property page:

Color Name	Element to which the color applies
Text	Color for the text printed by the control.
Highlight	Highlight color of 3D frames.
Shadow	Shadow color of 3D frames.
Horizontal Grid	Grid line color.
Buttons	Button graphics and page information in the navigation bar and text color for the label of the ACK button.
Button Backgr.	Background color of the acknowledge button.
Active Text	Text color for entries which are currently in alarmed state and either don't need acknowledgement or are waiting to be acknowledged.
Active Backgr.	Background color for above entries.
Not ACKD Text	Text color for entries which are no longer in alarmed state but still pending acknowledgment .
ACKD Text	Text color for entries which are in alarmed state but already acknowledged .
ACKD Backgr.	Background color for above entries.
Selection Frame	Available for older firmware only. Color of the frame drawn around the entire control, if the control is selected. Not used by recent firmware.
Container	Background color of the alarm entries in normal state.
Background	Color of the control background. If the transparent check box is set, all pixels using this color will be transparent. Note that this color is only visible in areas where no alarm entries are drawn.

Table 22: Color assignment for alarm list controls

9.5.10.5 Filtering

If multiple alarm server data points are connected to the control, it is possible to set a filter during run time to only show entries from a certain alarm server. To control the filter, connect a normal input data point to the control and use it to set a value between 0 and the number of connected alarm server data points. Value **0** turns the filter off, and the control shows all entries. All other values set the filter to only show entries originating from the corresponding alarm server data point. Alarm servers are numbered starting from **1** at the top end of the data point list in the tree view, not counting the filter data point.

9.5.10.6 Actions

Actions connected to the alarm list control are handled in different ways, depending on the selected action trigger. Actions which trigger **on value update** are executed whenever a *new entry* is added to the alarm list. All other actions, which trigger on user input, are executed when the user *interacts with entries* in the alarm list.

Actions may use a value condition to restrict their execution. The value condition for actions is used in the same way as the filter mechanism described above. Actions with value condition = **1** only trigger for alarms originating from alarm server **1**, and so on.

NOTE:	<i>Execution of value update actions is independent of the current display filter. That is, a value update action for new entries from alarm server 2 (value condition = 2) will trigger even if the current display filter shows alarm entries from server 1 only. Touch actions are of course inherently dependent on the display filter, since the user can only touch entries which are currently displayed.</i>
--------------	--

9.5.11 Vector Drawing Control

The vector drawing control is similar to a bitmap control (see section 9.5.2). The main difference is, that the control displays a vector drawing object instead of a bitmap. The drawing can be created from within the configuration software and is limited to simple objects. More complex drawings can be created by combining multiple drawing objects into one larger object using collection objects (folders). To create a new drawing object, select *Add Drawing* from the context menu of the page and choose the desired drawing type from the list of available choices:

- **Lines:** This adds a line drawing to the page and enters drawing mode. Please see section 9.5.11.1 below for detailed information on how to use the drawing mode.
- **Rectangle:** This adds a new rectangle to the page and draws it according to the selected properties, as detailed below.
- **Circle:** This adds a new circle to the page and draws it according to the selected properties.
- **Polygon:** This adds a new closed polygon drawing. Drawing mode is similar to line drawing (see also section 9.5.11.1) except that the drawing consists only of a number of points instead of a number of lines. This makes a difference when editing the drawing later on. Polygon drawings support shapes which have crossing lines.

The properties of the drawing object are set on the combined **Text / Bitmap / Drawing** property page. For the visual appearance, the following properties are available:

- **3D Outline:** Available for rectangles and circles, defines that the object should be drawn with a 3D outline. Line color 1 is used as the highlight color, whereas line color 2 is used as the shadow color, similar to other controls supporting the 3D effect.
- **Filled:** Available for rectangles, circles, and polygon drawings, defines that the object should be filled with the color specified as the container color, rather than the background color. To draw a frame only, leave this option unchecked and make sure the transparent background option on the **Color** property page is checked.
- **Line Width:** Defines the line width used to draw the lines of a line or polygon drawing and the outline of a rectangle or circle.
- **Corner Radius:** Available for rectangles, defines the desired corner radius for a rounded rectangle. Leave this value at zero to disable rounded corners. Note that this value defines the radius of the outer edge of the rectangles outline. The resulting inner radius depends on the selected line width.

The following colors may be set for drawing controls on the **Color** property page:

Color Name	Element to which the color applies
Lines	Line color of lines and normal frames.
Highlight	Highlight color of 3D frames.
Shadow	Shadow color of 3D frames.
Selection Frame	Frame drawn around the entire control, if the control is selected.
Container	Used to fill rectangles and circles, if the fill option is set. This color may be replaced dynamically by colors in the attached mapping table
Background	Color of the control background. If the transparent check box is set, all pixels using this color will be transparent.

Table 23: Color assignment for vector drawing controls

For data input, the push button mode can be used. This mode works the same as for text and bitmap controls (see section 9.5.3 for a description of the push button mode). From the attached mapping table, drawing objects use the assigned color to fill rectangles, circles, and polygons, or draw the lines of a line drawing. This can be used to create drawings which change their color based on the controls value.

NOTE: *For polygon drawings, the active area to touch the control is limited to the shape of the polygon. This includes all areas which would be drawn in the container color if the polygon is filled.*

For input methods which exceed the possibilities of push button mode, overlay the drawing control with the required type of data input control, for example a number or text control, and use this control to handle data input.

NOTE: *Vector Controls are the recommended way to define clickable areas on a page which uses a background graphic. In this case, the pages background already provides the graphic representation of buttons, lamps, building floors, or other elements, which should be selectable by the user. Using vector controls with transparent border and container, regions of any shape may be defined on the page. The control itself usually has an action connected to it, to jump to a new page or update a data point.*

9.5.11.1 Drawing Mode

Rectangles and circles always fill the entire control. To modify their shape or position on the screen, just change the size or position of the control as required. To edit a line or polygon drawing, a special drawing mode is required. This mode is automatically entered for a newly added line or polygon drawing and can be reactivated later at any time by selecting the item *Edit Contents* from the context menu of the control, or by pressing the key **e** in the LCD preview when the vector control is selected.

While in drawing mode, the control occupies the entire screen. New lines or points may therefore be drawn anywhere on the page. Start and end points of lines or line segments are indicated by small squares, to make it easier to find and operate on these control points. When leaving drawing mode, the control will be resized to the smallest possible outline around the drawing.

The following simple operations are currently supported in drawing mode:

- *Add new lines (line drawing):* To add one or more lines, left-click at the desired start position of the line. Release the left button and drag the line to the desired end point. To restrict the angle of the new line to multiples of 45 degrees, hold down the **shift** key. Left-click at the desired line endpoint to finish the current line and start a new line at the same point. Move to the next endpoint and repeat. If no further line segments are needed, right-click anywhere to cancel the current unfinished segment.

- *Add new points (polygon drawing):* To add a new point to a polygon, left-click at the desired position of the point. The new point will be inserted between the last and the first point of the drawing and the lines will be drawn according to the new shape. To add more points, repeat the process as needed. Press the right mouse button to end point insertion mode. See also the operation ‘*Insert points*’ below.
- *Move points:* Start and end points of a line drawing and points of a polygon drawing may be moved after they were drawn. Move the mouse near the point to move. The cursor changes shape to indicate that a moveable point was detected. Now press *and hold* the left mouse button while dragging the point to its new location. Holding down the **shift** key while dragging the point will apply angle locking similar to adding new points. Release the mouse button when done.
- *Remove lines (line drawing):* Lines may be removed from the drawing by a right-click anywhere on the line.
- *Remove points (polygon drawing):* Points may be removed from the polygon by a right-click on the desired point.
- *Append to an existing line (line drawing):* To start a new line segment at the start or end point of an already existing line, hold down the shift key while doing the left click. This will cancel point move mode, which is normally active near an existing line endpoint. Once the starting point was set, release the shift key, unless you want to use the 45 degree angle locking.
- *Insert points (polygon drawing):* New points can be inserted along a line between two existing points of a polygon by pressing and holding the left mouse button on the line to which a new point should be added. The new point is inserted and can be moved to the desired position immediately by dragging with the mouse and releasing the left mouse button when done. See also ‘*Move points*’ above.
- *End drawing mode:* To exit drawing mode, right-click anywhere on the page, except over a line or a point (see the remove functions above), or press the **Esc** key. The control will leave drawing mode and be resized to the smallest possible outline. It can now be treated like any other control and can be moved to its final location on the page.

9.5.12 Analog Meter

The analog meter control displays a numeric value by a needle moving over a scale, similar to the way a traditional moving-coil ammeter works.

On the **Common Properties** page, a font may be selected to be used for the scale labels. Similar to other controls, color configuration is done on the **Color** property page. The following colors are supported by analog meter controls:

Color Name	Element to which the color applies
Labels	Scale labels and tick marks.
Highlight	Outer left frame of the scale.
Shadow	Inner right frame of the scale.
Needle	Instrument needle or pointer.
Low Range	Scale color for the low value range.
Mid Range	Scale color for the mid value range.
High Range	Scale color for the high value range.
Selection Frame	Frame drawn around the entire control, if the control is selected.
Cover	Needle cover (round cap covering the needle base).
Background	Color of the background. If the transparent check box is set, all pixels using this color will be transparent.

Table 24: Color assignment for analog meter controls

All other properties are set on the **Analog Meter** property page. The basic setup of the analog meter is done on the left side of the property page. It includes the following options:

- **Total Range:** Defines the total value range of the meter.
- **Mid Range:** Defines the middle range or desired range of values. This is used to determine the scale colors.
- **Angle:** The opening angle of the meter scale. This can be anything from a few degrees up to almost 360 degrees. Especially when scale labels are displayed, start and end of the scale should not be too close to each other, so a maximum of around 350 degrees is advisable. The typical angle for ammeters is 90 degrees.
- **Rotation:** Clockwise rotation of the meter. This may be used to change start and end position of the scale as needed. Try 90 degrees opening angle and 45 degrees rotation for a typical instrument.
- **Counter clockwise operation:** The meter operates counter clockwise. For a typical configuration, this means higher values on the left and lower values on the right end of the scale. Increasing values cause the needle to move counter clockwise in direction.
- **Draw tick marks:** Add tick marks to the scale.
- **Draw scale labels:** Add scale labels.
- **Scale on inside:** Scale labels and tick marks are drawn on the inside of the scale, towards the needle center.

NOTE: *On devices supporting localization, scale labels will be formatted according to the current locale, with the number system set to western digits.*

On the right side of the property page, some extended settings can be changed to further modify the look of the control:

- **Needle Thickness:** Defines the width of the needle in pixel.
- **Scale Thickness:** Defines the width of the scale itself. Can be set to 0 to disable drawing of the scale (for meters with only tick marks and labels).

- **Needle Cover Size:** Diameter of the small round cap covering the needle base. Can be set to 0 to disable the needle cover.
- **Tick Length:** Length of the tick marks in pixel. Can be set to 0 to disable tick marks (same as disabling the tick mark option in the basic setup).
- **Line Width:** Width of the scale outline. Can be set to 0 to disable the outline.
- **Secondary Ticks:** Number of additional tick marks without label, to be drawn between two major tick marks (with label). Can be set to 0 for no additional tick marks or to any other value making sense for the value stepping of the labels. For example a label stepping of 0,5,10,15,... could use 4 additional tick marks, a stepping of 0,10,20 could use 9, 4, or 1 additional tick, depending on how far the major ticks are apart (depends on the font size).

NOTE:	<i>Unlike most other controls, width and height of the analog meter control define the outer boundaries into which the control is fitted. Depending on the settings, the actual size and position of the scale may vary. For this type of control, it is therefore advisable to set up all display options first and position the control on the screen afterwards.</i>
--------------	---

9.5.13 Knob Control

The knob control displays a numeric value as a rotating knob. Unlike the analog meter control, a knob can be turned by the user to change the value. The knob features built-in knob designs, including the knob and the faceplate, as well as the option of loading an image to use for the knob and a separate image for the faceplate. The faceplate can be back-lit, such that dots or other position markers light up as the knob is moved. Using a transparent color for the labels and tick marks, they can be back-lit as well.

Due to the complexity of this control, it is only available on LVIS-3ME models and LWEB.

9.5.13.1 Layer Structure

Knob controls are internally constructed from multiple layers, which are stacked on top of each other. Understanding the layer structure can be important to understand how the various configuration options can be used to create a number of different visual effects.

The layer structure from bottom to top:

- **Background:** The first layer is the background color. Depending on the configuration of upper layers, it may never be visible or play an important role.
- **Knob Background:** The next layer is the knob background. It is drawn over the general background, exactly where the knob will be. This color may be visible through transparent areas in the knob, for example LED cut-outs.
- **Backlight:** This layer consists of a circular light strip of configurable radius, width, and color. There are different modes in which the light strip can be drawn. Usually it is drawn in a way to indicate the current knob position, but it can also be a static light strip. For the light strip to be visible, cut-outs need to be placed in the faceplate or the knob, such that the backlight may shine through.
- **Faceplate:** On top of the backlight, the faceplate is drawn. This layer consists either of a single configurable color, or an image. The faceplate covers the entire control. In case of an image, the image usually already contains tick marks and labels as needed. Parts of the image may be transparent for the backlight to shine through and illuminate indicators or other decorations. Suitable ticks and labels, as well as cut-outs for the backlight may also be added to the faceplate through various configuration options. Note that ticks and labels are drawn onto the faceplate in a way that the faceplate becomes transparent if the color of the ticks

and labels is transparent. This way, the backlight can illuminate ticks, labels, as well as any other cut-outs.

- **Rotating Knob:** On top of the faceplate, the rotating knob is drawn. There are various ways to draw the knob, ranging from none at all (already part of the faceplate image), a knob image, or a predefined knob available in different styles like brushed metal or a plastic-like knob with selectable color. In case an image is provided, it can optionally be rotated to indicate the current knob position, but for many knob designs, the knob image itself actually does not need to rotate.
- **Indicator:** The last layer is a position indicator, which is drawn onto the knob or the surrounding faceplate. It may be turned off in case the knob image already contains an indicator and is rotated. Different shapes and styles are available to choose from. Indicator size, color, and offset from the knobs outer edge are also configurable.

9.5.13.2 Basic Configuration

On the **Common Properties** page, a font may be selected to be used for the scale labels. Note that only user fonts can be used for this control, so the ROM fonts will not show when selected. Also on this page, a **Bitmap** can be selected for use as a faceplate. If a faceplate image is loaded, the control will adjust its size to match the size of the loaded image and draw the image instead of just using a single faceplate color.

NOTE: Without a faceplate image, the control must be resized manually to fit the knob and any decorations around it. The **Auto-Resize** function, available for text and number controls, is currently not supported by knob controls.

Similar to other controls, color configuration is done on the **Color** property page. The following colors are supported by knob controls:

Color Name	Element to which the color applies
Labels	Scale labels.
Tick Marks	Tick marks.
Indicator	Fill color for the position indicator.
Range Start	Backlight color at range start.
Midpoint	Backlight color at the configured midpoint value.
Range End	Backlight color at range end.
Faceplate	Color of the faceplate, unless a faceplate image was selected.
Knob Color	Color of the knob, if the selected knob style allows configurable colors.
Knob Background	Color behind the knob, visible through transparent areas of the knob.
Background	Background color, visible though transparent areas of the faceplate.

Table 25: Color assignment for knob controls

Colors which are not needed can be unset (basically set to transparent). Here are a few suggestions and tips regarding color configuration:

- A knob with transparent outside can be archived by setting the background and the faceplate color transparent. Note that this configuration cannot use the backlight to illuminate ticks or labels. However, LED cut-outs can still be placed inside the knob. For this configuration, the knob background color is important, as it defines the color of the unlit LEDs.
- A knob with a fully opaque faceplate color, tick color, and label color, and without any LED cut-outs outside the knob can have the background color unset, as it will never be visible.

- A control with a fully opaque knob, with no LED cut-outs inside the knob may have the knob background color unset, as it will never be visible.
- In conjunction with the indented style, choosing a half transparent mid-grey as the indicator fill color creates an overlay effect, where the software drawn indicator blends nicely with the underlying knob image.

All other properties are set on the **Knob Control** property page. The basic setup of the control is done on the left side of the property page. It includes the following options:

- **Range:** Defines the total value range of the knob, from start to end position. Note that it is valid to set the start value larger than the end value to get a reversed knob. Currently, the knob always turns clock-wise from start to end position.
- **Positions:** Number of discrete positions. This setting defines the number of ticks from start to end position. Note that this includes both end positions. A range of 0-100 in increments of 20 yields 5 intervals and therefore 6 positions total: 0, 20, 40, 60, 80, and 100. It can be set to 0 to let the control calculate a suitable number of positions automatically.
- **Snap:** This option causes the knob to snap into place at the nearest position when the user releases the knob after input. Used in conjunction with a small number of discrete positions, visualizing the states of a multi-state data point, where only a certain set of values is acceptable.
- **Midpoint:** Defines a value which can be considered a neutral position of a two-way knob. Most often this will be located half way between start and end position, but it can also be used for other purposes. It influences the way the backlight is drawn. If unused, it may be set outside the knobs value range.
- **Gradient:** This option controls if the backlight will be drawn in solid colors or as a gradient. Depending on the midpoint setting and the backlight mode, different effects can be archived, which are explained in more detail later on.
- **Angle:** The amount by which the knob may be turned. This can be anything from a few degrees up to almost 360 degrees. Especially when labels are displayed, start and end position should not be too close to each other, so a maximum of around 350 degrees is advisable. The typical angle for a rotating knob is around 275 degrees.
- **Rotation:** Clockwise rotation of the knob start and knob end position. At 0 degrees, the knob position will be facing up when turned half way from start to end, which is the typical way a real knob would be mounted on a control panel.
- **Center Offset:** Offset of the knob center in X and Y direction, relative to the center of the control. At 0/0, the knob will be centered in the control window. When using faceplate images where the knob is located off center, or when the knob has a small opening angle and needs labels only on one side, the knob center may be shifted to the desired position using these values.
- **Knob Radius:** The desired radius of the knob. This will define the outer edge of the knob, to which a number of other elements are relative. When using a faceplate image with integrated knob, this value must be adjusted to fit the image.
- **Show Ticks:** Draw tick marks onto the faceplate. Turn off when using a faceplate image which already contains ticks or when no ticks are desired. Note that for calculation of the active area for input, tick mark dimensions may still need to be entered to match the ticks on the faceplate image.

- **Show Labels:** Draw labels onto the faceplate. Turn off when using a faceplate image which already contains labels or when no labels are desired. Note that if the label area of a faceplate image should be usable for input, a font of similar size than the one used on the faceplate image must be selected, such that the control can determine the outer boundary of the labels on the faceplate image.

NOTE: *On devices supporting localization, scale labels will be formatted according to the current locale, with the number system set to western digits.*

On the right side of the property page, the knob can be further refined and adjusted to match the desired look. The following sections explain these settings and their effect in more detail.

9.5.13.3 Knob Style

This section defines the visual appearance of the knob. There are a few options to choose from. The available list of predefined styles may depend on the firmware version for which the project is designed.

- **None:** No knob is drawn. Useful if a faceplate including the knob image is used.
- **Image:** A user supplied image is used for the knob. The desired image is selected in the bottom right corner of the property page. From this image, a circular area of the specified knob radius located at the center of the image is used to render the knob. Note that the image width and height must be at least twice the size of the desired knob radius in order to cover the entire knob surface.
- **Metal:** A brushed metal knob.
- **Color:** A plastic knob which can be set to a user defined color.

To the right of the style selector is the option **Rotate Knob**, to enable knob rotation. Availability of this option depends on the selected knob style. Currently, only the image knob may be rotated.

NOTE: *In cases where a faceplate image including the knob is used, select the **Color** knob at first, and set a half transparent knob color. This knob overlay will help you adjust offset and radius of the knob in the basic settings, to match the location and size of the knob in your faceplate image. When done, change the knob style to **None**.*

9.5.13.4 Indicator

This line defines the appearance of the position indicator. From the drop-down box, one of a few available shapes can be selected. In case of a rotating knob image which already contains an indicator, the option **None** can be used to turn off the software generated indicator. To the right of the drop-down, the **indicator size** can be set. Depending on the selected shape, either one or two values must be specified to define the indicator dimensions. To complete the indicator setup, two additional options need to be set:

- **Offset:** This value defines the location of the indicator center point relative to the outer edge of the knob. Negative offset values move the indicator inside the knob; positive values will result in the indicator being shifted out.
- **Indented:** When set, the selected shape will be drawn with a shadow to create an indented effect. When turned off, the indicator shape is drawn in a flat style using just the indicator color.

9.5.13.5 Backlight

This line defines operation of the backlight. Using the drop-down list, the basic operation of the backlight can be set. There are various settings to choose from:

- **OFF:** No backlight. Use this setting if the backlight effect is not used.
- **Static:** A fixed band ranging from range start to range end.
- **Band Left:** Backlight from range start, up to the current knob position.
- **Band Middle:** Backlight from **Midpoint** to the current knob position.
- **Band Right:** Backlight from the current knob position to range end.
- **Point:** Backlight around the current knob position.

Once the basic mode of operation is selected, the back-lit area must be defined such that any cut-outs in the faceplate, like LEDs or transparent tick marks, are covered. To optimize performance of the control during input, the backlight strip should be configured as small as possible to avoid unnecessary drawing operations. The back-lit area is defined by two options:

- **Width:** Desired width of the backlight strip.
- **Offset:** Location of the strip center, relative to the outer edge of the knob. Negative offset values move the strip inside the knob; positive values move the strip out.

When used to illuminate software-drawn LED cut-outs only (see next section), width and offset can be set to the same values used for the cut-outs to illuminate just the required area. In case of a faceplate image with transparent sections, try to start with a narrow backlight strip in static mode, set the offset such that the backlight is located at the center of the area to be illuminated, and then increase the width gradually until all transparent areas are covered. Then change to the desired mode of operation.

NOTE:	<i>All modes except Static will usually extend the illuminated area at the beginning and end of the light strip to avoid partially illuminated tick marks, labels, or LED cut-outs. That is, the light strip will change its length only in increments of one position width as the knob is moved. Transition from one position to the next happens half way between positions. There are only a few cases where the light strip starts and ends exactly at the indicated position. For example when the LED cut-out mode is set to Band and both tick mark and label color are fully opaque.</i>
--------------	---

The three colors related to the backlight strip are used to color the backlight depending on two additional settings: **Midpoint** and **Gradient**. Various combinations are possible and lead to different results. Note that this is about the color of the backlight at a certain position, if it is drawn at this position. The range in which the backlight is actually drawn is controlled by the backlight mode described above.

- **Gradient OFF, Midpoint unused (out of range):** Single color **Range Start**.
- **Gradient OFF, Midpoint in range:** Color **Range Start** from range start to midpoint, color **Range End** from midpoint to range end.
- **Gradient ON, Midpoint unused:** One color gradient, from range start to range end, blending from the start color **Range Start** to the end color **Range End**.

- **Gradient ON, Midpoint in range:** Two color gradients. One gradient running from range start to midpoint, blending from color **Range Start** to color **Midpoint**. A second gradient running from midpoint to range end, blending from color **Midpoint** to color **Range End**.

9.5.13.6 LEDs

This line defines cut-outs for the backlight which can be added to the faceplate, such that the backlight can shine through them. This option can be used with the software drawn faceplate as well as image faceplates, in case they do not provide the necessary cut-outs.

- **None:** No additional cut-outs. Use this setting if the faceplate image already contains the necessary cut-outs, or if no cut-outs are desired.
- **Band:** A circular contiguous strip from range start, to range end.
- **Rectangular:** A separate rectangular cut-out at every knob position.
- **Round:** A separate circular cut-out at every knob position.

Next to the drop-down, the cut-out dimension can be specified. Depending on the selected shape, either one value (width or radius) or two values (width and length) must be set. The **Offset** value specifies the location of the cut-out center relative to the outer edge of the knob. Negative offset values move the cut-outs inside the knob, positive values move them out.

NOTE:	<i>The color of non-illuminated cut-outs depends on their location. Transparent areas outside the knob (holes in the faceplate) will uncover the controls background color. Transparent areas inside the knob (holes in the knob) will uncover the knob background color. This results from the layer stack up of the control, as described in the first section.</i>
--------------	---

9.5.13.7 Ticks and Labels

Setting the option **Show Ticks** in the basic setup adds software-generated tick marks to the faceplate. The following options are used to configure their appearance:

- **Tick Length:** Length of the primary tick marks.
- **Line Width:** Width of the primary tick marks.
- **Clearance:** Space between the knobs outer edge and the tick marks.
- **Secondary:** Number of secondary tick marks between primary ticks. This setting does not change the total number of ticks; it only turns a certain number of ticks into secondary ticks. Secondary ticks are shorter, thinner, and are never labeled.

Setting the option **Show Labels** in the basic setup adds labels to the faceplate. A user loaded font needs to be selected for the labels, as the knob control **cannot use ROM fonts**. By default, a label is placed at every primary tick mark position. Note that it does not matter if the tick marks are actually displayed or not. Their position is still calculated and used to place the labels. The following settings influence the way labels are rendered:

- **Label Skip:** Skip this number of primary tick mark positions before placing the next label. This setting can be used to remove unwanted labels which would normally be placed. Note that this setting is related to the setting of secondary tick marks. Both settings can be used to remove unwanted labels. Note that they will influence each other, since only primary tick marks are taken into account for label placement in the first place.

- **Label Space:** Additional space to insert between the outer tick mark end and the label. A reasonable distance between the ticks and the labels is calculated automatically from the font dimensions. If required, the label space setting can be used to adjust the distance. Note that adjustment is possible in both directions. Positive values will insert additional space; negative values will bring the labels closer to the knob.

9.5.13.8 Input

There is currently one option affecting the way the knob behaves during data input:

- **Input by knob rotation only:** This option prevents the knob from aligning its position with the touch point when input mode is entered (see also Section 7.2.1 about knob control data input). Instead, the knob keeps its current rotational position and only moves relative to the movement of the touch point. This prevents sudden changes in value as it mimics the behavior of a physical knob. This mode of input may be desirable for a volume control, whereas the default mode, which allows to set a value directly by just touching the appropriate position, would be better suited for a mode selector, a set-point dial, or lighting control.

9.5.14 Pie Chart Control

The pie chart control is used to display a pie or donut chart based on a number of connected input data points. The control draws one slice for each connected data point, using the background color of the data point object. The size of each slice is proportional to the data point value and chosen such that the sum of all slices fills the entire pie (for a standard configuration). Data points with values less than or equal to zero are not shown and their value is not used when calculating the total sum.

Slices are normally shown clock-wise in the order in which the data points are connected to the control, but can also be sorted by size, starting with the largest slice.

Each slice can be labelled with the absolute data point value, as well as a relative value, with 100% representing the full pie. Labels can be placed inside or outside of the slice. On the outside of the pie, at the border between slices, additional labels can be added to show a running sum as absolute and/or relative value.

A variable size border line can be drawn around the slices to separate them, and a round center piece can be cut out to create a donut chart. A label showing the absolute total value of the pie can be placed in this center piece.

The entire pie can be rotated to change the 100% position (normally on top, if no rotation is configured) and the border lines between slices can be extended outside the pie to create tick marks, useful when showing the running sum labels.

Due to the complexity of this control, it is not available on older and already discontinued LVIS models like LVIS-3E1xx or LVIS-ME2xx.

9.5.14.1 Properties

Pie chart controls provide a number of properties to configure their visual appearance and function to fit various applications. Label font, as well as control position and size can be set on the **Common Properties** page. Color configuration is done on the **Color** page.

NOTE:	<i>To set the fill color for a particular slice of the pie, select the corresponding data point and change the Background color. For individual pie slice label colors, select the data point and change the Label color. For all other color settings, select the pie control itself.</i>
--------------	---

The following colors are supported by the pie chart control:

Color Name	Element to which the color applies
Labels	All label text (slices and sum labels). Note that the slice label color can be overridden for each individual slice by setting the label color for the connected data point.
Lines	Border around the pie slices, including pie outline and tick marks.
Cover	Fill color for the round center piece. Can be set transparent to create a donut chart with a hole in the middle.
Container	Empty space inside the pie. Only visible when no slice is drawn or the pie is not full (see option <i>Fixed value for 100%</i>).
Background	Color of the background outside the pie.

Table 26: Color assignment for pie controls

All other properties are set on the **Pie Chart** property page. The meaning of the values on the left side of the page are as follows:

- **Rotation:** Clockwise rotation of the entire pie.
- **Line Width:** Width of the outline drawn around the pie slices.
- **Center Radius:** Radius of the round center piece.
- **Tick Mark Length:** Length of the tick marks between pie slices. This setting is also used to adjust placement of the accumulated amount labels outside the pie when the tick mark option is enabled, no matter if the tick marks are actually visible or hidden due to a disabled outline.
- **Piece label position offset:** Positive or negative offset to adjust placement of the pie slice labels. The offset is added to the calculated default position of the label to shift it out (+) or in (-) as desired. The default position depends on a number of factors and should already result in a good placement in most cases (see below).
- **Minimum size for labelled piece:** The minimum relative size of a pie slice for drawing a label. Slices smaller than this value will not be labelled, to avoid overlapping text.
- **Fixed value for 100%:** Absolute total amount to represent a full pie (100%). Normally set to 0 to use automatic scaling, so the pie is always full. If set to a value > 0, this value will represent 100% and the pie may not be full if the total amount of all connected data points is less than this number. In case the total amount exceeds the 100% value, the pie slices will be scaled down proportionally to fit the pie. The total relative amount will exceed 100% in this case.

On the right side is a list of additional options to activate or deactivate individual parts of the pie chart, or otherwise modify its appearance:

- **Show relative amount:** Each pie slice will be labelled with its relative size, provided that it exceeds the given minimum size for a labelled piece.
- **Show absolute amount:** Similar to the above option, but the absolute amount as reported by the corresponding data point is shown.
- **Show relative accumulated amount:** Add a label between pie slices, showing the relative fill level of the pie up to that point. This value accumulates the relative size of all slices to the left (counter clock wise) of the label position. Especially for sorted pie charts, this type of label is useful to indicate how many percent of the grand total is related to the largest contributors.

- **Show absolute accumulated amount:** Similar to the above, but the total sum of all absolute values to the left of the label is shown.
- **Show absolute total amount:** Shows a label in the pie center, indicating the grand total sum of all connected data point values.
- **Show units for absolute amounts:** This option reads the unit string from the first connected data point and appends it to all absolute values shown in any of the labels. If the first connected value is a math object instead of a data point, the unit will be fetched from the first data point connected to the math object. This unit will be used to label all absolute values, including the calculated total sum.

NOTE:

For the diagram to make any sense, the unit of all connected data points must be the same. If in certain cases the data points have the same physical unit, but a different order of magnitude, for example some data points provide their value in Wh while others provide it in kWh, the scaling options of the connected data point object must be used to translate all values to the same type of unit, for example multiply the kWh values by 1000 to make them comparable to the Wh values from other data points.

- **Place labels outside the pie:** This option affects placement of the piece labels. For a standard pie chart, the default position of the label is inside the pie slice, near the geometric center of the piece. In case of a donut chart with a large center piece, the default position is centered between start and end of the donut section, with the radial position centered between inner and outer radius of the donut. Using the outside label option, the labels can be placed *outside* the pie instead. The default position in this case is centered between start and end of the slice, as near as possible to the outer edge of the pie, without overlapping pie and label.
- **Sort by relative amount:** Usually, the order in which the pie slices are drawn depends on the order in which the data points are connected to the control. When sorting is turned on, the pie slices are instead sorted by their relative size, starting with the largest piece.
- **Draw Outline:** Draw an outline around the pie and the pie slices. This will also include drawing of the tick marks, if enabled (see next option). Disabling the outline will disable all line drawing, including the tick marks.
- **Draw Tick Marks:** This option affects drawing of tick marks and placement of accumulated sum labels (if enabled). With the tick mark option enabled, sum labels are placed outside the pie. In addition, tick marks are drawn if outline drawing is enabled, and the tick mark length is set to a value greater than zero. Disabling the tick mark option will place the sum labels inside the pie (near the outer edge) and will not draw any tick marks, even if outline drawing is enabled.

9.5.15 Webcam Control

The webcam control is a specialized control to display live video streams of web cameras. For each control, an arbitrary list of cameras can be configured. For every camera, up to four different video stream URLs can be set. This is most useful for LWEB projects, where different browsers may support different stream formats, or for combined LVIS/LWEB projects, where the preferred stream format for LVIS is RTSP, but for LWEB running in a browser not supporting this format, an alternate MJPEG stream must be specified.

The active camera can be changed during run time by value updates from connected input data points. The current camera number and name will be sent to any connected output data points. Analog data points will receive the camera number, while string data points will be assigned the camera name.

Due to the complexity of this control, it is only available on LVIS-3ME and LWEB.

9.5.15.1 Basic Setup

On the **Common Properties** page, a font can be selected for the webcam control. Video streaming errors will be shown using this font and the associated text color. A **Bitmap** can be selected as a default image to show when no camera source is active. This image will be displayed centered in the control container. The control can be set to any size. The video received from the camera will be downsized to fit the control, in case the video resolution exceeds the control dimensions. Note that smaller videos will not be up scaled.

NOTE: *In general, the video stream from the camera should be configured to match the desired control dimensions, such that the control does not apply any downsizing. This will improve playback performance and avoid unnecessary use of network bandwidth. Most cameras allow setting the video stream resolution through arguments given in the stream URL.*

As with other controls, color configuration is done on the **Color** property page. The following colors are supported by webcam controls:

Color Name	Element to which the color applies
Text	Color for error message text.
Bitmap	Foreground color when drawing bitmap images (b&w).
Background	Background color, visible when no video is playing.

Table 27: Color assignment for webcam controls

In addition, there are some **General Options** to be set on the **WebCam** property page:

- **Auto-Cycle:** In case the default camera is set to **Cycle**, or the camera number is set to a value higher than the last camera during run time, the active camera will be changed to the next camera in the list automatically after the given amount of playback time for each camera source.
- **Default Camera:** Using this dropdown, a default camera can be selected for playback until the camera is changed by a connected data point. Apart from the configured cameras, the options **OFF** and **Cycle** are available to specify either no playback at all or auto-cycle through all available cameras.

9.5.15.2 Camera Configuration

On the left side of the **WebCam** property page, the list of associated cameras is configured. Using the buttons above the list, new entries can be added, available entries can be sorted in any order and unwanted cameras can be removed from the list.

The currently selected camera can be configured on the right side of the property page.

- **Name:** Name for this camera. This string will be assigned to string output data points connected to the control, to indicate the name of the camera being viewed.
- **Video Preview:** Set this option to start a live preview of the currently selected URL in the configuration software. The PC running the software must have access to the camera. This can be useful to validate the entered URL and detect any configuration problems early. To avoid delays when opening a page containing a webcam control, preview should normally be off.
- **URL:** Used to select the type of URL to be configured. There are up to four different URLs for each camera. The client will select and use the most appropriate URL, based on its capabilities. When the drop-down is used to select

a different URL type, the text field below the drop-down will show the associated URL and the preview, if active, will switch to this URL as well.

- **Preferred Stream:** Use this to configure the preferred video stream.
 - **Alternate Stream 1:** An alternate URL, for clients which cannot support the preferred stream. For example an MJPEG stream as an alternative to a preferred RTSP stream.
 - **Alternate Stream 2:** Similar to the above, a second alternative in case the primary one is not supported by the particular browser running the LWEB client.
 - **Fallback Stream:** A last resort, in case all other streams turn out to be unsupported. This could be a JPEG URL, from where individual still images can be fetched. Something that any browser can do.
- **Media Type:** An additional hint for LWEB clients regarding the media type of the specified URL. Even though the LWEB client can usually detect the media type automatically, playback will start faster if the media type is already known. For L-Vis devices, this option has no effect. Media type is always detected automatically without any additional delay.

9.5.15.3 Stream URLs

Individual video streams are specified in URL format, similar to the address of a web page. The basic format of a video stream URL is the same for all cameras:

```
<protocol>://[user:password@]<address>[/path][?options]
```

Parts in <> are required, parts in [] are optional. Here is a description of the individual parts which make up the stream URL:

- **<protocol>:** A prefix describing the stream protocol. In almost all cases, this will be either `http` or `rtsp`.
- **[user:password]:** Credentials used for authentication. If specified, a `@` character separates the credentials from the camera address.
- **<address>:** IP address or DNS name of the camera.
- **[/path]:** Additional path component, used to specify the type of video stream to access. Most cameras provide access to predefined, or user defined streams under different path names. However, there is no standard and every vendor does it differently. See below for ways to determine a suitable path for your camera.
- **[?options]:** Optional stream attributes to further specify the video properties. The option part is separated from the path by a `?` character. Individual options consist of an option name and usually a value, for example `videocodec=h264`. Multiple options are separated by a `&` character. Available options depend on the camera model and the type of stream being configured.

Here are some examples for Axis cameras:

Protocol RTSP, video codec H.264, resolution 320x240, 12 frames/sec:

```
rtsp://admin:admin@192.168.19.3/onvif-media/media.amp?resolution=320x240&fps=12&videocodec=h264
```

Protocol HTTP, MJPEG video, 320x240, 24 frames/sec:

`http://192.168.19.3/mjpg/1/video.mjpg?resolution=320x240&compression=30&fps=24&videocodec=jpeg`

To determine valid path and option settings for your particular camera model, the camera web interface often provides a way to configure a stream and see the resulting URL in the stream preview window opened by the web browser. Another possibility is to check the manual that came with the camera, or the vendor's web site. There are also a number of large camera URL directories online, which may list the URL and options for the desired camera. If everything else fails, customer support of the camera vendor will be able to provide the required data.

9.5.15.4 Runtime Control

Since a webcam control supports multiple cameras, data points can be attached to the control to change the active camera during run time and retrieve information about the camera which is currently viewed. The following data point types may be connected:

- **Simple analog or multi state inputs:** The received value will be used to set the active camera. The first camera in the list is camera 1. Values of zero or less will turn playback off, while values larger than the highest camera number will activate automatic cycling through all cameras.
- **Simple analog or multi state outputs:** These types of output data points will receive the number of the currently active camera. When playback is off, the assigned value will be zero. When auto-cycle is active, a new camera number will be assigned every time the active camera changes.
- **String outputs:** String data points can be connected as outputs only. These data points will receive the name of the currently active camera. When playback is off, an empty string will be assigned. When auto-cycle is active, a new camera name will be assigned every time the active camera changes.

The control will disconnect from the active camera and try to establish a connection to the new camera as soon as it receives an update from a connected input data point. However, it will update the connected outputs only after the connection was established and the video playback started, or the connection attempt resulted in an error which is shown in the control. This will ensure that the output data points will accurately reflect what the user currently sees on the display.

NOTE:	<i>Using run time control is optional. If there is only one camera configured, it may be set as the default camera to play when the control is visible. In this case, no data points need to be connected, as the control will start playback of the video stream automatically. The same is true for a control with multiple cameras, when the default playback is set to auto-cycle. However, an output data point may be desired in this case.</i>
--------------	---

9.6 Data Point

Data point objects control the data transfer between a register or a network value and the controls or other data processing objects on the device. To archive this data transfer, there are three objects involved:

- The **control** element, which acts as the interface to the user, or any other object which processes data, like mathematical objects, alarm generators, triggers, or action objects.
- A **data point object** which is connected to the control and references a specific data point, thereby linking the control to a data source and/or sink. It also defines the properties of this data link, for example when new data is to be transferred

between the data point and the control, and how the data should be converted when it is transferred.

- A **data point** as the source or sink of data, for example an internal user register, a system parameter, a network variable or a BACnet server object or client mapping. The data point itself is not visible in the object tree. Instead, it is referenced by data point objects by the unique ID (UID) of the data point.

The **control** is created as part of the object tree, as outlined in the tutorial. It exists independently of any register or network data point. Even if all registers and network data points would be deleted, the control would not be affected.

The **data point** itself is not represented as an object in the tree view, because it is not part of the user interface and is not bound to any of the other objects in the tree. Data points exist on the device independent from the user interface and are identified by a unique ID. They are created and managed in a separate window, the **Data Point Management and Selection** dialog. In some cases, they are created automatically, like the data points representing the system parameters or data points representing dynamic network variables, which may be present on a CEA-709 device. See also Chapter 10 for more information about data point configuration.

Structured data points, as available for example on CEA-709 devices, are represented by individual data points for each of the structure elements, in addition to the top level data point representing the entire network variable. However, most of the complex data points, which represent more than a single scalar value, are not supported by most controls and cannot be attached to them. The following exceptions to this rule apply for the current devices:

- **SNVT_switch:** Output data points of this type may be directly connected to an alarm generator to control the entire switch. See also section 9.11 for detailed information about alarm generators.
- **SNVT_alarm2:** As above, outputs of this type may be connected to an alarm generator to send out alarm messages.
- **SNVT_time_stamp:** This type of data point can be connected to date controls.
- **SNVT_str_asc:** This and other string data types (string registers) may be directly connected to a text control. The string received via these data points will be displayed directly on the text control, without a lookup in a mapping table. If the data point is used as output, the string selected from the mapping table is directly assigned to the data point, without converting to a number first. If the string contains non-ASCII characters, they are represented in the ISO-10646 character set and are transferred in UTF-8 encoded format. This way, it is possible to exchange strings in any language between two L-VIS devices.

The **data point object** is the L-Vis object created when *Add Data Point...* is selected from the context menu of a control to assign a data point to it. The reason why the assignment of data points to controls is done via a separate object; instead of directly connecting a data point to the control has a number of reasons. The most important ones are:

1. Data points represent communication objects which exist on the device. They must be independent of the existence of any control or other user interface object. If a data point was to be connected to a control and the control is deleted, the data point would be deleted as well. Using a reference object to link the control to an existing data point solves this problem. Deleting the control will also delete the reference object (the data point object), but not the data point itself.
2. Many projects need to link multiple controls to the same data point, for example to display a temperature which is received from the network on a bar control and

on a numeric control at the same time. This would not be possible if the data point would be directly connected to a control, since it could only be in one place of the object tree at a time. A separate link or reference object is required, which points to the data point from which the control should receive its value updates.

3. Sometimes it is required to apply simple linear transformations to the value received from the data point before it can be displayed. For example, a data point may receive a temperature value from the network in degrees Celsius, but the temperature displayed on the device should show degrees Fahrenheit, or a data point provides a speed in meters per second, but the display should read mph or km/h. Such translations can be done by the data point object on the fly, while passing values between the control and the data point. This makes it easy to display for example a temperature from an input data point both in degree Celsius and in degree Fahrenheit side by side, using two data point objects referencing the same data point, but using different value translations.

It is important to understand this concept to understand what can be configured on the **Data Point** property page of L-Vis data point objects. The key is that there can and usually will be multiple data point objects in the L-Vis object tree, referencing the same data point in the data point configuration.

9.6.1 Data Point Properties

The following properties can be set on the **Data Point** property page:

Value Translation: If the unit of the data point is known and there are some commonly used translations available for this unit, one of the pre-defined translations can be selected from the drop down list. A custom translation can be defined by filling out the formula below the drop down list. In this formula, NV stands for the value on the network side (Network Value), that is, the value of the data point, and DP stands for the value of the data point as seen by controls or other objects which are connected via this data point.

NOTE:

*The value translation of a data point can be very useful to modify a value on its way between a data processing object and a register or network value. Using this facility, simple data manipulation like the inversion of a state or the addition or subtraction of a fixed offset can be archived without using a separate math object. For example, an action which toggles a light switch may be implemented by assigning the inverted feedback input (current switch state) to the control output. The inversion is done with $DP = -1 * NV + 1$.*

The *Details* section shows details of the referenced data point, like the name of the underlying network object, the network object type, units, and other information. Also shown is the unique ID of the referenced data point, by which the point may be located in the data point management window for further inspection or modification.

In the *Data Point Update Flags* section, the value update mode is specified. There are three basic modes available, together with some additional options:

- **No Update:** This means that this data point object will not request the referenced data point to be updated on the network, even when the value on the control side changes. Useful if this data point object references one element of a structured data point, and updates to this element should not cause the top level data point to be transmitted on the network.
- **Focus Loss:** This update mode transmits the new value only when the input control returned focus and left input mode, that is, when the final value is known. Intermediate values, which are for example generated by moving the bar of a bar control around, are not immediately transmitted. Use this option for push button mode (see section 9.5.3 for more information).

- **Immediate:** All values are immediately transmitted to the referenced data point. This provides direct feedback to the user, while the control is still in input mode and the user is still modifying the value. Currently really only useful for bar controls and controls which use the keypad, since the keypad has + and – keys, which send out intermediate values.
- **System Startup:** The checkbox *System Startup* should be checked to instruct the device to update the value on system startup. If the option is set on an output data point, the default value or the last stored value will be sent out. If it is used on an input data point, the current value will be requested from the underlying data point layer and if available, will be received and processed like a normal value update. See also Section 9.6.2 for details about updates on system startup.
- **Value Changes Only:** This option causes the data point to forward updates only if the value actually changed. It is very useful to avoid unnecessary operations caused by repeated updates with the same value, provided that the data point does not represent an event, in which case all updates must be processed and this option should not be used.
- **No Invalid Updates:** This option causes the data point to filter out updates which set the value invalid (for example because the value cannot be determined from the network). Normally the invalid state is passed on to the other side. If this option is checked, the update is not forwarded, so that there is no change to the current value.
- **Map Invalid to Default:** This option causes the data point to map invalid updates to the default value specified in the *Default* field. This is useful for input data points which may be unavailable (for example not bound) and this situation is not considered an error. Using this option, unavailable data can be replaced by a safe default. If the invalid state would be forwarded and the value would be used in a calculation, the entire result of the calculation would become invalid, which may not be desired.
- **Reverse Direction:** This option inverts the data direction of the data point object to be the opposite of the referenced data point, for example, the value of an output NV may be used as input for a math object or a trigger or may be trended by a trend log control. Similar, the current value of an input NV may be set by a math object or from user input via a control.
- **Bidirectional:** If supported by the parent object of the data point, this option can be used to enable data flow in both directions using only one data point object. For standard applications, the result is the same as if two data point objects referencing the same data point would have been connected, with one of the data point objects using the reverse direction option. In special cases, it can still be necessary to connect two separate objects. This happens when different settings for data translation or filtering (ignore invalid updates, etc.) are required for input and output direction. On a bidirectional data point reference, these options are applied to both directions equally, except for the boot update option, which is applied in input direction only.

In the *Data Point Value* area, the **Default Value** for the data point can be set. This will be the value of the data point until a new value is received from either the network or the user or a value which was saved to non-volatile memory is loaded after system start. If this field is left blank, the initial value of the data point will be invalid (this state is also visible on the controls). Writing the invalid value to a BACnet client mapping will execute a withdraw operation, that is, the value is removed from the priority array of the remote device.

NOTE: *The default value of a data point object ultimately refers to the value of the referenced data point. Since the referenced data point can only have one value at a time, the entered default value will apply to all other data point objects which reference the same data point as well. The entered value will be run through the value translation to determine the resulting value for the underlying data point and this value will be distributed back to all connected data point objects to maintain consistency. Only if the data point object is marked as 'constant value', this process is not done (see next paragraph). When adding a constant value data point, set the constant value flag **before** entering the desired constant value, in order to not disturb the default value of the referenced data point.*

If the *Constant Value* checkbox is set, the point is marked as constant and the entered default value will not be overwritten by values received from the network or data entered by the user. In addition, you may set differing default values for data point objects which reference the same underlying data point, if the data point object is marked as constant.

NOTE: *Constant value data point objects are often very useful together with push buttons of any kind. Create two or more push buttons, for example three text controls with the static texts **LOW**, **MEDIUM** and **HIGH**, enable push button mode and request to always send the current value, then add a temperature output data point to each of the three controls (always using the same data point). Now you can set the data point objects to constant value and enter different values for each of the three objects (for example 18, 22 and 24). When the user now presses the **LOW** button, the value 18 will be sent out. When he presses the **MEDIUM** button, the value 22 will be sent.*

The *Persistent* option is used to request that the current value of the data point should be preserved across a reboot of the device and be used as the new default value, until updates are received. Since the value which will be saved is the value of the referenced data point, this option will also affect all other data point objects which reference the same data point.

The *Timestamp Offset* configuration is currently only used when the data point is connected to a trend control. All timestamp values of recorded data from this data point will be shifted by the indicated amount, causing the trend curve to shift left or right along the time axis. See Section 9.5.6.8 for details regarding this function.

9.6.2 System Startup

Note that the data point update flag **System Startup** is independent of the flag **Poll on Startup**, which may be available for certain network technologies such as CEA709. The two options are related to different functions and do different things:

- The **System Startup** update flag can be set for input as well as output direction:
 - **Input:** Causes the object to which the data point is connected, like for example a control, action, math object, or trigger, to receive an initial value update after device boot up. This update may come from a default value, a persistent value, or a value already received from the network while the system was booting. For data points representing a normal value like a temperature, this option is usually turned on, such that the display shows current information. For data points representing events which can trigger actions, this is usually not desired and turned off.
 - **Output:** For outputs, this function is not desired in almost all cases, since it would trigger an outgoing value update whenever the device starts. It may be useful to trigger actions during system start, for example to send a status mail, record the last boot time in a register or data log, or to switch to a certain page.
- The **Poll on Startup** function is located in the properties of the referenced data source and is only available on input or value data points. The option requests that a poll message be sent out on the network to query a remote device for the

current value. If this is possible at all, or if a current value is automatically received in the course of establishing communication with the remote device, is dependent on the underlying network technology. For example, if the polled flag of an input network variable is not set, the device will not be able to fetch the current value from the sender, since it has no knowledge of the senders network address and therefore cannot send a request for a value update.

NOTE: *The system startup update flag can be controlled separately for input and output direction, but it cannot be enabled or disabled individually for each data point reference in the project. Also note that making a data point persistent may cause an initial value update with the stored value after system boot up, regardless of the system startup update flag. Data points which are used as events should therefore never be set persistent.*

9.6.3 Data Point Icons

For easier recognition of a data point in the object tree, the data point object is shown with one of several different icons, depending on the type and configuration of the data point.

Icon	Data point
	Write to a system register
	Read from a system register
	Write to a user defined register (device internal)
	Read from a user defined register (device internal)
	Write a constant value
	Read a constant value (usually not used)
	Write to a network value
	Write to a network value with heart beat (max send > 0)
	Write to a network value rate limited (min send > 0)
	Write to a network value with rate limit and heart beat
	Passive read from a network value (wait for updates being received)
	Active read from a network value (polling or receive time out / poll)
	Write to a dynamic network variable (CEA709 only)
	Write to a dynamic network variable with heart beat (max send > 0)
	Write to a dynamic network variable rate limited (min send > 0)
	Write to a dynamic network variable with rate limit and heart beat
	Read from a dynamic network variable (CEA709 only)
	Active read from a dynamic network variable (poll or receive time out)
	Write to a configuration property (file access, CEA709 only)
	Read from a configuration property (file access, CEA709 only)
	Local alarm server
	Remote alarm server
	Local scheduler
	Remote scheduler
	Broken reference, data point with the given UID no longer exists

Table 28: Data point icons

9.7 Mapping Table

The mapping table is used to map values of numeric data points to text strings, bitmaps and/or colors. The mapping table object controls the display of text controls, bitmap controls, numeric controls and bar controls and may be added to these objects.

A mapping table can include text, bitmaps and colors at the same time. If connected to a text control, the text part will be shown together with the color settings. When the same mapping object is connected to a bitmap control, the bitmap will be shown. Number and bar controls use the color setting only.

9.7.1 Mapping Table Properties

The properties of a mapping table are set on the **Mapping** property page. The *New Entry* button on this page adds a new entry to the mapping table.

NOTE:	<i>If you want to add a new entry, but the New Entry button is grayed out, you most likely have a control selected in the tree view which does not yet have a mapping table object attached to it. Go back to the tree view and add a mapping table object, or use the Load button to load a mapping table from a file.</i>
--------------	---

The value component of the new entry defines the lower limit for the input value to display the given text, bitmap and/or color. Since multiple entries will be present in the table, the entered values build up ranges, which define the texts and bitmaps to display for every possible numeric value. The valid range for each entry in the mapping table is defined to reach from the value of the entry itself, up to but not including the value of the succeeding entry. If this is the last entry, it is used for all values up to +infinite, since there is no more entry to define an upper bound. For practical reasons, the first entry in the table always ignores the entered limit and is also used for any value lower than its own limit.

Here is an example with four entries:

Value	Text	Displayed when value is...
0	OFF	< 10
10	LOW	10 – 19.9999
20	NORMAL	20 – 49.9999
50	HIGH	>= 50

Once a mapping is defined, it can be stored to a file (*Save* button) and loaded (*Load* button) to define the mapping for other controls in subsequent projects. This way, an archive of frequently used mappings can be built for use in various projects. To use the same mapping for other controls of the same project, just copy and paste the mapping table object in the tree view.

The same mapping table can be used in both directions. Translating incoming values to texts, bitmaps, or colors, as well as looking up a value to send out, when the user selects a given element from the mapping table via a dropdown list or a push button browses the table to find the next entry.

NOTE:	<i>All texts, bitmaps and colors used in a mapping table are included in the file when the mapping table is saved, so that the mapping file is self-contained and can be loaded and used on different PCs, even when the source bitmap files may not be available.</i>
--------------	--

To define the color for a single entry, either select the entry from the list and click the color button to the right of the text input field, or directly click in the area to the right of the text whose color you want to set (in the color bar). A color selection dialog will appear, from which you may select the desired color.

NOTE: *To define a mapping table entry which should not modify the current color, set the color of the entry to Unset (shown by the black cross).*

To define a color gradient for a number of entries (for example going from red to green), set the desired colors for start and end of the gradient and then multi-select the range of entries over which the gradient should be calculated. Hit the button to the right of the New Entry button to calculate a color gradient from the color of the first entry in the selected range up to the color of the last entry in the selected range.

NOTE: *The gradient function will operate on the alpha channel in the same way as for the normal color value, so that on devices which support ARGB color mode, this function can be used to generate gradients from opaque to transparent.*

To set a range of entries back to unset or to a specific color, you can multi-select the entries from the list and then set or reset their color all at the same time.

9.8 Color Mapping

Color mapping objects are used to change individual colors of a control depending on the value of a data point. Mapping data point values to colors works the same way as in a mapping table. The desired color is selected from a table which lists different input values and their corresponding colors. Please also review the functional description of the mapping table object for more details.

There are two major differences when compared to a normal mapping table:

1. Mapping table objects are passive lookup tables. They are only used by the control to support the display of values received, and to build value selection menus for the user to select a value to enter. The control itself is the object receiving or sending values. The mapping table object only assists the control in value translation. In contrast, color mapping objects connected to controls are active, similar to trigger objects. They receive their own values from separate input data points connected directly to the color mapping object. Whenever a new value is received, the color mapping object will translate it to the desired color and reconfigure the control accordingly. This means that color changes through color mapping objects are completely independent from the value currently displayed by the control itself.
2. Any number of color mappings can be connected to the same control, with each mapping targeting one or more different control colors. While the color from a standard mapping table is always assigned to a specific, fixed color (usually the text or container color), a color mapping object can be configured to control any of the available colors.

NOTE: *There is one exception, where color map objects are used as passive lookup tables, similar to mapping table objects. This applies to color maps connected to **data points** instead of controls. In this case, the color mapping attached to the data point can be used by a control to assist in rendering a certain value related to this particular data point. If and how the attached mapping is used, is entirely up to the control.*

9.8.1 Color Mapping Properties

The properties of a color mapping table are set on the **Color Mapping** property page. The *New Entry* button on this page adds a new entry to the mapping table. Once two or more entries are added and their values are set, colors can be assigned to the individual entries by pressing the color patch next to the desired list entry. The *Gradient* button can be used to calculate color gradients in the same way as in standard mapping tables.

To create smooth color transitions, the option **Gradient Map** can be used. When this option is set, the color map will translate values to colors by interpolation of the color table entries below and above the actual value. This creates a truly smooth transition between the colors defined in the table, and avoids the need to add a lot of entries to the table to create a more or less smooth color transition. Only a few color entries are needed to define complex gradients, for example from red, through yellow, to green.

NOTE: *Gradient maps are not available in projects using the VGA color palette, for example projects designed for the discontinued 5 inch LVIS-3E100 and LVIS-ME200.*

To the left of the color table is the **Color Selection** area. This area looks similar to the property page used to set the initial colors for a control. The main difference is that all of the **color buttons** shown in this area **function as toggle switches**. Pressing a color button will select the color for being controlled by the color map. Pressing it again will return the button to normal state and the color will not be controlled by the color map any more.

NOTE: *If the same color is selected in more than one color map on the same control, the resulting color depends on the order of data point updates received. This is because color mapping objects are event driven. It is therefore usually not desirable to have the same color controlled by multiple color mapping objects. It is however common to have a number of different color maps controlling individual colors of the same control.*

9.9 Action Objects

Action objects are used to execute various actions on special occasions. As such, they are often connected to controls, to trigger an action when the user touches, releases, or selects a control. In some cases, they are connected to data point connector objects, to watch incoming data point values and trigger an action when a certain value is received.

In any case, the action objects need a parent object which is able to feed the required information to the connected action, such that the action may monitor what's going on and trigger at the right moment. Possible parent objects where actions can be connected include controls, data point connectors, and page objects.

For every action object, there are three basic things which need to be defined:

- **Action:** What should be done when the action triggers?
- **Trigger:** When should the action be considered for execution?
- **Condition:** Once triggered, under what conditions should the action be executed and how, if at all, should it be repeated?

When a new action is defined, it is important to think about these three components and make sure the selected choices actually make sense. Each of the above properties may be controlled individually, but not all of the possible combinations will actually make sense. Examples of good and bad configurations will be given later on.

Depending on the selected action type, the action object may accept certain child objects to work with. Always make sure to set the desired action type first, before trying to add any child objects to the action. The action may refuse to accept child objects which can't be used for the selected action type. Objects which are no longer useful after the action type was changed are not deleted automatically. Instead, they are marked with a caution sign in the tree view, to indicate that this object is ignored by the selected action type.

9.9.1 Action Properties

Actions are objects which are not visible on the screen of the device, so they do not use any of the common properties like font, position, size, or color. Aside from the **General** properties, which apply to all objects, actions are configured on the **Action** property page.

This page is divided in two sections. The top half is used to define the action that should be executed, when it will trigger, and the conditions under which it will be repeated. The bottom half is used to set options which may be required for some of the actions.

From the first dropdown list, select the action you want to execute. Currently available actions are:

- **Show menu:** Show the current menu, just like performing a press and hold operation on an empty spot on the display. Useful to build menu buttons which show the menu when pressed.
- **Show page:** Jump to the specified page. To select a page, drag a page and drop it on the action object in the tree view, or press the *Select...* button to open the page link dialog (see section about page links). The path to the page will be shown in the option area below.
- **Next page:** Show the next page of a multi-page menu item. This is equivalent to the 'next page' touch gesture and is useful to build a 'next' button.
- **Prev page:** Same as above, but show the previous page.
- **Sound buzzer:** Output a tone on the internal buzzer. The frequency and duration of the tone can be adjusted in the *Options* section. Useful to create audible alarms when a data point or register reaches a certain value.
- **Turn on backlight:** Turn on the LCD backlight, just as if the user touched the display. Useful to turn on the backlight based on information from a presence sensor.
- **Update data points:** Assign a value to all output data points, which are *connected to the action object*. The value to send out can be preset by also connecting one or more input data points *to the action object*. Useful to update the value of data points under certain conditions. Often used to save the current value of an input data point to an internal register when a button is pressed, or similar applications. When the connected output data point is a constant value point, the action may be used to write a fixed value to a register or network object when the action triggers.
- **Service Pin Message:** Execute the system command 'send service pin message', which is supported by CEA-709 models.
- **Lock Pages (Logout):** Immediately protects all locked pages. The user has to enter the PIN code again to unlock the locked pages.
- **Back one page:** Show the page from which the current page was reached, similar to the 'back' button in a web browser. The system keeps a record of the last 256 pages visited via the show page action and picks the most recent of them to navigate back one step.
- **Send E-Mail:** Send out the mail referenced by the action. To select the mail template to use, press the *Select...* button next to the link line and select the desired mail template. Please refer to section 10.9 for a detailed description of mail templates.

- **Play sound:** Available for 12" and 15" devices and LWEB, this action will play a sound selected from the list of loaded sounds. Note that this action will also stop any other sound which may currently be playing.
- **Stop sound:** This action will stop any sound playback which may currently be in progress.
- **Loop sound:** Similar to *play sound*, but will repeat the sound from the beginning automatically.
- **Open Popup:** Set all collections which are *connected to the action* to visible state, as if the collections had a visible on condition trigger connected to them. See also the option settings below for additional configuration of popups.
- **Close Popup:** Set the innermost popup containing the action to invisible state. This is used to implement a close button which is part of the popup itself.
- **Toggle Popup:** Similar to open popup, except that the connected collections will be set to invisible state if the action is triggered while the popup is already visible.

NOTE:

*The action **update data points** can also be used to increment or decrement the value of a register or of a network output / feedback input pair, which is required to implement up/down buttons to dim light or modify a temperature set point. To do this, connect the register read point (or the network feedback point) as well as the register write point (or network output point) to the action. Use the value translation of the read point to modify the incoming value as required (add or subtract a fixed value). When the action is executed, the modified input value will be assigned back to the output value. Since both refer to the same local register or remote actor, the value of the register or the remote actor will be changed.*

Once the action type is selected, choose the desired trigger mode from the dropdown list next to the action list (Execute <action> on <trigger>). The following trigger mechanisms are available:

- **selection:** The action is triggered when the control is selected. A control is selected when the user touches the control and releases the touch screen while still inside the control. If the user moves outside the control after the touch and releases the screen outside, the control will not be selected.
- **value update:** The action is triggered when the *parent object* (the control or data point connector to which the action is assigned) reports a new value, for example because a new value was received via an input data point, which is *connected to the control* or when the user enters a new value using the control. Make sure to distinguish between the trigger mechanism *value update* and the action type *update data points*. The trigger monitors data coming in from the action *parent*, the update action operates on the data points connected *to* the action itself.
- **state:** This is similar to the value update trigger, but it examines the first data point connected to the parent object to see if it is a multi-state value. If this is the case, the dropdown list next to the action trigger is initialized with the available states and the user can select a state from the dropdown list, instead of entering a numerical value. The action is triggered when the current state equals the state selected from the dropdown. If the parent value is not a multi-state value, this trigger cannot be used.
- **touch:** The action is triggered when the control is touched. Note the difference to the *selection* trigger, which fires only when the screen is again released inside the control. A touch is also triggered when the control was released by moving outside without releasing the screen, and then returning back inside the control.

- **release:** The action is triggered when the touch screen is released while inside the control, or when the touch position moves outside the control while the screen is still being touched.
- **short press:** The action is triggered when the control is released after it was touched for a period shorter than the configured long press threshold. This value can be defined in the project settings and defaults to 1 second. See section 11.1.2 for more information.
- **long press:** The action is triggered when the control is continuously touched for the duration of a long press, as configured in the project settings. The default is one second.
- **show page:** The action is triggered when the page to which this action object is connected is displayed. This trigger mode is therefore only available for page actions.
- **leave page:** The action is triggered when the page to which this action is connected is hidden. This trigger mode is therefore only available for page actions.

NOTE:

The trigger of touch and release actions while moving the touch position around between controls (without releasing the touch screen) is consistent with the display of the selection frame. Moving from one control to another without releasing the touch screen will cause the selection frame to follow the movement, always marking the control which is currently touched. The touch and release actions will follow the same pattern, triggering the touch action when the selection frame appears and triggering the release action when the selection frame disappears.

While the action type is independent of the parent object to which the action is connected, the available action triggers vary depending on the capabilities of the parent object, since the action object needs information from its parent to check for trigger and execute conditions. The following limits apply:

- **Control:** If the action is connected to a control, all available triggers and conditions may be used, since a control delivers both a value and user input events (touch, release, select and similar). If a value-dependent trigger mode or condition is used, at least one input data point will usually be connected to the control to deliver new values.
- **Data Point Connector:** If the action is connected to a data point connector, any triggers or conditions which require user input are not available. A data point connector can only feed a current value to the action, therefore the only trigger modes which will actually trigger are **value update** and **state**. From the conditions, the 'while touched' condition cannot be used, since it would require the parent to feed user input information.
- **Page:** If the action is connected to a page, none of the triggers which require a value update are available, since a page does not receive value updates.

NOTE:

*To avoid confusing situations, it is required that only one object at a time is processing user input. Therefore, if an action is connected to a control and the action uses a trigger mode which depends on user input, for example touch, release, or selection, **the user input will no longer be processed by the control but will instead be forwarded to the action.** This will effectively prevent the control from entering input mode and accepting user input, even if an output data point is connected to the control.*

The action type and trigger mode are now selected. The next step is to select the condition under which the action may be executed. The following conditions are available from a dropdown list directly below the action type:

- **Execute once:** When the action trigger fires, the action is executed once, provided that the value condition in the *Options* section is either set to **any**, or the current value of the parent object matches the value condition.
- **Repeat action:** As above, but the action is repeated for a fixed number of times, which is entered in the field to the right of the dropdown box. This is the total number of times the action is executed at most, including the first time. After each execution, a delay which is configured next to the repeat count is inserted. If there is an additional condition (for example a value condition) and the condition becomes false, any further execution of the action is stopped immediately.
- **Controlled by value:** The action is executed as long as the value of the parent object meets the requirements specified in the *Options* section. Note that if this condition is false at the time the action trigger fires, the action will not be executed. As long as the conditions are met, the action will be repeated, again with a configurable delay between each execution (as above). If the delay is set to 0, the action will be executed only once.
- **While touched:** The action is executed as long as the parent object is being touched. As above, observe that this condition must be satisfied at the time the action trigger fires, or the action will not be executed. In practice, this means that the condition ‘while touched’ is useful only in conjunction with the action trigger ‘on touch’ or, in rare cases, ‘on value update’, but not together with ‘on selection’ because selection of a control implies release of the touch screen.

At this point, the *Execute Action* part of the property page should be filled in completely. The contents of this section should result in a meaningful command when read as a sentence, for example:

Sound buzzer on touch. Repeat action 3 times, waiting 200ms after each time.

If you read your settings, make sure they make sense and match your goals. For example, you should not end up with a sentence like this:

Next page on release. While touched, waiting 100ms after each time.

The resulting action, even though it is possible to configure it this way, will not do what you want. There are two major problem areas:

- The action is requested to trigger on release of the control, yet, the condition under which the action is to be executed requests the control to be touched. This is a contradiction and will not be satisfied.
- Even if the trigger were to be corrected, using ‘on touch’, it is very unlikely that you would want this action to show a new page every 100ms while the control is touched. Aside from the delay, which would be too small, there is another major problem here: After the first time the action is executed, a new page will be on the display and the original control will be gone, so you cannot hold on to it.

NOTE:

*The delay value (waiting xxx ms after each time) **always** applies. The action trigger will always be locked for the given duration after each time the action fired, preventing the same action to fire again until the wait time expired. This can be used to limit the frequency at which an action may fire. If the delay is set to 0, it will be turned off. This also causes any kind of repeated execution to be skipped, so the action is executed only once per trigger event.*

Below the *Execute Action* part of the property page is the *Options* section. In this section, any additional information which may be required by action types, action triggers, or conditions, can be set here. Options which are settable are automatically activated, others are grayed out. The following options are available:

- **Value:** Used by the trigger **value update** and the condition **controlled by value**. If both are used together, they use the same settings (differing value settings for the trigger and the execute condition would not make any sense, since the condition must match at the time the trigger fires). The two special entries in the *Value* dropdown list called **to invalid** and **to normal** check the reported value state rather than the value itself. They will evaluate to true every time the value state changes from normal to invalid or vice versa.
- **PopUp:** Options to configure how popups are shown.
 - **Original Position:** The popup is shown exactly where it was placed in the project. No changes are applied to this original position.
 - **Center:** The popup is centered on the screen.
 - **Autoplace:** The popup is placed near the button which triggered the action, but not overlapping the button if possible.
 - **Close after idle time:** To use with any of the placement options, causes the popup to be closed automatically after the user did not touch the screen for the amount of time after which popups should be closed automatically (including popup menus). The timeout value can be changed in the project settings dialog.
 - **Close when parent is hidden:** This option causes the popup to close when the parent control, to which the popup action is connected, becomes invisible. It doesn't matter why the parent control is hidden, for example due to a visibility trigger, or because the control is part of a parent popup which is closed. Whenever the anchor control disappears from the screen, the popup will disappear with it. This can be used to automatically close child popups when the parent popup is closed, or to close open popups in a tabbed UI design, when the user switches to a different tab.
- **Sound buzzer:** Used by the action type 'sound buzzer' to set the frequency and the duration of the tone. Useful values for the frequency range from about 500 up to 8000. The most intense sound can be archived at around 4000 Hertz. The duration is normally a value between 100 and 3000ms. (0.1s to 3s). Note that the repeat delay should be larger than the tone duration, since the repeat delay starts to run immediately after the action was triggered, not when it completed.
- **Volume adjustment:** Available on LVIS-3ME devices for all buzzer and sound actions, to adjust the output volume within a range of -40dB to +12dB.
- **Link:** This is not an input field, but a display of the destination page for a *show page* action or the name of the selected mail template for the *send mail* action. To set a new page, drag and drop the desired destination page on the action in the tree view. To select a new mail template, use the button next to the link field.

9.9.2 Page Links

The action *Show page* requires a link to the desired target page to show. There are different link types available, depending on the project type and firmware version. On the property page of the action, the *Link* information line will show the current link, which can be edited either by dropping of the desired target page onto the action object, or by opening the link editor dialog by pressing the *Select...* button. The link editor provides the following options to set up the page link:

- **External Link:** This type of page link is available for LWEB projects. It consists of a URL to another LWEB project to load, and a UID of the page within the external project. The LWEB client will load the specified project and show the indicated page.
- **Internal Link:** This type of link is a relative path from the page on which the action is located to the desired target page. The path is built from individual path components, separated by a forward slash (/). Each path component can be the name of an object as shown on the *General* property page, or one of several special tokens to navigate through the object hierarchy:
 - / The forward slash is the path separator. When a path starts with this token, the rest of the path is relative to the root menu instead of the page where the action object is located. This can be used to specify an absolute path by name or location in the tree instead of object ID (see direct link below).
 - .. This token means: Move one hierarchy level up. For example, starting at the page where the action is located, the .. token will move up to the menu item in which the page is located. Another .. would move up further, to the menu in which the item is located, and so on.
 - \$U This token moves up one object on the same hierarchy level, that is, it moves to the previous sibling. For example, if there are multiple pages defined underneath one menu item, the \$U token would move to the previous page under the same menu item.
 - \$D Similar to \$U but moves down to the next sibling.
 - \$M1 Moves to the sub-menu of the current menu item. This token is currently only meaningful on menu items, since only they can have a sub-menu.
 - \$Ix Similar to \$M, but moves to the xth menu item of the current menu. Since a menu may have more than one item, the desired item number must be specified. The first item is referenced by \$I1. This token is only meaningful on a menu object.
 - \$Px Similar to \$M, but moves to the xth page of the current menu item. This token is only meaningful on a menu item.
 - \${bxfy} Where **x** is a digit between 1 and 4 and **y** is a digit >= 1. This token is valid when the action is defined in a template. It will be replaced by the specified component of a data point base folder path during template instantiation. See section 9.15.8 for detailed information about data point path placeholder tags. It can be used to create page links which change dynamically, based on the paths assigned to the template instance.

- **Direct Link:** This link type references a certain target page by its object ID, as shown on the *General* property page. A direct link can be set easily by dropping the desired page onto the action object.

Note that on the property page of the action, there is a button to convert between a direct link and an internal link. This can be useful for beginners, to find out how a relative link to a certain page could be built. Drag the desired target page onto the action to set the direct link, and then convert it to a relative path.

Here are a few examples of possible relative page links:

- `../$P1` Move to the first page of the current menu item.
- `../..$/I2/$P1` Move to the first page of the second menu item in the current menu. Note: The menu is two levels up from the page.
- `../Floorplan` Move to the page called *Floorplan* in the current menu item.
- `../$D/$P1` The first page of the next menu item in this menu.
- `../${b2}` Move to the page with the same name as the rightmost folder name of the second data point base path assigned to the template instance which contains the show page action.

9.10 Collections

A collection object works like a folder on a file system. It helps the user to organize objects in groups. Collections can be placed on pages to group objects together and manipulate them as one unit. It is also possible to create nested collections to build complex macro blocks which consist out of a number of smaller building blocks.

Following the root menu of the main tree view is a collection called *Global Objects*, where objects can be collected which work invisibly in the background, without a graphical representation on the screen and without a direct relation to any object on the screen, for example alarm generators, data point connectors, or mathematical objects. However, it is recommended to keep all objects which are required for a specific control or control group to function together on the same page. This way, the page continues to work when it is copied or moved to another project, because the required invisible objects are copied together with the page. The base collection outside of the main menu should only hold objects which are related to the device, rather than specific page functionality.

The last collection in the tree view is called *Templates*. It is used to collect all template pages of the project. Templates are a complex topic, which is covered in more detail in section 9.15 of this manual.

9.10.1 Collection Properties

Aside from the properties on the **General** property page, which apply to all objects, a collection keeps record of the position and size of the bounding box which can be drawn around all visible objects contained in the collection. This data can be seen on the **Common Properties** page. In the context menu of the collection are a number of commands to add new objects and control the visibility of the entire collection (*Hide / Show Contents*).

9.10.2 Collection Handling

If a collection is part of a project page and contains visible objects, a bounding box around all objects contained in the collection is drawn in the preview when the collection is selected. Using this box, the entire collection may be moved to another location on the page. It is also possible to resize the bounding box, which will evenly spread out the

contained objects. For example, if a collection contains three objects in a horizontal row where the second object is located in the center, resizing the bounding box will keep the leftmost object at the left edge, the rightmost object at the right edge and the middle object in the center of the bounding box, that is, the relative position of each object in the collection does not change.

NOTE: *A collection can only be resized if there are at least two visible items inside the collection. In addition, resizing is only possible in a direction where at least two controls are not at the same position. This is a direct consequence of the way resizing works.*

To select a control inside a collection, the easiest way is to select it from the tree view at the left. If the control is to be selected directly in the preview, there are at least two objects on top of each other (the collection and the control). If there are nested collections, or if controls overlap on the page, there may be even more layers of objects. There are different ways to select a specific object in such a case: Each of these methods will:

- **Left mouse button double click** moves the current selection down to the next object under the cursor in the layer stack. When the bottom most control is reached, the selection moves back to the topmost control.
- **Middle mouse button single click** does the same as the left double-click.
- **Tab key** moves the selection down to the next object on the current page in the tree view. From the last object on the page, the selection moves back to the first object on the same page.

Using the tab key is especially useful while using the keyboard cursor keys to move and resize a control. Using the tab key to switch the selection to the next object on the page can be convenient in this situation.

The command *Lock/Unlock Position* in the context menu of the collection allows locking the current position of all objects inside the collection, to protect them from accidental move while working on other objects in the LCD preview.

9.11 Alarm Generators

The alarm generator object is used to monitor input data points and issue alarms based on rules which are set on the **Alarm Generator** property page. For a discussion of alarming on L-VIS devices in conjunction with the underlying network technology and system wide alarms, please also refer to sections 10.3.1 and 10.11 of this manual.

NOTE: *It is highly recommended to consider using generic alarm server data points together with alarm conditions at data point level, in favor of alarm generator objects.*

To add a new alarm generator, use the context menu of a collection object in the object tree and select 'Add Alarm Generator'. A new alarm generator object will be created and connected to the collection object.

Once the alarm generator object is created, connect the input data points which should be monitored. Each of the connected input data points will show up on the alarm generators property page, where the monitoring rules can be defined.

NOTE: *It is sometimes useful to connect the same input data point twice or more times to the same alarm generator, when the desired alarm conditions are too complex to cover in one rule, for example different rules for alarm set and alarm clear or multiple non-overlapping value ranges which should trigger an alarm. In this case, assign different names to the connected data points on the **General** property page, so that you may distinguish between them later.*

Once the required input data points are connected, the alarm generator needs one output data point to which the result should be written. The desired output data point must be connected before the alarm generator can be configured, because the type of data point determines the configuration options. The following data point types may be used as outputs for the alarm generator:

- **Alarm Server:** On CEA-709 devices, a local alarm server data point may be connected to deliver the alarm to remote alarm clients and display the alarm on an alarm list control. The alarm server keeps record of all pending alarms, delivers them to any connected clients and manages acknowledgement of alarms.
- **Scalar Value:** Any data point representing a single scalar value, for example a register. In this case, the alarm generator uses the connected data point as a Boolean value, setting it to either 0 (OFF) or 1 (ON). Using the data points scaling factors, this Boolean value may be translated into any two values, one to signal 'alarm clear' and one for 'alarm set'. This is most useful if the generated alarm is to be processed further or displayed on the L-VIS device, for example to trigger actions or display warning messages.
- **Multi-State Value:** Similar to a scalar value, but the alarm generator can be configured to set the output to any of the available states, which can be selected easily from a dropdown box. Useful on BACnet devices, where a multi-state point may be used to signal a number of different alarm conditions.
- **SNVT_switch:** Available on CEA-709 devices, a data point representing an entire switch variable may be connected. In this case, the alarm generator sets both value and state of the connected switch to either ON/100 or OFF/0. Useful in cases where the generated alarm is to be processed by a remote device which uses one switch input for each alarm.
- **SNVT_alarm2:** Available on CEA-709 models, a data point representing a SNVT_alarm2 structured NV may be connected. In this case, the user may set all elements of the alarm2 structure to the desired values, including the alarm priority and a description. This is most useful to communicate the result to a remote device which provides a compatible alarm input variable.

9.11.1 Alarm Generator Properties

Once all inputs and the output are connected, the alarm generator may be configured on the **Alarm Generator** property page. For each of the connected inputs, an independent set of parameters is required. The available input points are listed in the top left area of the property page.

NOTE: *If logic or arithmetic operations are required to determine an alarm condition (for example signal an alarm when $T1 > T2$), a mathematic object is required to conduct the required operations. The result, which is conveniently stored into an internal register, is then monitored using the alarm generator.*

The suggested workflow to configure the alarm generator is as follows:

- First, select the input point for which the monitoring rules should be configured.

- Define the *Alarm Condition* using the radio buttons at the bottom left of the page. The available choices allow basic monitoring functionality and always define the condition, under which the alarm should be SET. If the value no longer satisfies the condition, the alarm will be cleared. To filter out spikes, it is usually good to set an amount of time which the alarm condition needs to be satisfied, before the alarm is actually set. The default of one second is often a reasonable value.
- Define the *Data Point Value Range*. Depending on your choice for the condition, you may need to enter values either for both (minimum value and maximum value) or only for one of the two. If your condition is inside or outside range, you need to enter both values to define the range, for the condition 'above max' only the maximum value is required and for the condition 'below min' only the minimum value needs to be filled in.
- Define the *Action* that should be taken. The basic choice here is to have the alarm generator SET the alarm when the condition is satisfied and also CLEAR the alarm when the condition is no longer satisfied. However, you may want to only SET the alarm when the condition is met and not clear it again, because you may require the user to manually clear the alarm later. In this case, use the 'set on alarm enter' action, which will only set the alarm, but will never clear it. To actually clear the alarm, you can either use other means to update the alarm output variable, or you can have a separate input on your alarm generator and define a rule for this input, which will eventually clear the alarm, using the 'clear on alarm leave' action.
- As the last step, define the desired output, either for both conditions (set and clear) or for only one of them. The required input fields will be active, while the other ones will be grayed out. Note that the description field may currently only contain ASCII characters.

The above procedure needs to be repeated for all connected input data points.

9.11.2 Application Hints

While it is pretty straight forward to configure a simple alarm generator to output an alarm when for example a temperature reaches a certain maximum value, the alarm generator can be used for more complex applications as well. Two of the most often used applications are described in more detail in the next sections.

9.11.2.1 Alarm Condition with Hysteresis

Consider you want to issue an alarm when a temperature rises above T1, but the alarm should not be cleared unless the temperature falls below T2, which would be less than T1. Similar to a thermal protection which, once it activates, turns off the device and does not turn it back on until the device significantly cooled down, to avoid constant ON/OFF cycles.

Such an application can be done by connecting the temperature data point two times. To distinguish between them, append SET to the name of the first data point and CLEAR to the name of the second data point (on the **General** property page).

On the **Alarm Generator** property page, configure the SET data point to cause the *Action* 'set on alarm enter', enter T1 as the maximum value of the *Data Point Value Range*, and select *Above Max* for the alarm condition. For the *Alarm Output*, choose the desired output value for the *Set Action*. The *Clear Action* should be grayed out, since the rule specifies no alarm clear action.

Now there are two options available to configure the CLEAR data point. Here is the first one:

Like on the SET data point, keep the *Alarm Condition* at *Above Max*, but enter T2 as the maximum, instead of T1. For the *Action*, select *Clear on Alarm Leave* and select the desired output value from the dropdown list at the right. This time, the *Value for Alarm Clear* box will be active, while the others will be grayed out.

This method is the correct way of doing it, but it may be counter-intuitive. Many people choose *Below Min.* for the alarm condition, because they want something to happen when the temperature falls below T2. However, the condition specified here is always the *Alarm Condition*. It defines the conditions under which the alarm is considered to be SET. It does not define *when* something will happen. This is done using the *Action* setting. The alarm condition in this example is still ‘above a certain temperature’ and action will be taken when this alarm condition is CLEARED, e.g. the device has cooled down enough. Now, if this seems unreasonable, here is an alternative approach:

You may configure the CLEAR data point using the alarm condition *Below Min.* and enter T2 as the minimum value of the *Data Point Value Range*. Since you now defined ‘below T2’ as the alarm condition, you need to take action when the alarm is SET, so the action must be *Set on Alarm Enter*. Now select the desired output value from the *Value for Alarm Set Action* dropdown list. This value will represent the cleared alarm, since the temperature now is below T2. Overall, this way seems more awkward than the first, because the alarm output is cleared by causing an alarm condition.

9.11.2.2 Alarm Clear via Data Point

Similar to the first example, but instead of defining a temperature T2 below which the alarm will clear automatically, a manual input should be used to clear the alarm. Of course, the two examples may be combined to have both, automatic clearance below a safe temperature T2 plus manual clearance as long as the temperature is at least below T1.

First, we need to connect the input data point, via which the alarm should be clearable. This may be the state component of a switch, a Boolean value, a register, or any other scalar value. We will assume an internal register for this example, which we assume will be set by the user pressing a button on the display. We assume that pressing the button will set the register from 0 to 1 and releasing the button will set it back to 0 again.

The first part is the same as in the previous example. The temperature variable is connected to the alarm generator and configured to cause an alarm when the temperature rises above T1.

To clear this alarm, we configure the connected register data point as follows:

Set the *Alarm Condition* to *Above Max*. and enter a time delay of for example 3s. This will require the user to press and hold the button for 3 seconds, before the alarm will be cleared. Enter a value between 0 and 1 for the maximum of the data point value range, for example 0.5 would be a good value. For the *Action*, select *Set on Alarm Enter* and select the desired output value for a cleared alarm from the dropdown list named *Value for Alarm Set Action*. This is because we actually define an alarm condition when the button is pressed. As a result of this alarm condition, we then reset the ‘real’ alarm.

NOTE:	<i>The above examples may of course be combined, to have automatic alarm clear when the temperature falls below T2 and manual clear via a button when the temperature is at least below T1.</i>
--------------	---

9.12 Data Point Connectors

Data Point Connector objects are used to receive a value from one or more input data points and copy the received value to all connected output data points or other objects which accept a value, for example action objects.

A data point connector has no visible representation on the display and is therefore not related to any of the visible objects. It can only be connected to a collection object, as described earlier. To create a new data point connector, use the context menu of the collection object to which you want to add the data point connector. Other than the general properties like object name and description, there are no further properties to set.

Whenever a new value is received via one of the connected input data points, the value will immediately be forwarded to all connected output data points and action objects. This can be useful for example to connect an internal register to a network output, or to trigger an action **on value update**.

It may also be used to connect input and output data points which are of similar but incompatible types, for example receive a value from a SNVT_temp variable and output its value via a SNVT_temp_f variable. If a structured data point should be connected entirely to another data point of the same type, each of the structure elements needs to be connected individually using a separate connector, because each connector is only able to handle single values.

If the device is running firmware 4.3.0 or newer, an alternative method of connecting data points, which also supports connections between entire data structures or connections between different devices, is available in the data point manager dialog. See section 10 for more information about this topic.

NOTE:	<i>The functionality of a data point connector or any other object for which a change of an input value causes updates to an output value without any further conditions or limits is inherently dangerous. You must be careful not to cause endless update loops using these types of objects. See section 15.6 for more details on this topic.</i>
--------------	--

9.13 Mathematic Objects

Mathematic objects are used to do calculations on connected input data points and assign the result to the connected output data points and the parent object to which the math object is connected. The math object calculates its formula every time an update is received via one of the connected input data points, so that the output value is always accurate. If a complex formula is used, which operates on a large number of input data points, the resulting update rate of each input data point should be considered, so that the resulting total update rate remains reasonable.

NOTE:	<i>Math objects are often used as standalone objects, reading from the connected inputs and writing the result back to a connected output. Such math objects are organized in folders, either below the global object folder or within a folder located on a page. However, math objects can also be used in place of input data points. In this case, they do not need an output data point. They deliver the result directly to their parent object, for example a control, a data point connector, or another math object.</i>
--------------	--

9.13.1 Math Object Properties

Mathematical objects are configured on the **Math Object** property page, where basically the formula is entered. Before doing so, it is important to connect the required input data points to the object, so that they are available as variables for use in the formula.

All connected input data points are listed on the left side of the property page and variable names are assigned to the data points in the order in which they are connected to the math object. Variable names start with v1 and continue with increasing numbers, like v2, v3, v4 and so on. There is a limit of 127 input data points per math object. Formulas using more than 127 input variables must be split into individual sub-expressions, which should be done anyways since a formula with such a large number of variables becomes unreadable.

NOTE: *Always keep in mind that the order in which the input data points are connected to the math object in the tree view defines the assignment of variable names for the formula. If you replace input data points when the formula was already specified, be careful not to mess up the data point order and thus the result of your calculations.*

Using the assigned variable names, the user may now enter a formula directly below the list. As you enter the formula, it will be parsed and the resulting sequence of calculations will be displayed in a list at the right of the property page. This list shows your formula in reverse polish notation (RPN), also known as postfix notation, as used by many scientific pocket calculators. Users of such calculators will be immediately familiar with this list and may find it more convenient to construct the formula using the list at the right, instead of the input at the bottom. To learn more about postfix notation, there are a lot of resources and articles readily available on the internet.

NOTE: *When the formula entered at the bottom is still incomplete and does not yield a meaningful command sequence, the list showing the RPN equivalent will be empty. This allows the user to immediately see if the current input is valid or not.*

The operators +, -, /, *, %, AND, OR, XOR, ^, &, |, =, !=, <, >, <=, >= can be used instead of the explicit function calls. Further, it is possible to use parenthesis to define the precedence of the operations.

The buttons and input fields in the middle of the page are used to directly edit the formula using the reverse polish notation. To add a variable to the stack, double-click the variable in the list at the left side, or select it and click the *Add Variable* button. To add a constant value to the stack, enter the value in the input field above the button *Add Value* and then press the button. To add a function to the stack, select the desired function from the dropdown list and press the *Add Function* button. When a function is added to the stack, it will first fetch as many values from the current stack as it needs input values, then execute its operation, and put the result back on the stack.

NOTE: *Even if you use the infix notation at the bottom to enter your formula, the dropdown list of available functions will be useful, to look up the names of functions you want to use and see how many arguments they accept.*

9.13.2 Application Hints

A few functions end with a ... (three dots) in the argument list. This means that they accept a variable number of arguments. When used in the formula, they will fetch all available values from the stack (in RPN syntax) and then calculate the result, which will be put back on the stack and be the only value on the stack, since all other values were used as input to the function.

This behavior causes some limits in how these functions may be used. You are on the safe side, if you use such a function only as the outermost function, or, in RPN, as the last function on the stack, for example:

```
sum(v1, v2, exp(v3, -1))
```

Or, in RPN notation: v1, v2, v3, -1, exp, sum

CAUTION: If you have to use such a function as an argument to another function, it may only be used as **the first argument**. Otherwise the formula cannot be processed by the math object, which internally uses an RPN machine with precompiled instructions for optimal performance. Example:

```
add(avg(v1, v2, v3), 5) or the equivalent avg(v1, v2, v3)+5 will work.
```

```
add(5, avg(v1, v2, v3)) or 5+avg(v1, v2, v3) will not work.
```

To limit the number of re-calculations, the data point update option *Value changes only* should normally be checked on all connected input data points. This avoids recalculating the formula and writing a value to the output data point when it is already clear that the result will be the same, because the input value did not change. The same option can also be checked for the output data point to avoid unnecessary writes to the output data point, in case the inputs changed but the result of the formula is still the same.

NOTE: *Especially for projects which use a lot of cascaded formulas, where the result of one formula is used as input to a number of other formulas, it may cause a big difference in CPU usage whether results of the same value are forwarded to other math objects in the system or not, because they will trigger recalculation of a potentially large number of other formulas, which will generate even more results of the same value (since the originating value did not change).*

Note that the function `rand()` takes no arguments. It always returns a random number between 0 and 1. To generate a random number between the two limits L and H, use a formula like this: $L + (H - L) * \text{rand}()$.

For cases where the low limit L is 0, this is reduced to: $H * \text{rand}()$.

Example for a random number between 20 and 30: $20 + 10 * \text{rand}()$.

9.13.3 Function List

The following function calls are currently supported:

Function	Return Value
add(v1,v2)	$v1 + v2$
sub(v1,v2)	$v1 - v2$
mul(v1,v2)	$v1 * v2$
div(v,d)	v / d
mod(v,m)	Returns the remainder of dividing v by m, where v and m should be integer values. Fractional values will be rounded to the nearest integer automatically
max(v1,...)	Returns the maximum of all values on the value stack
min(v1,...)	Returns the minimum of all values on the value stack
avg(v1,...)	Returns the arithmetic mean value of all values on the stack
log(v)	Returns the natural logarithm of v
log2(v)	Returns the base 2 logarithm of v
log10(v)	Returns the base 10 logarithm of v
exp(v)	Returns the value of e (the base of natural logarithms) raised to the power of v
exp2(v)	Returns the value of 2 raised to the power of v
exp10(v)	Returns the value of 10 raised to the power of v
sqrt(v)	Returns the non-negative square root of v
pow(v,exp)	Returns the value of v raised to the power of exp
round(v)	Round v to the nearest integer
floor(v)	Round v down to the nearest integer
ceil(v)	Round v up to the nearest integer
sum(v1,...)	Returns the sum of all values on the stack
and(b1,b2)	logical AND of the Boolean values b1 and b2 $(b1 \& b2)$
or(b1,b2)	logical OR of the Boolean values b1 and b2 $(b1 b2)$
xor(b1,b2)	logical exclusive OR of the values b1 and b2 $(b1 \wedge b2)$
not(b)	logical inverse of the Boolean value b $(!b)$
lt(v1,v2)	returns 1 if v1 is lower than v2, else returns 0 $(v1 < v2)$
le(v1,v2)	returns 1 if v1 is lower or equal v2, else 0 $(v1 \leq v2)$
eq(v1,v2)	returns 1 if v1 equals v2, else 0 $(v1 = v2)$
ge(v1,v2)	returns 1 if v1 is greater or equal v2, else 0 $(v1 \geq v2)$
gt(v1,v2)	returns 1 if v1 is greater than v2, else 0 $(v1 > v2)$
if(b,vt,vf)	returns vt if b is true, else returns vf $(b ? vt : vf)$
encode(b1,...)	Reads all values from the stack, converts them to Boolean values and encodes them into an integer value, where the first value is used as the LSB and the last value as the MSB.
sin(v1)	Returns the sine of v1, where v1 is given in radians
cos(v1)	Returns the cosine of v1, where v1 is given in radians
tan(v1)	Returns the tangent of v1, where v1 is given in radians
sinh(v1)	Returns the hyperbolic sine of v1, which is defined mathematically as $(\exp(v1) - \exp(-v1)) / 2$
cosh(v1)	Returns the hyperbolic cosine of v1, which is defined mathematically as $(\exp(v1) + \exp(-v1)) / 2$
tanh(v1)	Returns the hyperbolic tangent of v1, which is defined mathematically as $\sinh(v1) / \cosh(v1)$
asin(v1)	Returns the arc sine of v1; that is the value whose sine is v1
acos(v1)	Returns the arc cosine of v1; that is the value (in rad) whose cosine is v1
atan(v1)	Returns the arc tangent of v1; that is the value (in rad) whose tangent is v1

Function	Return Value
asinh(v1)	Returns the inverse hyperbolic sine of v1; that is the value whose hyperbolic sine is v1
acosh(v1)	Returns the inverse hyperbolic cosine of v1; that is the value whose hyperbolic cosine is v1
atanh(v1)	Returns the inverse hyperbolic tangent of v1; that is the value whose hyperbolic tangent is v1
gamma(v1)	Returns the value of the Gamma function for the argument v1. The Gamma function is defined by $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$. It is defined for every real number except for non-positive integers. For non-negative integral m one has $\Gamma(m+1) = m!$ and, more generally, for all x: $\Gamma(x+1) = x * \Gamma(x)$ For $x < 0.5$ one can use $\Gamma(x) * \Gamma(1-x) = \pi / \sin(\pi * x)$
abs(v1)	Returns the absolute value of the argument v1
rand()	Returns a random value between 0 and 1

9.14 Trigger Objects

Trigger objects are the inverse of action objects. While action objects usually receive input from a control and use this input to execute some actions unrelated to the control itself, for example update some data points, trigger objects work in the opposite direction. They receive input from data points and cause the control to which they are connected to execute some action, for example show another page of information or clear the input value.

Trigger objects are most often used with data log controls, to control the recording of data and navigate through the recorded information, if the built-in navigation bar is disabled.

9.14.1 Trigger Object Properties

To define the trigger condition, connect the input data points which determine when the trigger should fire. Using these data points, enter the required formula on the **Trigger Object** property page. Then select the action which the control should execute. Depending on the control, the supported actions will be available from the **Command** dropdown list. The possible commands to give to the control are:

- **No Operation:** This command does nothing. It is usually available as the only choice, if the control does not support any commands in its current configuration.
- **Show Prev Page:** Instructs the control to show the previous page of information.
- **Show Next Page:** Instructs the control to show the next page of information.
- **First Page:** Instructs the control to show the first page of information.
- **Last Page:** Instructs the control to show the last page of information.
- **Add Data Record:** Instructs the control to record the current data of the data record to which the trigger is connected. See also section 9.5.8 for more info about data log controls and data records.
- **Unset Data:** Instructs the control to clear/release the current value. Only available for input controls. The value will be set back to 'invalid', indicating that no value is available.
- **Hide:** Causes the control to become invisible.
- **Show:** Sets the control to normal visibility mode (in case it was invisible).

- **Show on Top:** Makes the control visible and places it in front of any other controls on the screen, overruling the Z-order defined in the project.
- **Start blinking:** Turns on an automatic blink function. Depending on the *Options* on the right side, the control starts to blink at a **slow** or **fast** speed. It continues to do so until it receives a different visibility command, like show or hide.
- **Blink on condition:** This command can be used to control the blink function based on a condition. While the result of the formula is nonzero, the control will be blinking. A zero result will stop the blinking and set the control either static visible or invisible, depending on the option **Visible when inactive**. Turn the option on to have the control visible while it is not blinking.
- **Visible on condition:** This command instructs the control to be either visible or invisible, depending on the result of the formula (0 = invisible, all other results = visible). For cases where the visibility of a control is directly related to the value of a data point or a logical expression, it is easier to use this command instead of two distinct commands to hide and show the control.
- **Top visible on condition:** Same as *Visible on condition*, but places the control in front of any other controls (in visible state).
- **Left Position = v1:** The control is moved on the screen such that its left edge is at the absolute horizontal screen coordinate specified by the variable *v1*, which is the first connected input data point.
- **Top Position = v1:** The control is moved on the screen such that its top edge is at the absolute vertical screen coordinate specified by the variable *v1*, which is the first connected input data point.
- **Left Offset = v1:** The control is moved on the screen such that its left edge is offset by the amount of variable *v1* from the horizontal screen coordinate specified in the project. An offset of 0 returns the control to its original horizontal position, a negative offset moves to the left, and a positive offset moves to the right.
- **Top Offset = v1:** The control is moved on the screen such that its top edge is offset by the amount of variable *v1* from the vertical screen coordinate specified in the project. An offset of 0 returns the control to its original vertical position, a negative offset moves up, and a positive offset moves down.

NOTE:	<i>All commands related to the visibility and position of objects can also be connected to collections. In this case, the same command will be sent to all controls inside the collection.</i>
--------------	--

Next, choose a trigger mode from the available choices. The trigger mode works together with the trigger condition, which was entered as a formula above. First, the formula is calculated and interpreted as a Boolean value, where 0 means false and all other values mean true. The result is then used to trigger the command based on the trigger mode:

- **Any value update:** This mode is a special case. It does not require a trigger condition. Instead, it fires the trigger every time any of the connected input data points receives an update. It is a very simple trigger mode, designed especially for cases where the trigger is used to record data whenever the data changes.
- **Updates resulting in a TRUE condition:** In this mode, the trigger will fire every time one of the input data points received an update and the trigger condition has a TRUE result.
- **Updates resulting in a FALSE condition:** This mode is the inverse of the above. The trigger fires for every update which results in a FALSE trigger condition.

- **Updates resulting in a condition CHANGE:** This mode fires the trigger every time the result of the trigger condition changes, from TRUE to FALSE or vice versa. Subsequent updates which lead to the same result do not repeatedly fire the trigger, as opposed to the above modes.
- **Updates resulting in a CHANGE to TRUE:** This mode fires the trigger only when an update is received which results in a TRUE trigger condition and the condition was previously FALSE. The trigger does not fire when the condition changes back to FALSE.
- **Updates resulting in a CHANGE to FALSE:** This mode is the inverse of the above mode. Only if the trigger condition changes its result from TRUE to FALSE, the trigger is fired.

To limit the frequency at which the trigger may fire and for example cause data to be recorded, a minimum time between two trigger events can be specified, together with a maximum burst rate, which is the maximum number of events that are allowed in fast succession after a long idle period.

NOTE:	<i>For commands which control the position or visibility of controls, a rate limit is usually not desired. Set the minimum time between events to zero to disable rate limiting.</i>
--------------	--

9.15 Templates

Templates are used as a way to design common building blocks of the project, which are needed on multiple pages. As opposed to a simple copy of a folder containing the required objects, a template is a central point where changes can be made and these changes will automatically be applied to all instances of the template in the project.

A template may contain any object which can be part of a collection. This includes controls, collections, math objects, triggers, actions, data points, and even instances of other templates.

Of all these objects, the data points in a template may be handled in a special way which is explained in detail later on. It allows multiple instances of the same template to refer to a different set of data, so that for example a template which displays the state of an AHU may be instantiated once for every AHU and display the state of the corresponding unit without modifying the template.

To add a new template, select *Add Template* from the context menu of a template folder.

9.15.1 Template Properties

Both templates and their instances share the same property page called **Template / Instance**. The page contains four *Base paths for relative data point references*. Their meaning is slightly different for templates and template instances. The next sections will explain how to use these base paths.

9.15.2 Creating a Template

To add a new template, use the *Add Template* command from the context menu of a template folder. A new template object will be created and a fresh page will be shown in the LCD preview. This page is used to place the visible objects of the template, similar to the design of a normal page. The position of the objects placed on this page will be the initial position when the template is instantiated on a page and the template instance is not moved to a different position.

In most cases, a template is used because it will be instantiated multiple times in the same project. However, it will normally be required for these instances to operate on different

data points. The way to make this possible is to distinguish between fixed data points which will be the same for every instance of the template, and variable data points which will be replaced by different data points for every instance.

This is done on the property page of the template by selecting up to four data point folders, which will be considered a base path for relative data point references. This means that any data point in the template object which is located below any of these base folders will be considered a variable data point, whereas all other data points will remain fixed.

NOTE:

If there are multiple choices of base folders, select any one of them and make sure to use the data points of the selected folder to create the data points. For example, if the data points for 10 devices of the same type were already created and organized in 10 folders, one for each device, and a template should be designed for a device of this type, pick the folder containing the data points of device 1 and use these data points to design the template. If a data point would be selected from the folder of device 2 while the base path is set to device 1, this data point would be considered fixed and not relative, since it would not be located below the selected base path.

Consider the following when designing a new template:

- Try to place the controls at the position which is most likely the desired final position when the template is instantiated. This avoids the need to move each template instance to the desired location.
- Determine which data points are needed for the template. For variable data points, make sure they are organized in folders which can later be replaced for each template instance. When creating data points from a network scan, this kind of organization is usually done automatically, since a new folder will be created for each device. Add each of the variable data point folders from which the template is using data points to the list of data point base paths.
- If the template will be using instances of other templates, check which base paths these templates require and add them to the list of base paths for your own template, if they are not already covered by one of your existing base paths. Note that sub-folders are automatically covered. This is because the system will always try to find the best way to derive the base paths of template instances in a template from the available base paths of the parent template. For example, if the top level template uses base path A/B and there is a template instance which was designed for folder A/B/C, the base path of the sub-template will be set to C, relative to whatever the instance of the top level template uses for A/B. If the top level template is instantiated and the base path is changed from A/B to X/Y, the sub-template will use X/Y/C as its base path.
- If the template requires a set of internal data points, like registers, place them in a separate folder and add it to the base paths. Later on, this register folder must be copied for each template instance and the base path must be changed, so that each instance will access its private set of register data points.

9.15.3 Instantiating a Template

There are a few different methods to create actual instances of one or more templates on a page of the project:

- In the tree view, drag the template object and drop it on the page on which a new instance of this template is desired.
- From the context menu of the page, select *Add Template Instance*. A new template instance object will be created, but it will not be connected to a template yet. To assign a template to the instance, drop the desired template on the

instance object. This can also be used to switch a template instance from one template to another.

- To create instances of multiple templates in one step, place the related templates in a template folder, drag the entire folder, and drop it on the page where the instances should be created. This will create one instance of each template located below the template folder.

The new template instance looks and operates similar to an object collection. The symbol in the object tree is also similar to a collection, but the color is different. Similar to a collection, the template instance can be selected and the bounding box in the LCD preview can be moved to change the location of the objects inside the collection.

NOTE:	<i>When a template is instantiated for the first time, the symbol of the template in the object tree changes to indicate that this template is now in use. This helps to quickly locate unused templates which may be removed.</i>
--------------	--

Whenever a template instance is selected, there are two new property pages available:

- **Template / Instance:** On this property page, the desired *data point base folders* for this instance must be selected. It depends on the template, how many base paths are available. For each path, the path which was originally used to design the template is shown above the input field. This makes it easy to know which kind of data point folder is expected at this location. The number in front of the selected replacement folder is used when a template is instantiated inside another template (explained later on).
- **Data Point List:** On this property page, a list of all data points which are used by the template instance after reassignment to the new base folders is displayed for review. In case the new base folder is missing some of the required data points, the broken data point symbol will be shown next to the affected data point and the template instance in the tree view will be marked with a red triangle.

9.15.4 Changes to Data Point Folders

In some situations, it happens that changes occur to the data point folder hierarchy after templates and their instances were already created. There are two major changes which can affect templates and their instances:

- The name of a data point folder may change due to the user renaming the folder manually, or an indirect name change due to other configuration changes, for example renaming an imported OPC server.
- An already designed template needs to be converted to use data points from a different base path, because the original base folder is going to be removed from the project.

The first case affects template pages as well as template instances. Changing the name of a data point folder which is part of any base path, be it a template or instance base path, would break the affected template or instance. For a template, data points which should be relative to a certain base path are suddenly fixed, since they cannot be found under any of the configured base paths anymore. For a template instance, data points relative to the affected base path can no longer be resolved since the configured base path no longer exists.

The second case is related to template pages only. It requires reworking all data point references to use the replacement points from the new folder before the old folder is deleted.

Starting with version 6.1 of the configuration software, these situations are handled automatically to avoid broken templates and instances. Renaming a data point folder will automatically adjust all affected base paths to use the new name, such that the change has no negative effect on existing templates and instances. Converting a template to a different base path is simply done by changing the base path on the template property page. All data point references which are relative to the old base path will automatically be converted to reference a suitable data point below the new base path. In case a suitable data point cannot be found, the reference will be marked broken for manual inspection.

Also, base paths and remapped data points are constantly checked and template or instance objects which contain errors are marked in the tree view with a small attention icon, to let the user know that a template or template instance definition is broken.

9.15.5 Nested Templates

It is possible to build a nested hierarchy of templates by adding template instances to templates. This can be useful to build a template for a room, which contains templates for light, HVAC, and other subsystems, which again may use templates for individual devices and so on. The system will make sure that no loops are built, where for example template A contains an instance of template B, and template B would try to instantiate template A.

Whenever a template is instantiated inside another template, the base paths of the sub template are derived from the base paths of the parent template. This is because the user will only instantiate the top level template, and therefore only one set of base paths will be available. As a consequence, all sub templates have to get their base paths relative to the paths which were set for the top level instance. Based on the paths set for the instance of the sub template and the paths set for the design of the top level template, the system will automatically select the most suitable relative path. It is therefore important to use a consistent set of paths during the template design. The paths used for the sub template instances should normally be sub folders of paths which are used to design the parent template. Although this is not strictly required and thus not enforced by the software, it is in most cases the only meaningful configuration.

Consider the following example: The data point setup contains a set of data point folders, one for each room (room1, room2, room3, and so on). Each room folder contains a sub folder for lighting, for example room1/lighting or room2/lighting. Now a room template is designed with room1 as the base folder and a sub template to handle the lighting is instantiated. The sub template instance expects a base path for the lighting related data points (contained in the lighting folders). The only meaningful setting is to select room1/lighting as the base path for the lighting template instance. In this case, the sub template will use a relative base path of \$1(room1)/lighting, which means that it takes the path which was set for slot 1 of the room template instance and searches for a sub folder called lighting to access the lighting related data points.

If we would have selected room2/lighting as the base path for the instance of the lighting template inside the room template, which we are designing based on room1, the relative path for the lighting data points would become \$1(room1)/../room2/lighting, which means that the lighting template will take the base path 1 of the room template instance, move one folder level up and search for room2/lighting to access the lighting related data points. In most cases, this is not what we want and it is caused by inconsistent selection of base paths.

NOTE:	<i>As a rule of thumb for selecting base paths for sub template instances, always stay within the data point folder hierarchy which is set for the design of the parent template. If an instance of the sub template actually needs data from an unrelated folder hierarchy, it is usually required to add another base path to the parent template to make the new folder hierarchy accessible and settable by the user when the template is instantiated.</i>
--------------	---

The first part of any relative base path has the following notation:

\$N(folder)

A \$ sign, followed by the number of the base path of the parent template which was chosen as the basis for the relative path (\$1 to \$4), followed by the actual path (in parenthesis) which is currently set at this location. Since a template may have up to four different base paths, it is necessary to know to which one of them the sub template path is relative.

NOTE: *In case the base paths of the template are changed, the relative paths of all sub template instances are re-evaluated and adjusted for the best match based on the new set of base paths available. Also, if a base path of a sub template instance is changed, a new relative path will be calculated based on the available base paths of the parent template and the selected data point folder.*

9.15.6 Nested Base Paths

In special cases, it may be required for the template to operate on a basic set of data points, for example folder A/B which can be replaced by X/Y in another instance of the template, but at the same time use data points from A/B/C which should be replaced by X/Y/Z in the instance, instead of X/Y/C, which would be the normal result of replacing A/B with X/Y. Consider the following example of base paths for a template:

1. Room1/Sunblinds
2. Room1/Sunblinds/East

The template needs to use general data points regarding the sun blinds of a certain room. These data points are located in a folder called `Sunblinds` which is available for each room, so this folder needs to be one of the template base paths (1). However, the template itself uses a nested template instance to handle display of a certain blind in the room, for example the one facing east. The sub template needs a base path to the data points of a specific blind, in this example located in `Room1/Sunblinds/East`.

To instantiate this template to display the status of the sunblind facing **west** in room **4**, two base paths need to be reassigned. The one for the general sunblind data related to the room, and the one for a specific sunblind. Therefore the folder containing the specific sunblind data needs to be listed as a separate base path (2), even though it is a sub folder of the more general base path. For the above example, the following base paths can be used:

1. Room4/Sunblinds
2. Room4/Sunblinds/West

Without the separate base path (2), only the room could be remapped and the resulting template instance would always show data for the east blind of the specified room.

To allow this configuration to work as expected, each of the data points used in the template is assigned to the most specific base path. This ensures that all generic sunblind data points related to a room are assigned to base path 1, while data points related to a specific sunblind are assigned to base path 2, even though they would also be covered by base path 1.

NOTE: *Prior to version 5.0.0 of the configuration software, this setup was not allowed. However, using configuration software 5.0.0 or newer, such a setup is supported and can be used even with devices running older firmware.*

9.15.7 Local Changes

Once a template is instantiated, it is often required to make some minor adjustments to the appearance of the template instance to distinguish it from other instances of the same template. The most important adjustment is the display offset, which allows the same template to be instantiated on the same page multiple times at different locations. However,

there are a few other things which may need to be modified, for example colors or text for headlines.

While many aspects of the instance will be controlled by the referenced data points and the actually used data points will vary from instance to instance, it is not always feasible to control everything just via data points. Therefore, a template instance allows the **position, text, alignment, colors**, and some other settings to be changed on the respective property sheets.

Note that settings which are not changeable at the instance level are disabled. To check and modify such properties, go to the corresponding template, for example by using the command *Show Template...* from the context menu of the instance, and modify the properties in the template.

In order to keep the adjustments even when the template itself is changed, they are stored together with the template instance object. If a stored change is no longer applicable, for example because the object to which it applies was removed from the template, the change is discarded. If a locally changed value for a property is found in the list of changes, the locally stored value overrides the default settings from the template. For example, if a text control was originally displaying the text 'Room1' and this text was changed locally to 'Room101', the changed value will be used, even if the original text in the template is changed to 'Room2' later on.

NOTE:	<i>All objects of a template instance for which at least one local change is stored show a small symbol of a pen in their tree icon, to indicate that the object was modified from its original appearance. In addition, the template instance icon itself will be marked with a pen symbol if at least one of the objects was changed.</i>
--------------	---

To revert back to the defaults from the template, use the command *Restore Defaults* from the context menu of the template instance. This will delete all stored local changes. To delete all changes for an individual object, use the *Restore Defaults* command from the context menu of the object.

NOTE:	<i>It is also possible to indirectly take back individual changes. This happens when the property in question is set back to the same value as currently set in the template. The system will recognize this and remove the corresponding change from the list of locally stored changes. From this time on, changing the property in the template will cause the instance to take over the change again, since the local change which would otherwise overrule the template value was removed.</i>
--------------	---

9.15.8 Nested Changes

In case of nested templates (see Section 9.15.4), local changes are also nested. When this happens, changes on outer instances override changes done on sub-instance level. Consider the following example:

- Template A contains an instance of template B, and B contains a text control.
- On the instance of B in template A, the text color of the control is changed to red. The text control is displayed in red, the control and the icon of the template instance B in A show the pen symbol.
- Template A is now instantiated on a page, including its instance of B with the text color change, such that the text control is shown in red. The blue icon of the sub template instance B and the text control itself show the pen symbol. The icon of template instance A has no pen symbol, because there are no local changes to this instance yet.

- Now the text color is changed to blue in the instance of A. This change is stored together with the instance of A, so the pen symbol now also appears on the tree icon of instance A.
- When this instance is updated in the future, the original text control will first have its color changed to red, based on the change stored with the sub instance of B, and then further to blue, based on the change stored with the top level instance of A. The text control appears in blue.

If changes on different levels are not overlapping, e.g. are not changing the same property of the same object, they apply independent of each other. For example, if the local change on instance A sets the background color instead of the text color to blue, the text will appear red due to the change in instance B and the background will appear blue due to the change in instance A.

NOTE: *The change done to the text color on the instance of B in template A can only be removed there, in template A, not in the instance of template A. The command Restore defaults has to be called on the instance of B in template A.*

9.15.9 Base Path Placeholder Tags

To minimize the need for local changes of a template instance, information from the data point base path settings of the instance can be used when designing the template. Special placeholder tags allow the designer of a template to access the individual components of the four data point base paths and use them for example to change the contents of a text control or to modify the target of a show page action. This may eliminate the need to apply such changes to each instance manually. The data point base paths need to be set up anyway.

A placeholder tag starts with the special character sequence `${`, followed by the index of the base path to use (**b1** to **b4**), optionally followed by the path component to use (**f1** is the right-most path component, with higher indices moving up in the path hierarchy), and a closing brace `}` at the end.

For better understanding, the following section lists a few examples of possible placeholder tags and the resulting replacement when the template is instantiated. All examples assume a template instance with the first two data point base paths filled in like this:

1. L-Vis/Register/Floor1/Room101
2. L-Vis/Trend/HVAC/South-Wing

Base paths 3 and 4 may be used as well, but for the purpose of this illustration, we will only work with the first two of the four base paths. Possible placeholders and their result:

- `${b1}` Room101
- `${b1f1}` Room101
- `${b1f2}` Floor1
- `${b2} - ${b1}` South-Wing – Room101
- `${b2} - ${b2f2} ${b2f3}` South-Wing – HVAC Trend

Placeholder tags are currently accepted at the following locations:

- **Text Control:** The text field of a text control may contain placeholder tags. Note that you will want to turn off the *Auto-Resize* option and make sure the text field is wide enough to hold the replacement text, which depends on the data point path names assigned to the template instance later on. Also note that placeholder

tags are not supported in mapping table entries. The placeholder must appear directly in the text field of the text control.

- **Show Page Action:** The show page action, when used with a relative path to the target page, accepts placeholder tags in the path string. If the project is designed such that menu item or page names match up with data point folder names, a show page action of a template instance may be designed to automatically jump to a different target page, depending on the setup of the data point base path.

10 Data Point Configuration

As outlined in the description of the data point object in section 9.5.10.5, the network variables, BACnet server objects, client mappings, user defined registers, and system registers are represented by objects which exist on the device independent of any controls or other objects in the object tree. These objects are called data points and are referenced through the L-Vis data point objects described in section 9.5.10.5 by their unique ID.

While some of these objects are created automatically, most of the data points need to be created by the user. There are ways to manually create individual data points, import object definitions from CSV files or scan the network and create new data points based on the information collected by the network scan. Once created, a data point may be used to exchange data with other devices on the network, store and load internal registers, or access internal system parameters. Working with data points is the same for the entire line of LOYTEC products.

This chapter explains the use of the data point system on which L-Vis projects are built.

10.1 Data Points

10.1.1 Overview

Data points are part of the fundamental device concept to model process data. A data point is the basic input/output element on the device. Each data point has a value, a data type, a direction, and a set of meta-data describing the value in a semantic context. Each data point also has a name and a description. The entire set of data points is organized in a hierarchy.

At the data point level, the specific technological restrictions are abstracted and hidden from the user. Working with different technologies at this level involves common work-flows for all supported technologies.

The direction of a data point is defined as the “network view” of the data flow. This means, an input data point obtains data from the network. An output data point sends data to the network. This is an important convention to remember as different technologies may define other direction semantics. If a data point can both receive and send data on the network, its direction is set to value, indicating no explicit network data flow.

The basic classes of data points are:

- **Analog:** An *analog* data point typically represents a scalar value. The associated data type is a *double precision* machine variable. Meta-data for analog data points include information such as value range, engineering units, precision, and resolution.

- **Binary:** A *binary* data point contains a Boolean value. Meta-data for binary data points includes human-readable labels for the Boolean states (i.e., active and inactive texts).
- **Multi-state:** A *multi-state* data point represents a discrete set of states. The associated data type is a signed integer machine variable. Each state is identified by an integer value, the *state ID*. State IDs need not be consecutive. Meta-data of a multi-state data point includes human-readable descriptions for the individual states (state texts) and the number of available states.
- **String:** A *string* data point contains a variable-length string. The associated data type is a character string, encoded in UTF-8 for international language support. A string data point does not include any other meta-data.
- **User:** A *user* data points contains un-interpreted, user-defined data. The data is stored as a byte array. A user data point does not include any other meta-data. This type of data point also serves as a container for otherwise structured data points and represents the entirety of the structure.

10.1.2 Timing Parameters

Apart from the meta-data, data points can be configured with a number of timing parameters. The following properties are available to input or output data points, respectively:

- **Pollcycle** (input): The value is given in seconds, which specifies that this data point periodically polls data from the source. This is referred to as static polling.
- **Receive Timeout** (input): This is a variation on the poll cycle. When receive timeout is enabled, the data point actively polls the source unless it receives an update. For example, if poll cycle is set to 10 seconds and an update is received every 5 seconds, no extra polls are sent. The receive timeout setting cannot be used for feedback data points.
- **Poll-on-startup** (input): If this flag is set, the data point polls the value from the source when the system starts up. Once the value has been read, no further polls are sent unless a poll cycle has been defined.
- **Minimum Send Time** (output): This is the minimum time that elapses between two consecutive updates. If updates are requested more often, they are postponed and the last value is eventually transmitted after the minimum send time. Use this setting to limit the update rate.
- **Maximum Send Time** (output): This is the maximum time without sending an update. If no updates are requested, the last value is transmitted again after the maximum send time. Use this setting to enable a heart-beat feature.

Dynamic polling is an option that some network technologies offer. With static polling the poll cycle is used to permanently poll values over the network. This is required for data points that require constant value updates a fixed poll cycle (for example to trend the data). For other data points that do not need permanent value updates, so-called dynamic polling is activated, as soon as the values are needed. When the data is not needed, polling stops and no longer puts a burden on the network. The advantage is that a few data points can be refreshed at a higher rate at a time compared to static polling, where all data points must permanently share the available network bandwidth.

10.1.3 Default Values

Default values can be defined for data points when needed. The value of a data point will be set to the defined default value, if no other value source initializes the data point. Default values are beneficial, if certain input data points are not used by the network and need a pre-defined value, e.g., for calculations. Default values are overridden by persistent values or values determined by poll-on-startup.

10.1.4 Persistency

Data point values are by default not persistent. This means that their value is lost after a power-on reset. There exist different strategies for initializing data points with an appropriate value after the device has started.

For input data points, the value can be actively polled from the network when starting up. Use the Poll-on-Startup feature for this behavior. Polling the network values has the advantage that intermediate changes on the network are reflected. An input data point can be made persistent, if the last received value shall be available after a power-on reset before a poll-on-startup completes. This can be beneficial, if the remote device is temporarily offline and the last value is considered usable.

For output data points, the value can be restored after starting up by the application. For example, if the output data point's value is determined by an input data point and a math object, or the output data point is in a connection with an input, the input can poll its value on startup. If the output data point has no specific other value source, e.g., it is a configuration parameter set by the user, it can be made *persistent*.

To make a data point persistent, enable the Persistent property of the respective data point. The persistency option is only available for the base data point classes analog, binary, multi-state, string and user. More complex objects such as calendars, schedules, etc., have their own data persistency rules.

For structured data points, only all or none of the structure members can be made persistent. The configuration of the top-level data point, which represents the entire structure, serves as a master switch. Setting the top-level data point to be persistent enables persistency for all sub-data points. Clearing it disables persistency for all sub-data points.

NOTE:	<i>Persistent input data points may receive value updates after system start up; regardless of the data point update flag System Startup (see Section 9.6.2). This happens when a saved data point value is restored after system boot up. Data points which use updates as events to trigger certain actions should therefore not be set persistent.</i>
--------------	--

10.1.5 Parameters

A data point can be qualified as a *parameter* data point. This is accomplished in the configuration software by setting a **Parameter** check box on the data point. Those parameter data points are automatically persistent and will typically have a default value. Their purpose is to store parameterization values, which can be changed from the default value at run time and influence the behavior of the device or the logic running on the device. This way, a number of devices can have the same basic configuration and be adapted by parameter values. Examples are sunblind run times for control logic or descriptive strings for the L-WEB visualization.

The qualified parameter data points are also exported via a parameter file, which contains the entire set of current parameter values including meta-information for external tools to display parameter data in a human-readable way. The LWEB-821/900 master parameter editor can process such parameter data points and manage them for a large number of devices.

10.1.6 Behavior on Value Changes

The value of a data point can change, if it is written by the application or over the network. For all data points (input, output and value) the application (connection, user control, etc.) can be notified, when the value is written to. The property **Only notify on COV** defines, whether the notification is done with each write or only if the value changes (change-of-value, COV). If only notify on COV is disabled, writing the same value multiple times will result in multiple notifications.

NOTE: *A notification received by the L-Vis application from a data point may trigger execution of actions or commands like visibility or color changes, or it may result in a recalculation of a formula and cause a new notification about the changed result to be sent.*

When the value of an output data point is updated, an update is usually sent out onto the network. The property **Send-On-Delta** decides how the update is reflected on the network. If send-on-delta is inactive, each update of the value is sent, even if the value does not change. If send-on-delta is active, only value changes are sent. The send-on-delta property is only valid for output data points.

For analog data points, the COV or send-on-delta takes an extra argument, which specifies by what amount the value must change to regard it as a change for action. Both, COV and send-on-delta for analog data points check the **Analog Point COV Increment** property. A change is detected, if the value increment is bigger or equal to the specified increment. If the property is zero, all updates are considered.

10.1.7 Custom Scaling

Custom scaling is applied to all analog data points when they communicate values to or from the network. This feature can be used, if a network data point has engineering units not suitable for the application (e.g., grams instead of kilograms). The scaling is linear and applied in the direction from the network to the application as:

$$A = k N + d,$$

where N is the network value, k the *custom scaling factor*, d the *custom scaling offset*, and A the application value. When sending a value to the network, the reverse scaling is applied. If this property is enabled, the analog values are pre-scaled from the technology to the data point. The custom scaling is in addition to any technology-specific scaling factors and can be applied regardless of the network technology.

10.1.8 User Registers

The device can have user registers. User registers are data points on the device that do not have a specific technological representation on the control network. Thus, they are not accessible over a specific control network technology.

A user register usually serves as a container for intermediate data, persistent parameters, or data points which need to be accessed by other devices via global connections or OPC. The register can have the following, basic data types:

- **Double:** A register of base type *double* is represented by an *analog* data point. It can hold any scalar value. No specific scaling factors apply.
- **Signed Integer:** A register of base type *signed integer* is represented by a *multi-state* data point. This register can hold a set of discrete states, each identified by a signed stats ID.
- **Boolean:** A register of base type *Boolean* is represented by a *binary* data point. This register can hold a Boolean value.
- **String:** A register of base type *string* is represented by a *string* data point. This register can hold a variable-length character string in UTF-8 format.
- **Variant:** A register of base type *variant* is represented by a *user* data point. This register can hold any user-defined data of up to a specified length of Bytes. This length is defined when creating the register and cannot be changed at run time.

Since a register has no network direction, it can be read and written. Therefore, two data points are generated for each register, one for writing the register (output) and one for reading the register (input). A suffix is added to the register name to identify the respective data point. For example, the register *MyValue* will have two data points connected to it: *MyValue_Read* and *MyValue_Write*. If supported by the device firmware, the user can

choose to generate one data point of direction *value* instead of two separate data points. This data point can be read and written, so the desired direction will have to be specified at each place the data point is used, whereas it is implicitly known with separate data points.

10.1.9 Structures

Complex data belonging semantically together may be structured. The data point model allows mapping structure types onto user-defined data points of *variant* type. This can be necessary, if a network technology carries such structured data or if a user-defined register shall provide structured data for access through a single data point. In any case, the structure is modeled as a top-level data point and a hierarchy of sub-data points representing the structure members.

The top-level data point is a user data point of variant data type. It contains the image of the entire structure as a Byte array. Each structure field is then modeled as a sub-data point of the appropriate class (e.g. analog, binary, or multi-state). A structure field may itself be a structure going down one level in the hierarchy of sub-data points.

An example is shown in Figure 3. In this case a user register of two Bytes is bound to a structure type mapping the two bytes on analog data points. The two sub data points *byte_0* and *byte_1*.


example2_Read	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	In
byte_0	1.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	In
byte_1	1.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	In

Figure 3: Example of a structured data point.

The structure types are available in a type repository. This repository is divided into scopes. Within each scope a type has a unique name. When selecting a type, the scope and the type name needs to be specified.

10.1.10 Property Relations

A data point possesses a number of properties which influence the behavior and appearance of the data point. Examples are data point name, poll cycle, or alarm limits. Most of those properties are determined by the configuration and are static during operation of the device. Some of those properties, however, can get a default value from the configuration and be modified during run-time later on.

In some cases, property values need to be updated by other data points, e.g. a user register or a technology data point. In this case, the data point property is linked to another data point, following a given semantic relation. This is modeled as a *property relation*. Property relations appear as data point links with the respective property names underneath their governing data point. An example is shown in Figure 4. Relations are marked with a link symbol . When hovering with the mouse over the link symbol, a bubble help appears describing the property relation.





reg_bool_fb_alarm_Read	7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	In	
feedbackValue -> User Registers.master_feedback_Read	7.1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	In	
enableAlarm	7.2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Value	
inAlarm	7.3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Value	


Figure 4: Example of property relations.

The property relations can be accessed like regular child data points. For this type of use, no linkage against other data points is necessary. Property relations may, however, also be linked to other data points, e.g. 'feedbackValue' in Figure 4. In this case, the linked data point is used as the related property. The user may right-click on a linked property relation and choose **Go to related data point** from the context menu. For mass engineering property relation links to other data points refer to Section 10.6.9.

The following properties are available as property relations:

- **feedbackValue:** This property relation is used for feedback alarm conditions. The data point value is compared against the feedback value. An alarm is generated, if these values differ (by a certain amount). It exists only, if an alarm condition has been created.
- **enableAlarm:** This property relation is used to enable or disable alarm generation on the data point. It exists only, if an alarm condition has been created.
- **inAlarm:** This property relation is TRUE, if the data point is in an alarm. It exists only, if an alarm condition has been created.
- **ackPend:** This property relation is TRUE, if the data point's alarm needs acknowledgement. It exists only, if an alarm condition has been created.
- **highLimit:** This property relation defines the high limit for analog alarms. It exists only, if an alarm condition has been created.
- **lowLimit:** This property relation defines the low limit for analog alarms. It exists only, if an alarm condition has been created.
- **deadband:** This property relation defines the dead band for analog alarms. It exists only, if an alarm condition has been created.
- **nativeAlarm:** This property relation links to a technology data point, which is required for alarms reported to another technology. It exists only, if an alarm condition has been created and alarms are reported to the given technology (e.g. BACnet). This property relation cannot be modified by the user.
- **reportTo:** This property relation exists only in generic alarm servers. It may be linked to a technology alarm server to report alarms to that network technology.
- **totalActive, totalUnacked, totalAcked:** These property relations exist only in alarm servers. They contain counters for active unacknowledged, inactive unacknowledged, active acknowledged alarm records of the alarm server, respectively.
- **ackAll:** This property relation exists only in alarm servers. When writing TRUE, all alarms on that alarm server are acknowledged.
- **historicFilter:** This property relation exists for data points that have at least one historic filter assigned (see Section 10.3.5).
- **enable:** For Schedulers can be enabled or disabled by the use of this property relation. If the enable data point has been defined in the scheduler's configuration, this property relation is linked to that data point.
- **enableFb:** This property relation shows the enable state of the scheduler. If the enable feedback data point has been defined in the scheduler's configuration, this property relation is linked to that data point.
- **presetName:** This property relation of type string shows the preset name of the currently scheduled value. If the preset name data point has been defined in the scheduler's configuration, this property relation is linked to that data point.
- **timeToNext:** This analog property relation contains the number of minutes till the next scheduled state changes. It can be used to implement an optimum start algorithm.
- **nextState:** This property relation contains the next scheduled state. It can be used to implement an optimum start algorithm.
- **nextPresetName:** This property relation contains the preset name of the next scheduled state, if such name exists. Otherwise it stays at invalid value.

10.1.11 Engineering Units

Analog data points have an engineering unit property that defines the physical unit of the underlying scalar value. The engineering unit is displayed to the user as a human-readable text which can be selected from a drop down list of available units. Once a valid engineering unit was selected, a green checkmark appears .

Units are linked to additional meta-information in the metric (SI) or U.S. unit system. Based on this data, the configuration software offers compatible units of the respective other unit system, of different magnitude, or different usage. Important properties of convertible units when used are:

- An alternative display unit can be configured that will be displayed on the web interface of the device.
- Automatic unit conversion in local connections is performed, if data points with compatible convertible units are connected (e.g. '1000 W' are converted to '1 kW'). No custom scaling is required.

10.2 Connections

10.2.1 Local Connections

Through the use of local connections, values can be transferred between data points within the device. Various types of connections – from “1:n” to “m:n” connections – are supported. Data points added to a connection specify whether they feed a value into the connection (send) or receive a value from the connection (receive).

This means that the following connections are possible:

- 1 input data point is connected and writes to n output data points
- m input data points are connected and write to 1 output data point
- m input data points are connected and write to n output data point

The most common connection will be the 1:1 connection.

In the 1:n connection the input value is distributed to all n output data points. In the m :1 connection, the most current input value is written to the output data point. When polling the output data point in poll-through mode (maximum cache age is set on the output), the value from the first input data point is polled. The same holds true for a m :n connection. The default data flow of data points in a connection is a result of the data point direction. This can be overridden by a custom setting (i.e. an output data point can be configured as an input to the connection).

Connections can connect data points of different technologies with each other (also mixed among the target data points). When connecting data points of different classes, the exchanged values need to be converted. The connection inherits the type of the first data point class. If data points of a different class are added to this connection, an *adapter* needs to be defined. For example when an analog input is connected to a multi-state output. Conversion adapters can be saved in a library and reused later for similar conversions.

The following conversions apply for default adapters:

- **Analog to Analog:** The value range is capped on the output data points. This means, if the input value in the hub does not fit into the range of an output data point, the value is capped to the biggest or smallest allowed value. If the input and output data points both have convertible units the value is converted. The user can also specify a simple math formula as an adapter. In this case no implicit unit conversion is performed.

- **Binary/Multi-state to Analog:** The Boolean or state value is directly converted to an analog value (e.g. state ID '4' is written as '4.0') as a default. The user should specify an adapter to map the Boolean or state value to designated analog values.
- **Analog to Binary/Multi-state:** As a default the analog value is converted to the next Boolean or state value (e.g. '1.2' is written as state ID '2'). The user should specify an adapter with its own translation of value ranges to state values.
- **Multi-state to Multi-state:** Multi-state data points that have different state maps lead to a conversion of their state values. The state maps of inputs and outputs are ordered by state ID in ascending order. The state value of the input is then ranked as the n-th state and propagated over the connection. For example, the input state ID '1' is the 2nd state and the output's 2nd state has the state ID '0'. If the output data point has fewer states than the input, the output state is limited to its highest state ID. The user should specify an adapter that defines which input state maps to which output state.
- **Binary to Binary:** Binary data points can be connected without conversion.
- **String to String:** String data points can only be connected to string data points.
- **User to User:** User data points can only be connected to user data points. If the length is different, only valid bytes are written or excess bytes are truncated, respectively.
- **SNVT_switch to Analog/Binary/Multi-state:** The user data point of a SNVT_switch can be connected to analog, binary, and multi-state data points.
- **Analog/Binary/multi-state to SNVT_switch:** Analog, binary, and multi-state data points can be directly connected to a SNVT_switch user data point.

10.2.2 Multi-Slot Connections

Connections between structured data points often need to connect each structure member separately. To increase the overview in the project on the involved, single connections, a *multi-slot connection* can be created for local connections. This is a connection with several slots for transporting separated values over the connection. Each slot has a number and a name and can connect two or more data points. Data points added to other slots do not share their values across slots. One can think of such a connection as a cable with many wires. An example is shown in Figure 5 (a). The data point 'IN A' sends its value to 'OUT X' but not 'OUT Y'.

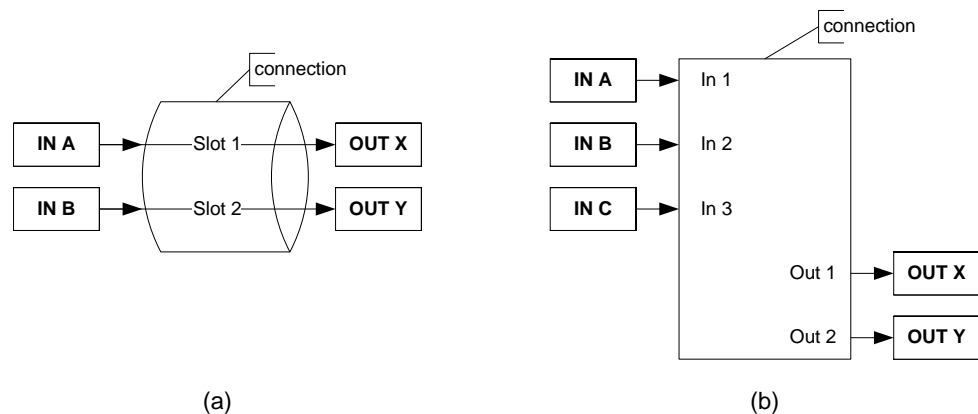


Figure 5: Multi-slot connection (a) and multi-slot with math block adapter (b).

Some applications also require a functional mapping between different data point structures in one connection. A multi-slot connection can be used with a math block adapter to accomplish this task. A math block has n inputs and m outputs. The multi-slot connection

has a slot for each input and output, which can be connected to the respective data points as depicted in Figure 5 (b). For this multi-slot connection the math block adapter defines a fixed layout of the slots; no more slots can be added to this connection.

In a math block adapter with n inputs v_1, v_2, \dots, v_n each output o_i is calculated as a formula depending on all inputs $o_i = f_i(v_1, v_2, \dots, v_n)$. Each output has two math formulae following the same format as used in math objects (see Section 9.13):

- **Output value formula:** This formula calculates the output value as a function of all input values.
- **Output enable formula:** This formula calculates an output enable (result > 0 is enable) for the output. If the output is enabled, the output value will be written to the output. If the output is disabled, the calculated output value is not written to the output.

In addition, each input slot can be configured whether it shall trigger the calculation or not. Normally, any change in any input triggers the calculation of all outputs.

10.2.3 Global Connections

Global connections provide the same notions as local connections but extend beyond the scope of one device. A global connection establishes a data cloud with a system-wide name. Data points added to a global connection can send data into that connection or receive data from the connection. The data is transferred over an IP-based network. All data is automatically matched by the global connection name. This makes global connections especially useful to provide certain global data in a system, without knowing who will be reading that data. Examples are weather station data, wind alarms or global on/off.

Global connections cannot use adapters for conversions as done in local connections. If conversions are needed, an intermediate register data point must be used to receive/send data from/to the global connection. The adapter needs to be installed with a local connection between the register and the data point, which requires the conversion.

The system in which the data cloud of a global connection is established is defined by an IP-852 channel. This channel is not related to the CEA-709 technology; it is purely used to define the set of devices exchanging data through global connections. It can, however, coexist with an IP-852 channel for CEA-709. The configuration of the IP-852 channel is done by adding all devices taking part in data exchange via global connections to a configuration server.

A global connection has the following properties:

- **Max Send Time:** This timing parameter of the global connection specifies a time in seconds, in which a value update is transmitted into the connection, even if not value has changed. This is typically used for heartbeat functions.
- **Min Send Time:** This timing parameter of the global connection specifies a time in seconds, for which transmissions will be delayed after sending out a value into the connection. This setting can be used to limit the transmission rate to the connection.

The following properties are derived from the data points in a global connection:

- **Receive Timeout:** A data point with a receive timeout will be put into the state offline, if it does not receive a value within the specified period of time (see Section 10.1.2). This also applies to values received from the global connection.
- **Poll on startup:** If a data point in the global connection has the poll on startup feature enabled (see Section 10.1.2), an initial value update will be triggered for the global connection.

How a global connection is created and configured is described in Section 10.8.5. Note that the number of configurable global connections on a device is limited (see Section 15.11).

10.3 AST Features

10.3.1 Alarming

The alarming architecture comprises a number of entities. Objects that monitor values of data points and generate alarms depending on an *alarm condition* are called *alarm sources*. The alarms are reported to an *alarm server* on the same device. The alarm server maintains a list of alarm records, called the *alarm summary*. The alarm server is the interface to access the local alarms.

Generic alarm servers provide the maximum set of alarming. Data points of all network technologies can be alarmed through generic alarm servers. Technology alarm servers can be used to expose access to the alarms to network technologies that support it. Generic alarm servers can be configured to report their generic alarms to technology alarm servers. For example, a generic alarm server may report its alarms to both CEA-709 and BACnet alarm servers.

An alarm record contains the information about a specific alarm. This includes information about the alarm time, the source of the alarm (i.e., which data point caused the alarm), an alarm message, an alarm value, an alarm type, an alarm priority, and an alarm state. An alarm record undergoes a number of state changes during its life-cycle. When the alarm occurs, it is *active*. At this point the alarm time, alarm message, alarm value is notified using the alarm priority. When the alarm condition subsides, the alarm becomes *inactive*. At this point the clear time and the clear message is notified using the normal priority. The priority levels are configurable on the alarm server, where 0 is the highest and 255 is the lowest priority.

Alarm transitions (to an alarm state, to the normal state) can be acknowledged by an operator. Which of those transitions requires an acknowledgement is configurable on the alarm server. If an active alarm is acknowledged it becomes *active acknowledged*. Active alarms can also become inactive, but an acknowledgement is still required. Then they become *ack-pending*. When an alarm is inactive and was acknowledged it finally disappears from the alarm summary.

An alarm state can be of different alarm types. The alarm type specifies the class of the alarm. The following alarm types exist:

- **Off-Normal Alarm:** This alarm type is a generic alarm class that applies to binary and multi-state alarm conditions. It indicates that the alarmed data point is on an off-normal operating condition that triggered the alarm. An alarm value is supplied. In technology alarm servers, restrictions may apply.
- **High/Low Limit Alarm:** This alarm type is typical for analog alarm conditions. It applies when the alarmed value is over or under the defined alarm limits. An alarm value is supplied. In technology alarm servers, restrictions may apply.
- **Fault Alarm:** This alarm type is indicating that the monitored data point is in a fault state. This is different from off-normal or high/low limit alarms. The value of the data point is within the specifications of the alarm condition but the data point itself is considered faulty. This can stem from an unreliable value or an offline value, i.e., if the data point is offline. No alarm value is supplied.

Alarms may be generated from a given data point value (alarm value or value range) or by comparing a data point command value with a feedback value (feedback alarm). When defining a feedback alarm, the alarmed data point represents the command value and has a 'feedbackValue' property relation (see Section 10.1.10). This property relation can be linked to another data point, which effectively provides the feedback value.

Alarmed data points also possess other property relations. The 'enableAlarm' property relation can be used to disable or enable alarm conditions when linked to a data point. The property relations 'highLimit', 'lowLimit', 'deadband' can be used to modify analog alarm conditions. The property relations 'inAlarm' and 'ackPend' are TRUE if a data point is in an alarm state or needs acknowledgement, respectively.

When a data point is alarmed by a generic alarm server, which reports to a technology that requires a dedicated technology data point (e.g., an alarm for a user register is reported to BACnet), the required data point is automatically created and linked via the 'nativeAlarm' property relation.

Alarm server objects possess property relations that provide a counter value of active unacknowledged, active acknowledged, and inactive unacknowledged alarms. These property relations may be linked to other data points that can be used to process this information.

Other devices can access the alarm information through a technology alarm server or the Web service. These devices are *alarm clients*. They register with the alarm server and get notified about changes to the alarm summary. Alarm clients can be used to display the current alarm summary and to acknowledge alarm transitions. Depending on the underlying technology, some restrictions may apply to the available alarm information and acknowledgement behavior. Refer to the technology sections for more information.

10.3.2 Scheduling

Schedulers are objects that schedule values of data points on a timely basis. A scheduler object is configured by which data points it shall schedule. This configuration is done by the system engineer once, when the system is designed. The configuration of the times and values that shall be scheduled is not part of that initial configuration and may be changed later. This distinction has to be kept in mind.

A scheduler object sets its data points to predefined values at specified times. The function of the scheduler is state-based. This means, that after a given time, the scheduler maintains this state. It can re-transmit the scheduled values as appropriate (e.g., when rebooting). The predefined values are called *value presets*. A value preset contains one or more values under a single label (e.g., "occupied" schedules the values { 20.0, TRUE, 400 }).

Which value preset is scheduled at what time is defined by a *scheduled event*. The event defines the starting time, value preset and end time in a 24-hour period. Events can be one-time events or recurring events. A schedule typically consists of a number of recurring and one-time events, for instance one event for the weekdays Monday through Sunday. See Figure 6 for an example of a schedule in a given calendar week.

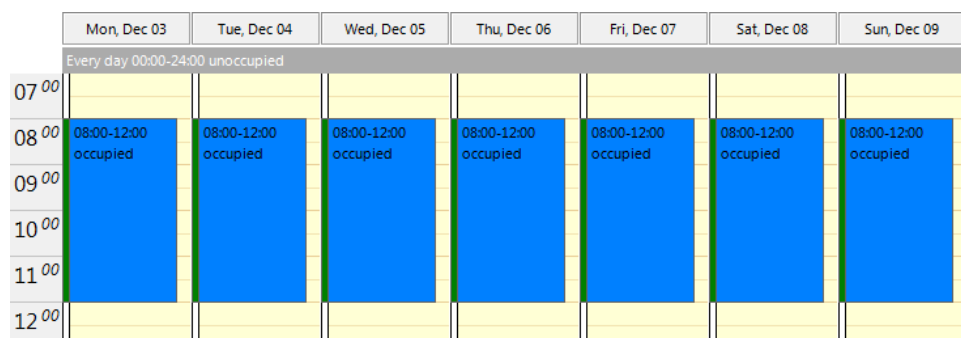


Figure 6: Example of a recurring event in a schedule.

How scheduled events are recurring can be defined by choosing the appropriate event type:

- **One-time:** This event occurs exactly on one defined date.

- **Daily:** This event occurs every day, starting at a given date and ending at a given date.
- **Weekly:** This event occurs every week on the specified weekday.
- **Monthly:** This event occurs every month following a date range or a defined rule (e.g. every last Friday).
- **Yearly:** This event occurs every year following a date range or a specific day every year.
- **Default:** This is a special event. The selected preset value will be in effect 00:00 to 24:00 hours every day if no other event occurs.
- **Calendar:** For some tasks the regular recurrence such as on weekdays is not sufficient. This can be implemented by defining events based on a *calendar*. For instance, there may be a calendar for holidays. The calendar contains a number of *calendar patterns*. Each calendar pattern describes a pattern of dates on which an event shall occur, e.g., *Holidays*.

One can define a set of scheduled events that are recurring differently. For example one event is defined for regular workdays (Monday through Friday). Another event is defined based on the holidays calendar pattern. This will lead to overlapping events between workday and holiday for those weekdays, which are holidays.

The resolution of this overlap is simple: Each event is configured with a *priority*. Should an overlap occur, the event with the higher priority will be in effect (e.g., Dec 25th in Holidays overrides the regular workday event). An example is shown in Figure 7. The detailed view shows the two overlapping events and the preview shows the effective schedule. Note, if two events with the same priority exist, it is not defined, which one is in effect. Therefore, always use distinct priorities.

Priorities are numbers, but some priorities have been pre-assigned, e.g. highest, override, normal, low. Please also refer to the technology-specific limitations described in Section 10.10 to learn about special behavior of the respective networking technology.

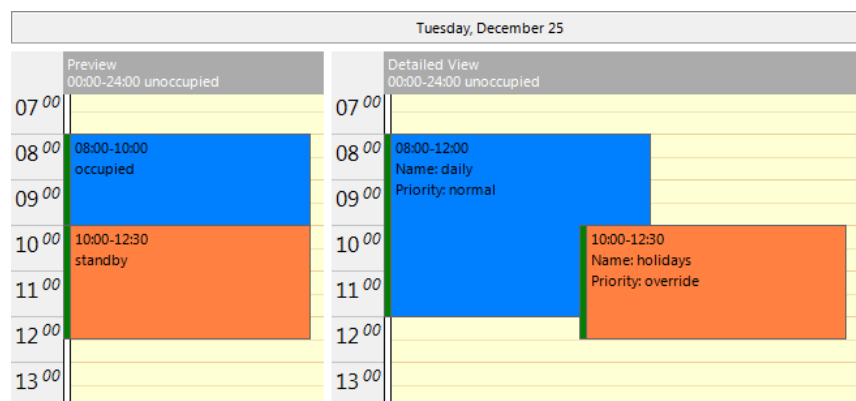


Figure 7: Example with overlapping events and different priorities.

The configuration of calendar-based recurrence is done by calendar patterns in the calendar. Each calendar pattern contains a number of pattern entries. These entries can define the following:

- A single date: This defines a single date. Wildcards may be used in the year to specify Dec 25th of every year.

- A date range: This defines a range. Starting with a start date and ending with the end date. No wildcards should be used.
- A Week-and-Day definition: This defines dates based on a week, such as every 1st Friday in a month, every Monday, every last Wednesday of a month.

A schedule defines at which time instants certain states of the scheduled data points are maintained. The *next-state* feature allows looking ahead into the future and predicting when the next scheduled state will occur. There are two data points involved: the time-to-next-state is a counter in minutes to the next scheduled event, and the next-state data point is the state of the next scheduled event. This information can be used by controllers to optimize their algorithms (e.g., pre-heat a room for the scheduled occupancy state). Use the `SNVT_tod_event` in CEA-709 to accomplish this task.

When a scheduler is executing the schedule on the local device, it is called a *local scheduler*. Such a scheduler is configured to schedule data points and later its daily schedules can be modified. When accessing the daily schedules of a scheduler, which executes on a remote device, the object is called a *remote scheduler*. A remote scheduler has the same interface to the user to modify daily schedules. A remote scheduler object can be used as a user-interface for schedulers that execute on different devices.

10.3.3 Trending

Trending refers to the ability to log historical values of data points over time. A trend log object is responsible for this task. The generic trend log object provides the maximum set of features. It can be configured to record historical data of any data point on the device. Log records are generated either in fixed time intervals, on change-of-value (COV) conditions, or when a trigger is activated. The fixed intervals can be optionally aligned to the wall time (e.g., to the top of the hour). After a reboot, the recording is resumed at the aligned intervals. Trend log objects can trend either local or remote data points. Technology trend log objects can be used to record historical values of the respective technology data points and expose them to network technologies that support it. These historic logs are separate from the generic trend logs and certain restrictions of the technology may apply.

The trend data is stored in a binary format on the device. The capacity of a given trend log is configured. The trend log can be operated in one of two modes: In *linear mode* the trend file fills up until it reaches its capacity. It then stops logging. In *ring buffer* mode the oldest log records are overwritten when the capacity is reached.

Devices with SD cards also allow backups of the trend logs on external Flash storage. This backup can be triggered by the user over the LCD display or be triggered by certain actions. The trend data is stored in CSV format under a folder identifying the device by serial number and the trends sub-directory, e.g. '016101-8000000DEA51/trends'. The SD card can be used on different devices. In this case different device directories will be created. The trend backup files can be opened directly on a PC. The backup on external storage can be enabled individually per trend log.

A fill-level action can be activated, whenever the trend log has logged a percentage of its log size with new log records. A fill-level condition of 70% on a trend log with 1000 items capacity will activate the fill-level trigger every 700 logged records. This trigger can be used to send E-Mails or backup trend data on external storage if available.

Trended data points can be logged as their actual values at given time instants or as an aggregated value over the defined log interval. Aggregation can be calculated as minimum, maximum, or average. Aggregation can be beneficial, if the trended value changes more frequently than the selected log interval. Using aggregation, the log interval can be chosen to limit the amount of logged data while preserving information of the trended value.

For technology trend log objects, certain restrictions apply as to how many data points can be trended in one trend log and which trend modes are available. Refer to the technology sections for more information.

10.3.4 E-mail

The e-mail function can be combined with the other AST features. The format of an e-mail is defined through *e-mail templates*. An e-mail template defines the recipients, the e-mail text, value parameters inserted into the text and triggers, which invoke the transmission of an e-mail. An e-mail template can also specify one or more files to be sent along as an attachment.

A prerequisite to sending e-mails is the configuration of an e-mail account on the device. This can be done through the web interface or in the project settings (see section 11.6). It is recommended to use the e-mail server of your internet service provider or the company SMTP server on the intranet. For public mailers, enable the required authentication. SMTP over SSL as well as TLS is supported for using Hotmail, Gmail, or Yahoo.

The amount of generated e-mails can be limited using a token bucket algorithm. The transmission of e-mails can be disabled altogether by using a special data point. That data point can be scheduled or modified over the network.

If an e-mail cannot be sent (e.g. the mail server is not reachable), the mail delivery is retried up to 24 times every 30 minutes.

10.3.5 Historic Filters

For certain applications historic values of a given base data point, both recent and far into the past, can be of interest. This can be accomplished with *historic filters*. Historic filters allow processing historic values of the base data point according to a filter function. One or more such functions can be defined per base data point. The result of the historic filter is written to *historicFilter* property relations. For each historic filter function a time period can be defined at which the base value is sampled, e.g., every first of the month at midnight, and how many samples ago. Historic filters can be created for any analog, binary, or multi-state data point. It is not necessary to create a trend log.

The following sampling periods can be defined:

- Value every x minutes aligned to full hour ($x = 1, 2, 5, 10, 15, 20, 30$ min), 0 or 1 samples ago,
- Hourly value at full hour, 0..24 samples ago,
- Daily value at HH:MM:SS of the day, 0..60 samples ago,
- Weekly value at HH:MM:SS on weekday (Mon..Sun), 0..10 samples ago,
- Monthly value at HH:MM:SS on day of month (1..31), 0..24 samples ago,
- Yearly value at HH:MM:SS on DD/MM of the year, 0..5 samples ago.

By using historic filter data points it is possible to implement numerous calculations on historic values of the base data point. For example it is possible to create two filter data points with a daily sampling period recording the energy consumption at midnight, one holding the most current sample (today at midnight) and the other the previous sample (previous day at midnight). This is shown in Figure 8. Those historic filter relations can be processed in a math object to calculate the consumption of the previous day.

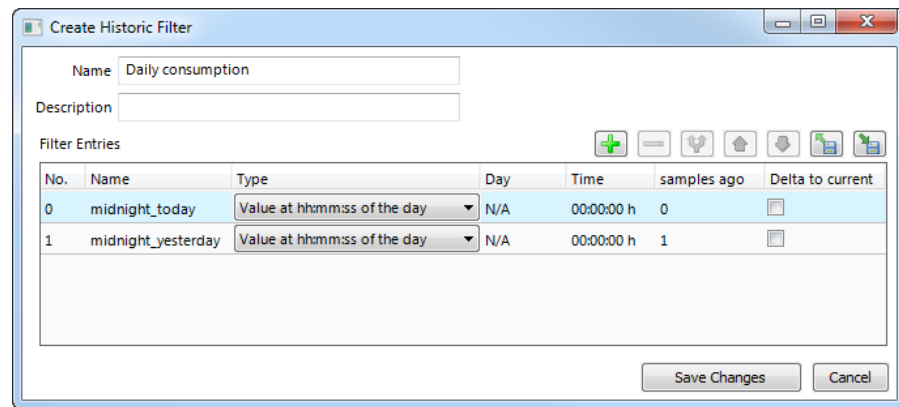


Figure 8: Example historic filters for daily consumption.

For calculating the difference between the current value and any historic value, the filter definition can be configured to delta mode. This is a shortcut to creating a math object processing the historic filter data point and the underlying data point. The results of the math object for the previous day and the consumption to-the-hour of the current day are available in data points, which can be visualized, alarmed or trended.

The historic filters definitions are managed by historic filter resources. These are templates and stored in the project resources. They can be applied to data points. When editing an historic filter template, all existing historic filter relations are updated accordingly. For more information on how to configure historic filters please refer to Section 10.15.

10.4 CEA-709 Technology

10.4.1 CEA-709 Data Points

Data points in the CEA-709 network are known as network variables (NVs). They have a direction, a name, and a type, known as the standard network variable type (SNVT) or user-defined network variable type (UNVT). In addition to NVs, also configuration properties (CPs) in the CEA-709 network can be accessed as data points. Both standard CP types (SCPTs) and user-defined CP types (UCPTs) are supported.

The CEA-709 NVs on the device can be created in three different ways:

- **Static NV:** For each selected NV on the network, there is a static NV created on the device. This NV can be bound to the NV on the network. Note that adding static NVs to the device results in a change to the default XIF file. The device is assigned a new model number to reflect this change (see Section 10.4.7). Static NVs are the way to use NVs in systems where bindings are used for the NVs instead of polling.
- **Dynamic NV:** Compared to static NVs, dynamic NVs do not change the XIF interface of the device. Dynamic NVs are created by the network management tool. Currently, only LNS-based tools can manage dynamic NVs. As for static NVs, with dynamic NVs it is possible to use bindings instead of polling.
- **External NV:** The selected NVs on the network are treated as external NVs to the device. The device doesn't create any NVs on the device, but instead uses polling to read from those NVs and explicit updates to write to the NVs. Therefore, no bindings are necessary for external NVs. For input data points using external NVs, a poll cycle must be configured. If not configured explicitly, a default poll cycle of 60 seconds is chosen. Note that the receive timeout option cannot be used with external NVs.

Based on the NV the data point is derived from, the following kinds of data points are created:

- Simple NVs that hold only one scalar value, e.g., SNVT_amp: Those kinds on NVs are represented as analog data points. The data points holds the current value, NV scaling factors are applied.
- Simple NVs based on an enumeration, e.g., SNVT_date_day: Enumeration types result in multi-state data points. They represent the state of the NV.
- Structured NVs that consists of a number of fields, e.g., SNVT_switch: All structured NVs are represented as user point. That is, the data point is structured similar to the NV it is based on. Beneath the user data point, the individual structure fields are presented as child data points.

10.4.2 CEA-709 Schedulers

CEA-709 schedulers and the CEA-709 calendar adhere to the LONMARK standard objects. For CEA-709, certain restrictions exist that need to be kept in mind. Attached data points can either represent an entire NV or individual elements of a structured NV. CEA-709 schedulers may have several different data points attached, i.e., the value preset may consist of more than one element. For example, a CEA-709 scheduler might schedule a SNVT_temp and a SNVT_switch and have 3 elements in each value preset as depicted in Figure 9. To control similar data points synchronously, they may be added to the same group to share the same value preset.

Datapoint	Description	Location	Group	Default	day	night
nvo_setpoint		LINK-110.CEA709 Port.Datapoints -	▼	0.00	21.00	16.00
nvo_switch.value		LINK-110.CEA709 Port.Datapoints -	▼	0.00	0.00	50.00
nvo_switch.state		LINK-110.CEA709 Port.Datapoints -	▼	0.00	0.00	1.00

Figure 9: Example value presets in CEA-709 schedulers.

Priorities of exception days in a CEA-709 scheduler range from 0 (the highest) to 126 (the lowest). The value 127 is reserved as a default for weekdays.

Further, the implementation as LONMARK standard objects requires the use of configuration properties. If the number of CEA-709 schedulers or their capacities for daily schedules and value presets is changed, the resource and static interface of the CEA-709 port changes. The resources reserved for LONMARK calendar and scheduler objects can be changed in the network settings (see section 10.10.10). When downloading a project, the software verifies if sufficient resources have been configured. If it detects a problem, the user is notified to update the resource allocation as needed.

NOTE: *The Auto-Set button adjusts the resources such that the **currently known** configuration will fit. Any additional resources required to add more schedules during runtime need to be considered and added manually to the automatically determined values.*

10.4.3 CEA-709 Alarming

Local CEA-709 alarming supports only one alarm server object. This alarm server object is represented by the device's LONMARK node object and facilitates the SNVT_alarm2 output network variable. Acknowledging alarms in the alarm server is adhering to the LONMARK specification and relies on the RQ_CLEAR_ALARM mechanism.

10.4.4 CEA-709 Trend Logs

Due to the absence of a common standard for trend data transfer over CEA-709 networks, there is currently no implementation of a CEA-709 trend object available. CEA-709 devices like the LVIS-3E100, 3E112, and 3E115 use generic trends and expose them via OPC only.

10.4.5 Dynamic Polling in CEA-709

External input NVs in CEA-709 rely on polling. Static polling can be configured by setting the poll cycle (see Section 10.1.2). In addition, this technology also supports dynamic polling. When dynamic polling is enabled, polling is only active for those NVs for which an external OPC client, the current page displayed on the LCD screen, or the current page displayed on the web interface requires a refresh. See also section 11.5.3 for details on how to enable dynamic polling for a project.

Dynamic polling has no effect on static or dynamic input NVs. These NVs are supposed to have bindings and rely on update events. If regular polling is configured via the poll cycle, no change is made at run-time for dynamic polling.

10.4.6 CEA-709 Data Points in Connections

All types of CEA-709 data points can be used in local and global connections. There is a special option for SNVT_switch, which allows the state member to be represented as a binary data point, making it easier to use in connections to other network technology.

10.4.7 Static Interface Changes

The device can be configured to use static NVs. Unlike dynamic NVs, static NVs cannot be created in the network management tool. They are part of the static interface and are usually compiled into the device. When static NVs are used, the device changes its static interface and boots with a new one.

See section 12.3.3 for more information about the procedures which apply when the static interface of the device is changed.

10.5 BACnet Technology

10.5.1 BACnet Data Points

Data points in the BACnet technology are known as BACnet objects. They have a specific type (e.g. analog input or binary output) and a set of properties which describe the data point more closely. The actual value is stored in the property called *Present_Value*.

On the device, there exist two classes of BACnet data points:

- **BACnet server objects (SO):** These BACnet objects are defined by the configuration software to be allocated *locally* on the device. These objects can be accessed by the BACnet building control system, operating workstations, or other devices on the BACnet network. They support COV subscriptions to deliver value changes in an event-driven way. Local server objects can be created as AI, AO, AV, BI, BO, BV, MI, MO, or MV objects.
- **BACnet client mappings (CM):** Since LVIS devices are used to display and control values of other devices in the network, they are typically operating as BACnet clients. This functionality is available through the use of *client mappings*. Client mappings can be of the type *Poll*, *COV*, *Write*, *Auto* or *Value*. This specifies how the BACnet client accesses other BACnet objects on the BACnet network. *Poll* is used for objects that need to read data from other BACnet objects in a periodic manner. *COV* is used to subscribe for COV at other BACnet objects in order to get updates in an event-driven fashion. *Write* is used to send updates to other BACnet objects. *Value* refers to a combined read and write client mapping. The *Auto* setting determines the best fitting method (poll, COV, or write). When writing a value to this client mapping, the value is written to the remote BACnet object. As soon as the *Present_Value* of the remote BACnet object is updated, the value is transferred back.

The direction of BACnet server objects deserves a closer look. The direction specified for data points in the configuration software always refers to the network view of the communication. The definition of input and output objects in BACnet, however, refers to the process view, which is opposite to the network. Therefore, a BACnet analog input (AI) object is modeled as an analog output data point. The direction of client mappings naturally refers to the network communication. Therefore, a write client mapping is represented as an analog output data point.

In BACnet, commandable objects can be written with values at a certain priority. The value with the highest priority is in effect. To revoke a previously written value, a special NULL value is written. This takes back the value from the server object at the specified priority. When all written values are withdrawn, the *Relinquish_Default* value is in effect.

The default value property of a technology independent data point is mapped to the *Relinquish_Default* property for commandable objects. For BACnet objects which are not commandable, the default value is used to initialize the *Present_Value* of the server object directly. In this case, the default value is overwritten by successive writes and is therefore lost, while in commandable objects, the default value is saved and reactivated whenever there is no other known value for the object.

10.5.2 BACnet Schedulers

BACnet schedulers and the BACnet calendar adhere to the standard schedule and calendar object in BACnet. For each scheduler a BACnet schedule object is created. The calendar deserves a closer look. For each calendar pattern, a BACnet calendar object is created. The visible calendar on the web interface is therefore a collection of BACnet calendar objects and each calendar pattern is associated with the corresponding BACnet calendar object instance number. The calendar pattern *Holidays* is for example visible as *CAL,1* on the BACnet port.

The BACnet schedule object allows only objects of one selected data type to be scheduled. Therefore, schedulers on BACnet can only schedule one class of data points (only one group of analog data points). As a consequence, the value preset in BACnet always has only one element. The name of the value preset is not stored in BACnet technology and it is not accessible over the BACnet network either. A default name is created for each value used, such as '22 °C' for an analog value. An example of two scheduled BACnet objects is shown in Figure 10. When extended BACnet features are available, a preset label can be assigned to a specific scheduled value. For example the value '16 °C' can be assigned 'night'. Click in the column header and type the desired text.

Datapoint	Description	Location	Group	Default	22 °C	night
bac_temp1	temp	BACnet Port.Datapoints 1	1	0.00	22.00	16.00
bac_temp2	temp	BACnet Port.Datapoints 1	1	0.00	22.00	16.00

Figure 10: Example value presets in BACnet schedulers.

Priorities of exception days in a BACnet scheduler range from 1 (the highest) to 16 (the lowest). Weekdays in BACnet have no priority.

Changing the number of calendar patterns in a BACnet calendar can only be done through the configuration software and not during run-time. The individual calendar pattern entries in the calendar patterns can be changed at run-time. Therefore it is advisable to reserve a suitable number of calendar patterns in a BACnet calendar and leave them empty if not needed immediately.

10.5.3 BACnet Alarming

BACnet alarming on the device is based on the *intrinsic reporting* mechanism. Currently, algorithmic reporting is not supported. Alarm conditions can only be applied to data points which map to BACnet server objects. If defined, the intrinsic reporting properties of the

underlying BACnet objects are enabled. Alarm conditions can be specified for analog input, output, value objects (AI, AO, AV), for binary input, value objects (BI, BV), and for multi-state input, value objects (MSI, MSV).

Alarm servers in the BACnet technology are mapped to BACnet Notification Class (NC) objects. Each alarm server is mapped to one NC. The notification class number can be configured in the object instance number property of the alarm server object.

Remote alarms in the BACnet technology refer to a remote NC object. When the device starts up, the remote alarm object reads out the current alarm state of the remote NC and reporting objects. To get notified about alarm transitions during run-time, the device registers in the *Recipient_List* of the remote NC object.

Some BACnet devices do not send a usable text in their alarm notification messages. For those devices, the alarm client provides the option **Ignore alarm message text**. If this option is enabled, the alarm client ignores the message text of an alarm notification and reads the description property of the alarmed object instead.

10.5.4 BACnet Trend Logs

Trending in the BACnet technology is based on the BACnet TrendLog object. A number of restrictions apply to trend log objects in BACnet. Trend log objects must be created by the configuration software. These objects are accessible over the BACnet network for other BACnet devices and operator workstations (OWS). All configuration properties can be modified by the configuration software as well as an OWS. The number of trend log objects cannot be changed at run-time. Therefore, if it is intended that an OWS configures the trend logs, a suitable number of empty trend log objects (i.e., without attached data points) must be created in the configuration software.

In BACnet trend logs, only one data point can be trended per trend log object. The trended data point can either be a local BACnet server object or a remote BACnet object accessed through a client mapping, showing the referenced property for trending to the OWS. Data points of other technologies and the min/max/avg algorithms can be trended as generic data points without having a BACnet property reference.

BACnet trend logs support interval, COV, and trigger-mode logs, aligned intervals are available in interval mode. The settings for linear and ring-buffer logging are mapped to the *Stop_When_Full* property of the underlying BACnet trend log object. This setting in the configuration software is a default and can be overridden by writing to the *Stop_When_Full* property by the OWS. The trend log object adheres to BACnet revision 12.

If an enable data point is configured in the project, the *Log_Enable* property is written with the value of that data point. If no enable data point is configured, the *Log_Enable* property is set to TRUE as a default and can be modified over the network.

The fill-level action is mapped to generating a *buffer event notification* in the BACnet trend log object. The fill-level trigger can still be used for e-mails even if no notification class is configured in the BACnet trend log object. The fill-level percentage maps to the *Notification_Threshold* property. The percentage setting in the project is a default and can be changed by the OWS over the network.

The BACnet technology also supports *remote trend logs*. A remote trend log is basically a BACnet trend log client, which accesses trend data on another device. The remote trend can load the trend data from the remote device and supply it to trend controls, the OPC server, or make it accessible via the trend CSV files.

10.5.5 Dynamic Polling in BACnet

Read client maps in BACnet rely either on COV subscriptions or on polling. Polling can be configured as a fallback, if COV is not supported, by setting the poll cycle (see section

10.1.2). In addition, this technology also supports dynamic polling as described in section 11.5.3 by stopping any COV subscriptions and polls for inactive data points.

10.5.6 BACnet Data Points in Connections

BACnet data points can be used in local and global connections. Since in BACnet, commandable objects can be written with values at a certain priority, it may be necessary to revoke a written value over a data point connection from another technology data point. However, in other technologies, there is no notion of revoking a written value. To model this behavior, a distinctive *invalid* value can be written to such a data point. For those data points that do not have an intrinsic invalid value, a suitable value can be specified in the data point properties. To make a BACnet object convey an invalid value to the opposite side of a connection when itself activates the *Relinquish_Default*, enable the option **Relinquish to Invalid**.

10.6 Data Point Manager

Since the data points are not part of the object tree shown in the main window, there is a separate window to manage these objects. The same window is also used to select one or more data points, when a data point reference is to be added to the object tree. It is therefore possible to create and select a data point in one go.

The window to manage and select data points can be opened in two ways:

- Via a tool bar button, or the command *Edit->Datapoints...* from the main menu.
- Via the context menu of an object where a data point reference can be attached.

When opened via the main menu or tool bar, the window can be left open while working on the project in the main window. Data points can be edited, created, and connected to objects in the main windows tree view via drag & drop. In this mode, the *Select* button is deactivated and the only option is to use the *Close* button to close the manager window.

NOTE:	<i>While the data point manager window is open, some operations are not available. This includes device communication when in LNS plug in mode as well as the option to close the main program or to attach a data point to an object via the context menu.</i>
--------------	---

When opened via the context menu of an object to connect a data point reference, the main program is locked and waits until the user selected the desired data point or closed the manager window. The result of the selection is then used to create references to the selected points and connect them to the selected object.

The window itself is divided into different sections:

10.6.1 Folder List

At the left is a list of folders which is used to sort the available data points by their category. There are a number of predefined folders available:

- **Imported:** This folder has some sub-folders. One is used to hold data retrieved from a network scan, the other ones are used to display data imported from files. Data points in the import folder are not stored on the L-VIS device when the project is downloaded. They represent data points which are available on remote devices and are shown here as templates to create suitable data points for use on the L-VIS device (see section 10.16.4 for details).
- **Favorites:** This folder contains symbolic links to other data points.

- **System Registers:** This is the folder in which the available system data points are located. All these objects are automatically created and cannot be removed or modified in any way. They can only be selected to create a new data point reference for the object tree.
- **User Registers:** This is a folder for internal registers created by the user. To add a new register, select this folder and then use the *New* button or the context menu to add a new register object, represented by one or two data points.
- **Alarm:** This folder contains generic alarm servers. These provide technology-independent alarms. Any data point can be alarmed using a generic alarm server but the alarms are only accessible to the local L-Vis or to remote devices via OPC.
- **Trend:** This folder contains generic trend log objects. These can record historical values for any data points but are not visible to other devices on the network except for access via OPC.
- **CEA709 Port / BACnet Port:** This folder contains sub folders holding data points which are used to communicate with other devices on the network:
 - **Local NVs / Server Objects:** Data points which represent a local network variable or local server object are located in this folder. Select the folder and use the *New* button or the context menu to create a new local data point.
 - **External NVs / Client Mappings:** Data points which reference remote network variables or server objects on other devices, without connection to a local network object, are located in this folder. These are usually data points which were created from points in the import folder, by using the option *Use on Device* from the context menu of the imported point.
 - **Calendar:** This folder is used to hold a locally available calendar object with its calendar patterns (definitions of day classes like holiday, maintenance day, and so on). Current devices allow one local calendar object. To create it, select the folder and use the *New* button or the context menu.
 - **Scheduler:** This folder is used for local scheduler objects. Each of these objects will connect to a local scheduler on the device and will be configurable through this data point, that is, the data point transfers *scheduler configuration data* between the actual scheduler present on the device and the user interface.
 - **Alarm:** This folder is used for local alarm servers. On CEA-709 devices, only one alarm server may be created, which basically represents the node object of the device which is used to manage system alarms. On BACnet devices, an arbitrary number of alarm server objects may be created, where each alarm server object will be connected to a separate notification class object on the device.
 - **Trend:** Only available for BACnet models. This folder is used for local trend service data points which are accessible to other devices on the network via the BACnet trend service.
 - **Remote Devices:** This folder is used to collect all remote calendars, schedulers and alarm client objects which were created from network scan data. For each remote device, a subfolder will be created where the objects referencing this device are collected. Note that references to normal external data points go into the External NV or Client Mappings folder.

- **Modbus Port TCP:** This folder has one subfolder called *Datapoints*. It contains Modbus data points which are configured for the Modbus TCP port.
- **Modbus Port RS485:** This folder is available on BACnet devices only. It has one subfolder called *Datapoints*, which contains Modbus data points configured for the Modbus/RTU port.

Using the context menu on a folder, sub-folders may be created to organize the available data points.

10.6.2 Data Point List

At the top right, a list of all data points which are available in the selected folder is shown. From this list, objects may be selected (including multi-select) in order to modify some of their properties. Click on the **Include Subfolders** button to show all data points of the selected data point folder and all its sub-folders. This can be a convenient way for multi-select across folders.



The columns of the list will vary depending on the type of objects shown. Always visible are the data point name and the unique ID by which this data point may be referenced. Also shown is the number of times this object is currently referenced from the main object tree (the L-VIS project). Only data points which are currently unused (the Use column shows 0) may be removed from the list.

To filter for data point names, enter a search text into the **Datapoint Name Filter** text box and hit *Enter*. A drop-down list holds the previously used filters available. Filters can also specify name patterns for sub-data points using a dot. Typing the first dot expands all filtered data points to the first sub-level. Continue typing to filters on names of those child data points. For example, typing 'sw.val' shows all data points having 'sw' in their name, expands those data points, showing all child points which have 'val' in the name. For complex name filters, regular expressions may be used (see Section 15.8.3).

The list can be sorted by clicking on one of the column headers. For example, clicking on the **Direction** column header will sort the list by direction. Other columns display **Datapoint Name**, **NV name**, **SNVT**, **Object Name**, object **Type** and **Instance** number, allocation (**Alloc**) of server object (SO) and/or client mapping (CM), number of attached **Client Maps**, and the data point unique **ID**. To apply the current sort order as the new data point order on the device, right-click on the column header and select **Renumber Datapoints**. Alternatively, select from the menu **Tools → Renumber Datapoints**.

The **OPC** column provides check boxes for each data point. If checked, the respective data point is exposed to OPC on the device. Deselect the check box, if a data points shall not be exposed to OPC. Note, that deselected data points do not add to the OPC tag limit.

The **Par.** column provides check boxes for each data point. If checked, the respective data point is exposed to the parameter file.

New objects may be created in the selected folder by pressing the **New** button to the right of the list or via the **New** command in the context menu. A plus  sign in the list indicates that the data point contains child points. Clicking on the plus  sign expands the view.

For the alarming, scheduling, and trending features (AST), there are columns, which display icons for data points that are attached to an AST function. See Table 29 for details.









Icon	Data Point Usage
	Data point is scheduled
	Data point has an active alarm condition
	Data point has an inactive alarm condition.
	Data point has sub-data point with alarm condition.
	Data point is a trigger for e-mails.
	Data point used for trending.
	Data point is a link.
	Data point has sub-data points, which are links.

Table 29: Icons used for data points in the data point list view

The background of the data points in this list are color coded to make general information visible to the user. Some of the colors are only used in selection mode, when the data point window was opened to select a data point to connect to an L-Vis object. The complete color coding is described in Table 30.

Color	Coded Information
Yellow	The data point is currently unused and selectable for the object in question (in case selection mode is active).
Green	The data point is already referenced at least once and it is selectable for the object in question (in selection mode).
Orange	The data point is not selectable for the object in question (only happens in selection mode).
Blue	The data point is fixed / read-only. It may not be modified but is selectable for the current object (in selection mode).
Pink	The data point is fixed / read-only and it is not selectable for the current object (only in selection mode).

Table 30: Color coding used for data points in the data point list view

When an entry is selected, detailed property information is shown below the list, in the property view.


10.6.3 Property View

When one or multiple data points are selected, the available properties are displayed in the property view. Properties which are read-only are marked with a lock sign. When applying multi-select, only those properties common to all selected data points are displayed. According to the frequency of usage, different views for the properties are provided. **Basic Properties** list the most frequently used ones. **All Properties** is a list of all available properties for the data point. Depending on the network technology and data point class, different properties may exist.

Data point properties common to all technologies:

- **Datapoint Name:** This is the technology-independent data point name. This name may be longer than and different to the name of the native communication object (i.e., network variable). Data point names must be unique within a given folder. The maximum length is limited to 64 ASCII characters. International characters are allowed but will count as 2 or 3 ASCII characters each, due to UTF-8 encoding.
- **Datapoint Path:** This informational property specifies the entire path of the data point within the data point hierarchy.

- **Datapoint Description:** This is a human-readable description of the data point. The maximum length is limited to 64 ASCII characters. International characters are allowed but will count as 2 or 3 ASCII characters each, due to UTF-8 encoding. If available, the description will also be used as the object description for L-Vis data point objects referencing this data point.
- **OPC Tag:** If enabled, the data point can be accessed via OPC.
- **Parameter:** If enabled, the data point will be listed in the parameter file. Those data points are visible in the LWEB-900 parameter editor. A parameter data point is automatically persistent. See Section 10.1.5.
- **Use Polycle value as:** For input data points, this property defines whether the input shall use a receive timeout or be constantly polling. See Section 10.1.2.
- **Poll on Startup:** For input data points this property defines, whether the data point shall be polled once at start-up. Poll-on-startup can be enabled independently of the poll cycle. See Section 10.1.2.
- **Polycle:** For input data points, this property defines the poll cycle in seconds. Set this property to 0 to disable polling. See Section 10.1.2.
- **Receive Timeout:** For input data points, this property defines the receive timeout in seconds. Set this property to 0 to disable it. See Section 10.1.2.
- **Min Send:** For output data points, this property defines the min send time in seconds. See Section 10.1.2.
- **Max Send:** For output data points, this property defines the max send time in seconds. See Section 10.1.2.
- **Send-on-delta:** For output data points this property defines if value updates shall be sent only once they meet the COV condition of the data point. For analog data points the analog COV increment is used. If not checked, updates are always transmitted according to min and max send times. See Section 10.1.7.
- **Use Linear Scaling:** If this property is enabled, the analog values are pre-scaled from the technology to the data point. This scaling is in addition to any technology-specific scaling factors and any scaling done by the L-Vis data point objects referencing this data point. If enabled, the properties **Custom Scaling Factor** and **Custom Scaling Offset** accept the scaling factors. See Section 10.1.7.
- **Custom Scaling Factor, Custom Scaling Offset:** These properties only exist, if linear scaling is enabled. They apply to analog data points only. See Section 10.1.7.
- **Only notify on COV:** This property exists for binary and multi-state input data points. It defines, whether a data point should trigger an update only when the value changes, or on every write. If this is enabled, consecutive writes with the same value do not trigger an update. If you want to convey every write, disable COV on the data point.
- **Persistent:** This property defines, if the last written value shall be stored as a persistent value. Persistent data points restore that value after a restart from the persistent storage. See Section 10.1.4.
- **Default Value:** This property defines a default value (see Section 10.1.3). Enter a default value to enable this feature in the data point. Delete the value entirely to remove the default value. If no default value is defined, this property reads "N/A". The default is no default value.
- **Historic Filter:** This property allows defining historic filters for the scalar data point. See Section 10.3.5.
- **Point Type:** This is the base data point type, e.g., "Analog Datapoint".

- **Direction:** This is the data point direction. Use input, output or value as directions.
- **Unit:** For analog data points, this property contains the definition of an engineering unit of the scalar value. A human-readable text for the engineering unit is displayed and can be selected from a list of known units. If a correct unit was selected, it is displayed with a green checkmark ✓ (see Section 10.1.11).
- **Unit for display:** If the data point has a convertible unit, an alternative unit for display can be chosen. The data point value on the web interface is converted to and displayed in the display unit (e.g., °F instead of °C).
- **Analog Datapoint Max Value:** For analog data points this property contains the upper limit of the supported value range. Note that this does not define an alarm limit and has no impact on the data point function in general.
- **Analog Datapoint Min Value:** For analog data points this property contains the lower limit of the supported value range. Note that this does not define an alarm limit and has no impact on the data point function in general.
- **Analog Datapoint Precision:** For analog data points this property defines the number of decimals. '0' specifies an integer value. Display units may use this to format the floating point value accordingly.
- **Analog Datapoint Resolution:** For analog data points, this property defines the smallest possible value increment.
- **Analog Point COV Increment:** This property is valid for analog input data points. It specifies by which amount the value needs to change, before an update is generated. If every write shall generate an update even when the value does not change, specify 0 as the COV increment. If any value change shall generate an update, delete the value, which results in **Any**.
- **Active Text:** For binary data points, this property defines a human-readable text for the active state (true).
- **Inactive Text:** For binary data points, this property defines a human-readable text for the inactive state (false).
- **Current State Map:** For multi-state data points this property defines the multi-state map. It must be set to a valid multi-state map or it points to User/UndefinedStates. Click on  to assign a state map.
- **State Count:** For multi-state data points this property defines the number of discrete states.
- **State Text:** For multi-state data points this property defines a human-readable state label for each state.

10.6.4 CEA-709 Properties

Apart from the common data point properties discussed in Section 10.6.3, data points of the CEA-709 technology have additional properties. Depending on whether a NV is local or external (remote), the properties may vary.

- **NV Allocation:** This property defines how a data point shall be allocated on the device. Choices are *Static NV*, *Dynamic NV*, and *External NV*. If the allocation type cannot be changed, this property is locked.
- **SNVT:** This property defines the SNVT of the NV, e.g., *lux (79)*.
- **Invalid Value:** This property defines the invalid value for the NV. If set, this specific value will be interpreted as invalid in the data point. If known by the SNVT, the invalid value is filled in. Otherwise, the user can specify an invalid value.

- **CEA-709 Mapping Information:** This information is derived from the SNVT. It defines how the NV contents are mapped to the data point.
- **NV Scaling A, B, C:** These are the scaling factors known from the SNVT table. The scaling factors are applied to translate a raw NV value into the scalar representation of the data point.
- **Data Type:** This is the basic NV data type. This is usually filled in from the SNVT definition.
- **Local NV Member Index:** This property specifies the NV member index within a given functional block. This must be a unique index in the functional block, which identifies the NV after other NVs have been added or removed from the interface.
- **Local/Remote NV Index:** This property specifies the NV index. For local, static NVs this is the NV index of the static NV. For external NVs, this is the NV index of the NV on the remote device.
- **Local/Remote NV Name:** This property specifies the programmatic name of the NV. For local, static NVs this is the programmatic name of the static NV. For external NVs, this is the programmatic name of the NV on the remote device.
- **Local/Remote Functional Block:** This property specifies the functional block the NV should reside in. For local NVs, one of the available functional blocks can be selected.
- **Local/Remote NV Flags:** This property specifies the NV flags. For local (static or dynamic) NVs, the flags can be configured. For external NVs, these flags are only informational.
- **Remove NV Information:** For external NVs, this property contains the information on the remote device and the NV selector on that device.
- **Remote Device ID:** For external NVs, this property contains information on the remote device by listing the program ID and location string.
- **Remote Device Address:** For external NVs, this property contains the CEA-709 network addressing information to access the node, i.e., subnet, node, and NID.
- **Retry Count:** For external NVs, this property defines the retry count. The default is 3.
- **Repeat Timer:** For external NVs, this property defines the repeat timer in milliseconds. The default is 96 ms.
- **Transmit Timer:** For external NVs, this property defines the transmit timer in milliseconds. The default is 768 ms.
- **LNS Network Path:** If available from an LNS scan, this property specifies the LNS network path of the device where the given NV exists.
- **LNS Channel Name:** If available from an LNS scan, this property specifies the LNS channel name of the device where the given NV exists.

10.6.5 BACnet Properties

Apart from the common data point properties discussed in Section 10.6.3, data points of the BACnet technology have additional properties. Depending on whether a server object and/or a client mapping exists, the properties may vary.

- **Engineering Units:** For analog BACnet server objects, this property defines the engineering units from the BACnet standard. One of those units can be chosen from a drop-down box, if this property is not locked.
- **Server Object Type:** This property defines the BACnet object type of the underlying BACnet server object. It can be changed within the class, i.e., for an

analog data point, the server object type analog input, analog output, or analog value can be chosen.

- **Commandable:** This property defines if the underlying BACnet server object is commandable. For BACnet value objects (AV, BV, MV) this property can be edited to create commandable or non-commandable BACnet value objects.
- **Relinquish to invalid value:** This property defines whether the data point maintains the *Relinquish_Default* value when the priority array is empty, or assumes the invalid value. By default, this property is false and the *Relinquish_Default* value is used. Setting this property to true can be beneficial when used in a connection to withdraw a value in another technology.
- **Server Object Name:** This property defines the object name of the underlying BACnet server object. It must be unique among all server objects. It can be up to 64 characters.
- **Server Object Instance No:** This property defines the object instance number of the underlying BACnet server object.
- **Server Object Description:** This property defines the object description of the underlying BACnet server object. It can be left blank.
- **Server Object Device Type:** This property defines the object device type of the underlying BACnet server object. It can be left blank.
- **Get Active Priority:** Set this check box to let the data point reflect the active priority of the local or remote BACnet object. The priority is a number between 1 and 16. This property is only applicable for commandable BACnet objects.
- **Allocate Server Object:** This property defines whether a server object shall be allocated for the data point. This option is useful when a local server object shall be allocated for a client mapping.
- **Allocate Client Mapping:** This property defines whether a client mapping shall be allocated for the data point. This option is always set when at least one client mapping is attached.
- **Client Map Count:** This property defines the number of client mappings attached to a data point. A data point can have one read client map or *n* write client mappings.
- **Client Map [n]:** This is a list of client mappings. The property shows a summary of the client mapping parameters.
- **Client Confirmed COV:** This property defines whether a client map subscribes with the confirmed COV service. If disabled, unconfirmed COV is used.
- **Client Map Type:** This property determines the type of client mapping. It can be one of the following: Poll, COV, Auto, Write, or Value (see Section 10.5.1).
- **Client Write Priority:** For a write or value client map, this property defines which priority is used for writing.
- **Remote Instance Number:** This property specifies the object instance number of the remote server object. The object type cannot be edited.
- **Value Read Mode:** For value client mappings, this property defines the read mode: Poll, COV, or Auto.

10.6.6 Modbus Properties

Apart from the common data point properties discussed in Section 10.6.3, Modbus data points have the following additional properties:

- **Modbus Device Name:** This property defines the name of the Modbus slave device which contains the data point.

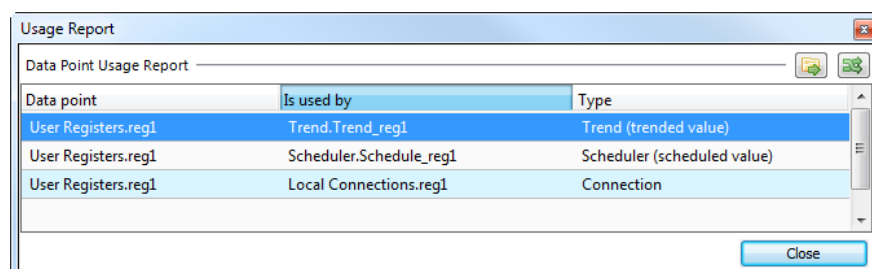
- **Modbus Device Address:** This property defines the address of the Modbus slave device which contains the Modbus data point.
- **Modbus IP Address:** This property is available for Modbus/TCP master only. It specifies the IP address of the Modbus slave device, which contains the Modbus data point.
- **Modbus Register Start Address:** This property defines the address of the Modbus register.
- **Modbus Register Type:** This property defines the register type of the data point. Also the function code, which specifies the register, is displayed. When the Modbus register type is changed from a read to a write register, also the direction of the data point is changed.
- **Modbus Data Type:** This property defines the representation of data in the slave. This is the data type the Modbus slave uses for the data point internally. This can for example be float, double, int16 or uint32.
- **Modbus Scaling (multiplier, exponent and offset):** These properties define the scaling parameters for the data point. The value of the data point in the device is calculated as follows:

$$\text{Value} = (\text{ModbusValue} + \text{Offset}) \cdot \text{Multiplier} \cdot 10^{\text{Exponent}}$$
- **Modbus Swap 16 bit, Swap 32 bit and Swap 64 bit:** This information specifies, if the order of received Modbus data has to be changed. When Swap 16 bit is set, the two bytes of a 16 bit word are swapped, if Swap 32 is set, the two words of 32 bit are swapped, and if Swap 64 bit is set, the two 32 bit words of 64 bit long data are swapped. Also combinations are possible. This configuration is necessary because the Modbus slaves can store information in any byte order (the Modbus protocol only specifies, how 16 bit data is transferred).
- **Pollgroup:** This property is only available for input data points. It shows the poll group, the data point is connected to.

10.6.7 Data Point Usage


Data points can be used by other objects, such as connections, trends, alarm servers, controls, actions, triggers, math objects, and many more. To get a usage overview of a certain data point, the data point list contains the column **Use**. This column provides a usage counter for each individual data point. If this counter is '0', the data point is not used by any other object.

If the usage count is greater than zero, choose the item **Show Usage...** from the data point context menu. This shows a report window listing all other objects referring to the selected data point. An example is shown in Figure 11.



Data point	Is used by	Type
User Registers.reg1	Trend.Trend_reg1	Trend (trended value)
User Registers.reg1	Scheduler.Schedule_reg1	Scheduler (scheduled value)
User Registers.reg1	Local Connections.reg1	Connection


Figure 11: Data point usage report.

Each line corresponds to an object using the data point. Select a line and click on the **Go to data point** button  to navigate to the selected object.

10.6.8 Multistate Maps

Multistate data points have a descriptive set of state texts for their state IDs. To manage those state IDs and state texts among many multistate data points, they refer to *multistate maps*. Some technologies have a fixed set of such multistate maps others have freely configurable multistate maps (e.g. user registers). Editing a multistate map affects all multistate data points which are using that particular map. It is not necessary to edit each data point individually.

To Edit a Multistate Map

1. Click on the **Configure** button  in the **Current State Map** property of a multistate data point. This opens the multistate map manager as shown in Figure 12.

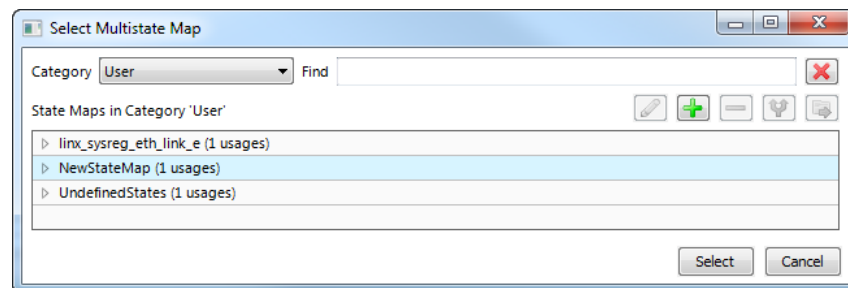


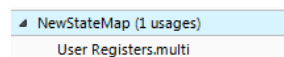

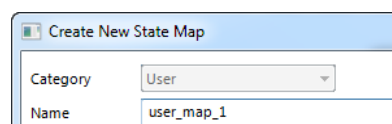


Figure 12: Select multistate maps in the multistate map manager.

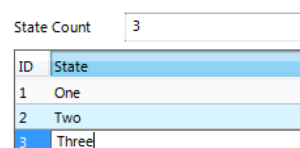
2. Select a **Category** and an existing state map in the **State Maps** list and click on **Select**. Maps that are fixed and cannot be changed are marked with a lock symbol .
3. Expand a state map line to see where this state map is used. Select the usage and click the **Go to Data Point** button . This navigates to the data point.



4. If a new multistate map shall be created, click the **Create State Map** button .
5. In the **Create New State Map** dialog enter a state map Name.




6. Then enter the desired number of states and edit the state texts as needed and click into the list of states. Edit state IDs and texts to your needs. Pressing Enter goes to the next line. Finally click the **Save** button.



7. Select the newly created multistate map and click the **Select** button. The selected map is now assigned to the data point.

10.6.9 Property Relations

Property relations can be linked to data points in various user dialogs (e.g. alarm condition dialog). For editing links of property relations for a large number of data points, the

Manage Relations tab in the property view area of the data point manager provides a fast solution. The data point tree is replicated in that tab as shown in Figure 13. Select a folder and optionally click on the button  to include property relations of data points in sub-folders. Enter a filter expression to **Filter**, which applies to both the data point name and relation type. For example, enter 'feedback' to display all feedback value property relations.

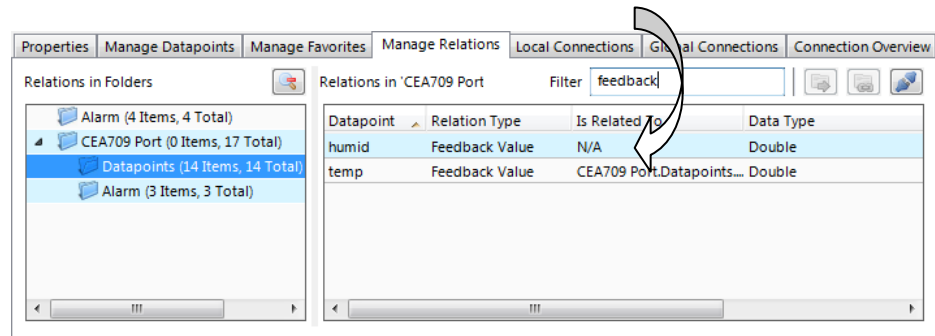




Figure 13: Manage relations tab.

To link property relations to other data points using the manage relations tab, navigate to the desired folder in the main folder tree of the data point manager and select the data point to be linked. Drag the data point onto the **Is related To** column. For detaching links, use multi-select on the desired property relations and click the detach button . On linked property relations you may click on the button , which jumps to the linked data point in the data point list.

10.6.10 Favorites


There is a special **Favorites** top-level folder in the device data point folder hierarchy. This folder contains freely configurable symbolic links to data points, which may actually reside anywhere in the folder structure. This folder represents a way to assemble an alternate logical view to the data point hierarchy.

To configure favorites, select any data point from any location in the data point folder hierarchy and drag it onto the favorites folder. This will create a data point link, which is displayed in the data point list:

Favorites - Favorites				
Link Name	OPC	PLC	Link Path	ID
my_name_humid_link	<input checked="" type="checkbox"/>	<input type="checkbox"/>	LINX-120.CEA709 Port.Datapoints.abs_humid	10FE

The **link name** can be edited to something different than the original data point name. The contents of this folder are also available on the Web UI and the link names are displayed there. The data point links can also be individually exposed to the OPC server independent of the OPC expose status of their original source.

Furthermore, the user can create sub folders in the favorites folder and beneath, to build a hierarchy of data point links. Folder links are, however, not allowed.

For editing links of favorites for a large number of data points, the **Manage Favorites** tab in the property view area of the data point manager provides a fast solution. The data point tree is replicated in that tab as shown in Figure 14. Select a folder and optionally click on the button  to include favorites in sub folders. Enter a filter expression to **Filter**, which applies to the favorite name. For example, enter 'room1' to display all favorites that contain this as a sub-string.

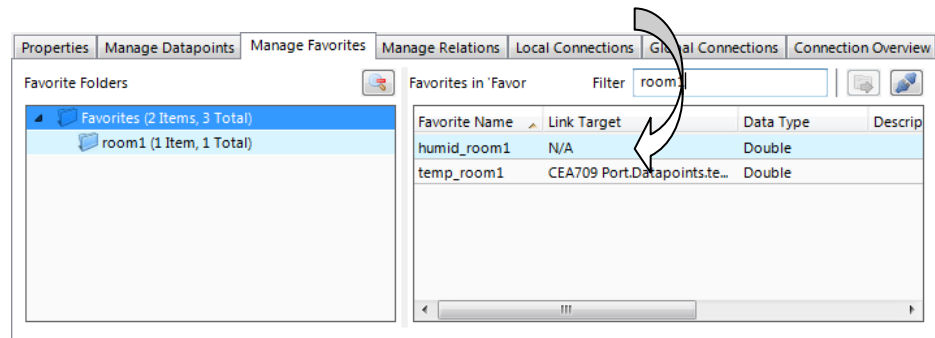




Figure 14: Manage favorites tab.

To link favorites to other data points using the manage favorites tab, navigate to the desired folder in the main folder tree of the data point manager and select the data point to be linked. Drag the data point onto the **Link Target** column. For detaching links, use multi-select on the desired favorites and click the detach button . On linked favorites you may click on the button , which jumps to the linked data point in the data point list.

10.6.11 Buttons

At the right side of the window are a few buttons to create new data points as well as duplicate and remove objects, which provide the same commands as are available through the context menu in the object list.

There are some buttons which provide additional functionality:

- **Cleanup:** This button scans all data point folders and removes any objects which are currently not referenced by any other objects in the project.
- **Devices:** This button opens the remote devices dialog, listing all remote devices which are currently involved in data exchange with the LVIS device, excluding OPC server definitions (see next button).
- **OPC Server:** This button opens the OPC device manager. This dialog is used to create new OPC client data points from imported LINX device configurations as well as update existing data points when the LINX configuration has changed. See Section 10.7 for details about this dialog.
- **Modbus:** This button opens the Modbus management dialog.
- **Adapter:** Opens the dialog to manage connection adapters, as used in data point connections with math formulas.
- **Scripts:** Opens the script manager dialog.
- **Find:** This button tries to find the first reference of the selected point in the object tree of the main window and, if available, selects the corresponding object. From this first occurrence, you may use the *Find Next* function (or F3) of the main window to find similar references throughout the project or open the find dialog and use the *Like Selected* to get a list of all similar data point references.

10.7 OPC Device Manager

L-Vis devices support the OPC XML-DA standard for communication with compatible OPC server implementations, for example LINX devices or an LWEB-900 server running on a PC. To add new OPC servers to the project or update the data points associated with an existing OPC server, the OPC device manager window is used. It can be opened by a click

on the button **OPC Server**, which is available on the right hand side of the data point management window (see also Section 10.6.11).

10.7.1 Tool Bar

On the top right, a tool bar is available to execute various operations.



The buttons from left to right are:

- **Add:** Use this button to add a new OPC server from a configuration file of the device. Supported configuration types include all L-INX, L-GATE, and L-DALI configuration files. Once the device configuration was loaded, suitable OPC client data points will be generated and added to the L-Vis project.
- **Refresh Status:** This button refreshes the status of the selected OPC device. It will check if the current data point configuration of the device is still in sync with the data points present in the L-Vis project, or if an update is needed.
- **Reload:** This button reloads the device configuration and updates the OPC data points in the L-Vis project to match the current configuration of the OPC server. Use this button to update the OPC data points whenever changes were made to the data point configuration of the OPC server.
- **Select Source:** This button allows you to change the device configuration file associated with the selected OPC device. In case the name or location of the configuration file was changed, the OPC device definition must be updated with the new location of the file.
- **Open in Editor:** This button will launch the configuration software suitable for the selected OPC device and load the associated data point configuration.
- **Delete:** This button deletes the selected OPC devices and all associated data points. Note that once the data points have been deleted, there is no easy way to restore them in a way that references from L-Vis objects to these data points will work again. Even after adding the same device again, the new data points will have different UUIDs and all references to the old data points will be broken.

10.7.2 Device Properties

Below the tool bar is a list of all OPC devices. Each of the device entries can be expanded to show a number of properties. These properties can be edited and are explained below:

- **Name:** Internal name for this OPC device. Note that changing the name will also change the data point folder name in which the associated OPC data points are located. In case there are templates or template instances referencing the old data point folder name, they will need adjustment to fit the new name.
- **Local Address / Port:** These properties specify the primary IP address and port number which should be used to contact the OPC server.
- **Local use HTTPS:** When this option is checked, the OPC client will use HTTPS instead of HTTP when contacting the server.
- **Public Address / Port:** These properties define a secondary IP address and port number, which can be tried by the client in case the server is not reachable via the primary address. This can be used in NATed environments, where different addresses need to be used depending on the location of the client.

- **Public use HTTPS:** Same as the option for the local address, but related to the public address.
- **Operator Password:** The operator password to use when contacting the server.
- **Write Aggregation:** Time in milliseconds to wait for more values to write, before a write request is sent to the server. This increases efficiency by reducing the overhead involved in building and transmitting a SOAP request, compared to the number of values written with that request.
- **Lower Limit to Min. Send:** Minimum time in milliseconds for the Min. Send property of the OPC data point. Any value lower than this will be replaced by the specified minimum time.
- **Config Status:** Shows the current status of the server configuration. The configuration is either shown as *up to date*, meaning that the L-Vis project is still in sync with the server configuration, or the status indicates that the server configuration was changed or removed since it was last imported.
- **LWEB Time:** The time when the server configuration was last imported.
- **Source Time:** The time when the server configuration was last modified.
- **Source Path:** The location of the imported server configuration on disk.


10.8 Manage Connections

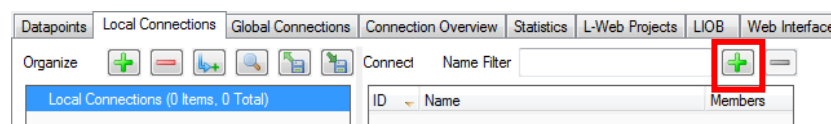
10.8.1 Create a New Connection

After configuring the device's data points, internal connections between those data points can be made using local connection objects.

A connection is an internal mapping in the device between input and output data points. A connection always consists of one or more data points. A value update from an input data point is distributed to all output data points. A status change of an output data point is propagated back to all input data points. All data points in the connection must be of compatible type.


To create a new connection

1. Click on the **Local Connections** tab in the data point configuration window and press the button  **Create new Connection**.



In the menu choose **Standard Connection**. A new connection is added to the connection list. Rename the connection if you want to do so.

ID	Name	Members
1000	New Connection	0

2. Above the list **Datapoints in connection** on the right-hand side click on  **Attach Data Point** to add data points for this connection. This opens a list of all available data points. Select one and press **OK**. You may use multi-select to select more than one data point at a time.

NOTE:

By default, only compatible data points are displayed. Sometimes compatible data points are available as member points (e.g., a SNVT structure member). Expand the data point and select the desired member point if required.

- Now the connection tab contains the new connection and below the list of data points in that connection as shown in Figure 15.

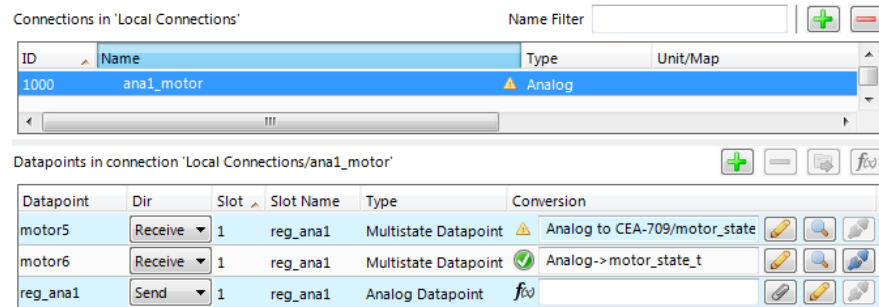


Figure 15: Connection tab with a connection and data points.

- Change the direction by modifying **Send** or **Receive**. For changing multiple data points use multi-select. Optionally, select **Disable** to temporarily exclude this data point from communication in the connection.
- If the attached data point needs conversion, the item displays a yellow exclamation mark ⚠ and the default conversion (e.g. 'Analog to CEA-709/motor_state_t'). Click on the button to view the current conversion.
- To add a new conversion to this item, click on the button. A dialog opens, which displays the matching adapters already available in the library as shown in Figure 16.

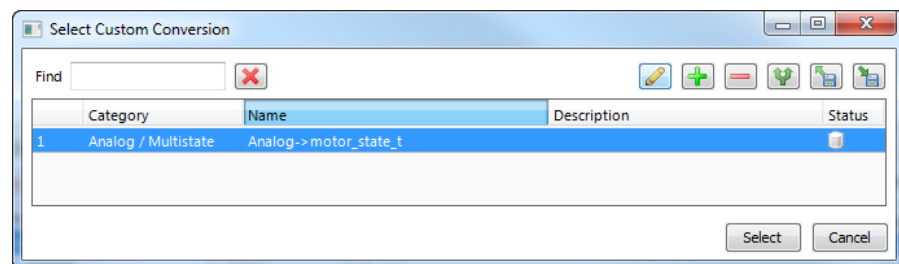


Figure 16: Choose a custom conversion.

- Select an existing conversion, click the plus button to create a new conversion, or click the edit button to modify an existing conversion.
- An example for editing an analog to multi-state value conversion is shown in Figure 17. Enter a **Conversion name**, then edit the **Value range from** column and select the desired **Target state** mapping.

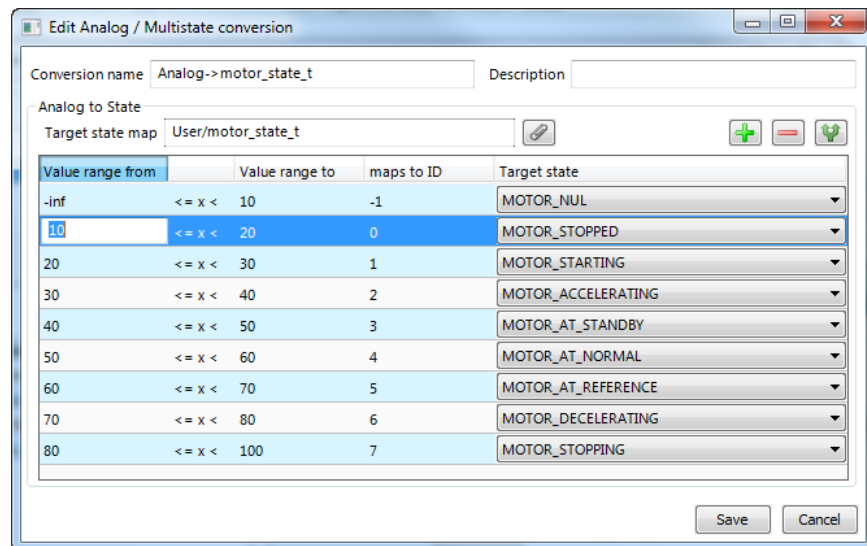


Figure 17: Edit an analog to multi-state conversion.

9. Click **Save** and then choose the newly created conversion by clicking **Select**.
10. The items with an assigned value conversion appears with a green check mark ✓.

To Create a Connection via Drag-and-Drop

1. Using the data point folder tree on the left and the list of data points on the top right, navigate to the data point that shall be added to the new connection.
2. In the properties view below the data point list click on the **Local Connections** tab as shown in Figure 18.
3. Then simply drag a data point from the data point list and drop it onto an empty area in the connections list as shown in Figure 18.

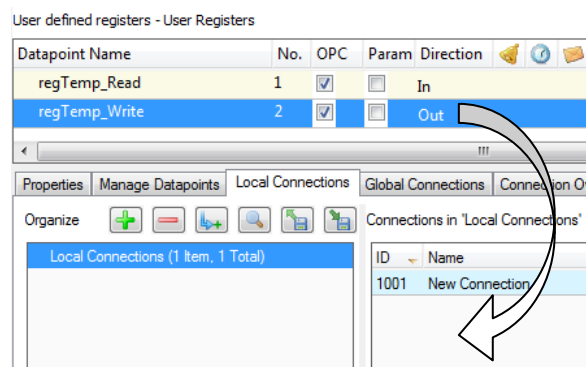



Figure 18: Create a connection with drag-and-drop.

4. To add data point to that connection, drop the new data point into the empty area in the connected data points list below.

10.8.2 Modify Connections

Connections can be edited and deleted. Editing connections does not influence the data point configuration. This means, when deleting a connection or removing data points from a connection, the data points are not deleted.

To Edit a Connection

1. Change to the **Local Connections** tab.
2. Select the connection to edit and follow the same steps as when creating a connection.
3. To detach a data point from the selected connection, select the data point and click on the button  **Detach selected data points** over the connection member list.

Datapoints in connection 'folder1/nviTemp'

Datapoint	Dir	Slot	Slot Name	Type	Conversion
nviTemp	Send	1	nviTemp	Analog Datapoint	f ₀

4. Change the direction of a data point in the connection by choosing one from the **Dir** drop-down. You may select **Disable** to temporarily exclude this data point from the connection altogether.

To Add Data Points via Drag-and-Drop

1. Navigate to the data point to be added.
2. In the properties view below the data point list click on the **Local Connections** tab and select an existing connection as shown in Figure 19.
3. Drag the selected data point and drop it into the empty area of the **Datapoints in connection** list as shown in Figure 19. This adds the data point to the selected connection.

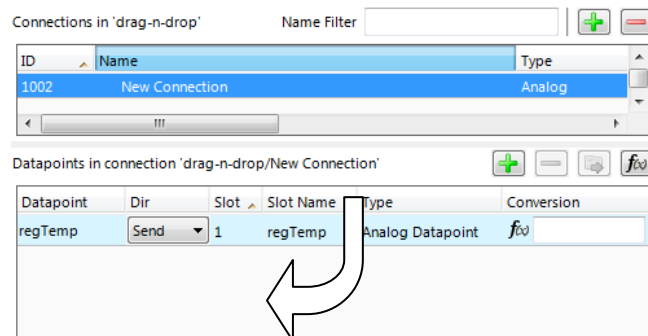





Figure 19: Modify connections in the properties view.

4. To replace a data point in a connection drop the new data point right onto an existing data point in the connection that shall be replaced.

To Delete a Connection

1. Change to the **Local Connections** tab.
2. Select the connection for removal. Use multi-select to select more than one connection.
3. Click on the button  **Delete Connection** above the connections list.

Connections in 'Local Connections' Name Filter  



ID	Name	Members
1001	New Connection	3

10.8.3 Create a Multi-Slot Connection





A multi-slot connection can be used to connect a number of different data points together under one umbrella. It can be considered as a cable with many wires, each wire represented by a slot with a label. For example, one can create a multi-slot connection for a structured data point, connecting each child data point to another technology. All those slots appear under the same connection, but data points only exchange data with other points in the same connection slot.

To Create a Multi-Slot Connection

1. Navigate to the (structured) data point to be connected.
2. Select the data point and drag it into the connections list to create a new connection.
3. A dialog prompts the user whether to create a multi-slot connection or use the user data point as a single entity in the connection. Choose **Yes** to create a multi-slot connection.
4. A multi-slot connection is created as shown in Figure 20. The multi-slot connection can be collapsed or expanded. In the expanded view it shows all slots. Select the top-level multi-slot connection to view all data points in the connection. Select a single slot beneath to view only data points in that slot.

Connections in 'multislot' Name Filter  

ID	Name	Type	Unit/Map
1003	setting1	-	-
	setting1.function	Multistate	CEA-709/setting_t
	setting1.setting	Analog	%
	setting1.rotation	Analog	°

Datapoints in connection 'multislot/setting1'    



Datapoint	Dir	Slot	Slot Name	Type	Conversion
setting1.function	Receive	1	setting1.function	Multistate Datapoint	
setting1.setting	Receive	2	setting1.setting	Analog Datapoint	f_{ω} 
setting1.rotation	Receive	3	setting1.rotation	Analog Datapoint	f_{ω} 

Figure 20: Multi-slot connection.

5. Add more slots by dropping data points onto the top-level multi-slot connection.
6. Connect other data points to existing slots by dropping them onto the respective slot. For example, connect the register 'regRotation' by dropping it onto 'setting1.rotation' as depicted in Figure 21.

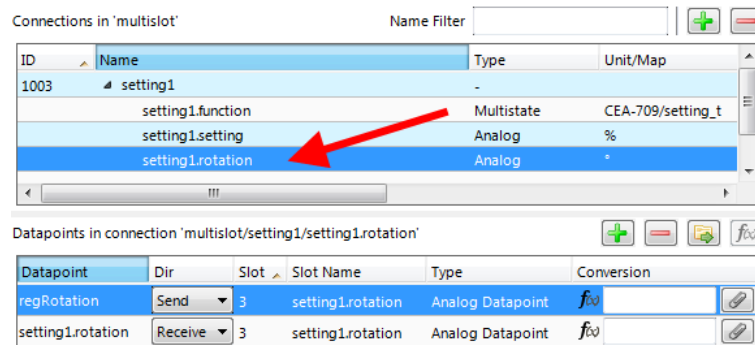


Figure 21: Add data point to connection slot.

10.8.4 Create a Math Block Adapter

When connecting structured data points, the multi-slot connection can be used. If a simple mapping of the child data points is not possible and a more advanced mathematical conversion is required, a math block adapter can be created. This math block is based on a multi-slot connection with n inputs and m outputs (see Section 10.2.2).

To Create a Math Block Adapter

1. Create a multi-slot connection from a structured data point, e.g., the input data point.
2. Add output slots to the multi-slot connection by adding a structured output data point. An example is shown in Figure 22.

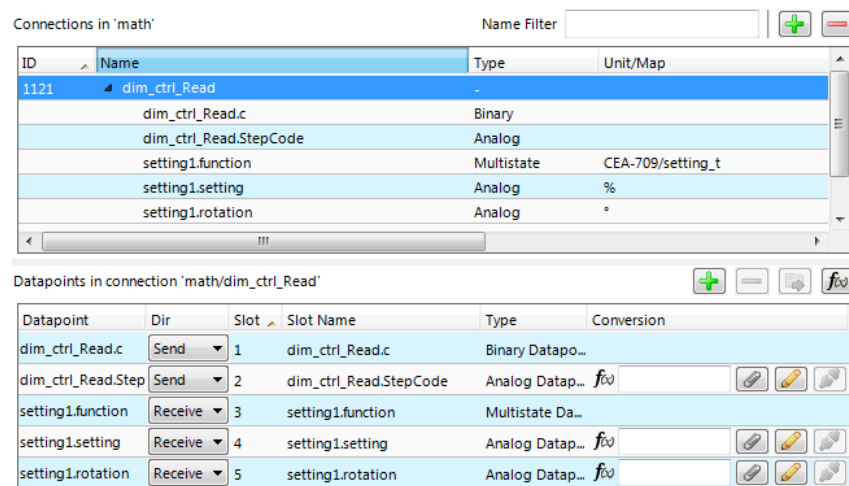

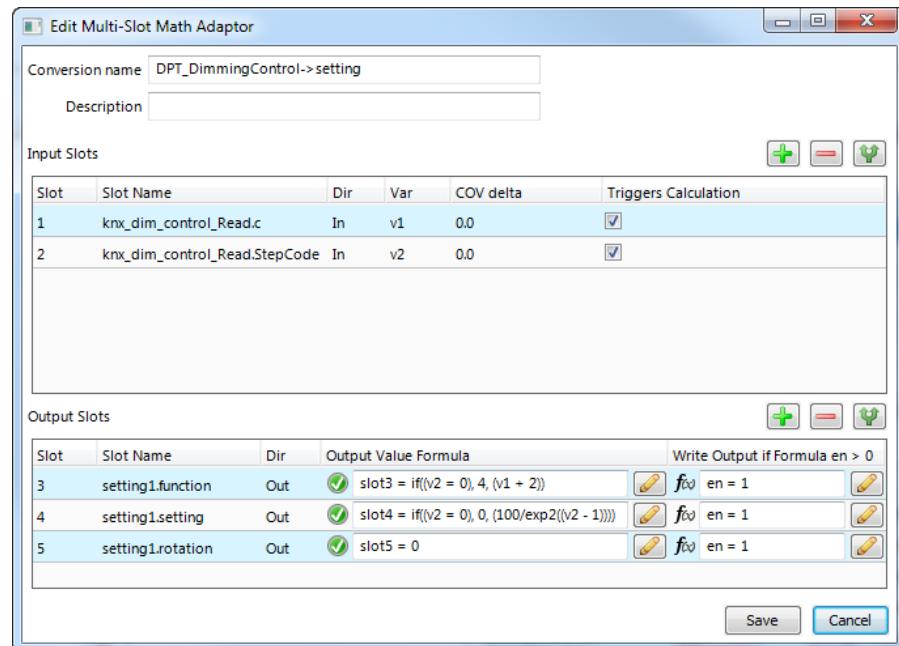


Figure 22: Multi-slot connection for math block adapter.

- Click on the **Create math adapter from connection**  button. The dialog **Edit Multi-Slot Math Adapter** opens as shown in Figure 23.



Conversion name: DPT_DimmingControl->setting

Description:

Input Slots


Slot	Slot Name	Dir	Var	COV delta	Triggers Calculation
1	krx_dim_control_Read.c	In	v1	0.0	<input checked="" type="checkbox"/>
2	krx_dim_control_Read.StepCode	In	v2	0.0	<input checked="" type="checkbox"/>

Output Slots


Slot	Slot Name	Dir	Output Value Formula	Write Output if Formula en > 0
3	setting1.function	Out	slot3 = if((v2 = 0), 4, (v1 + 2))	<input checked="" type="checkbox"/> f_{∞} en = 1
4	setting1.setting	Out	slot4 = if((v2 = 0), 0, (100/exp2((v2 - 1))))	<input checked="" type="checkbox"/> f_{∞} en = 1
5	setting1.rotation	Out	slot5 = 0	<input checked="" type="checkbox"/> f_{∞} en = 1

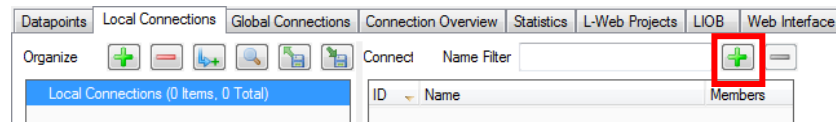
Save Cancel

Figure 23: Edit a math block adapter.

- Enter a **name** and **description** for the adapter.
- For each output enter an **Output Value Formula**. This can be done by directly typing the formula or by clicking the edit button .
- Optionally, enter an enable formula into **Write Output if Formula en > 0**. By default, enable is '1'.
- When finished with the math block, click **Save**.

To Use an Existing Math Block Adapter

- Click on the **Local Connections** tab in the main window and press the speed button  **Create new Connection**.



In the type menu choose **Connection with Math Adapter**.

2. In the dialog **Select Multi-Slot Math Adapter** select an existing adapter and click **Select**. A new multi-slot connection is added to the connection list with empty slots as depicted in Figure 24.

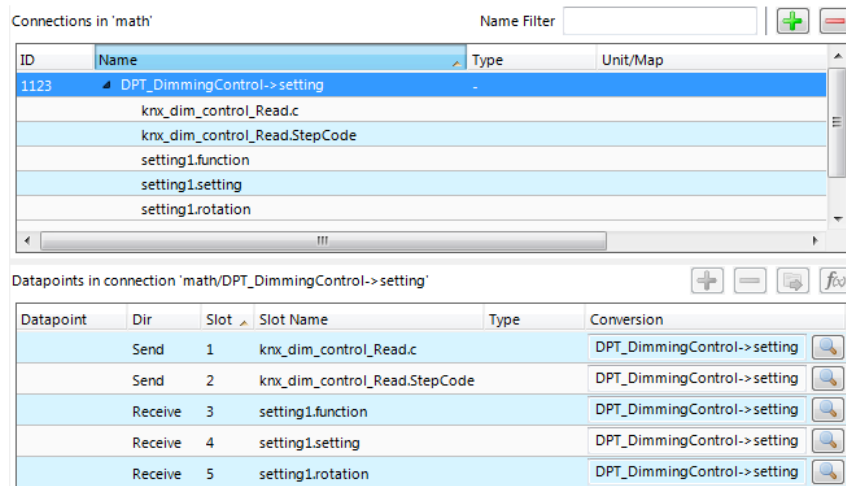



Figure 24: Created new multi-slot connection from math adapter.


3. Now connect data points by dragging and dropping them onto the empty slots in the data point list below.
4. To view the math conversion click the magnifier button .

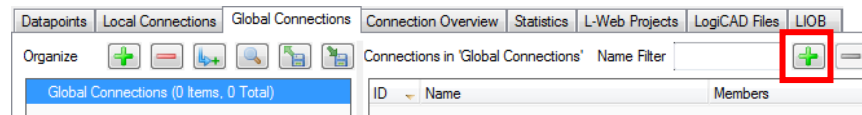
10.8.5 Create a Global Connection

Global connections are an easy way to publish or subscribe to global data, which is distributed among devices. To configure such communication, a device needs to be member of a CEA-852 channel. Once it is member of that channel, global connections need to be configured in the data point configuration. This is similar to creating local connections and most of the configuration steps apply also for global connections. In addition, also network timing parameters can be configured for global connections. For a description see Section 10.2.3.

If other devices already have global connections configured that publish data on the network, their definition can be exported and imported into the new configuration. That will make all the global connection names available. When creating manually, the connection names can be edited.

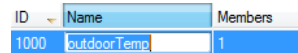
To Create a Global Connection

1. Click on the **Global Connections** tab in the main window and press the button  **Create new Connection**.



A new connection is added to the connection list.

2. Define a name for the global connection. This name is required to be unique on the network. Data will be published or subscribed to under this name.



3. Add data points to the global connection as described in Section 10.8.2. As a default, output data points will be added as sending, input data point as receiving data under the global connection.
4. Change the direction by modifying **send** or **receive**. For changing multiple data points use multi-select.



5. Define timing parameters for a global connection that is sending out data. On the global connections tab, the connection properties are listed below the data point member list. In the tab of the property area click the button

Properties of connection

Name	Value	Description
Connection Name	outdoorTemp	Name of the c
Max. Send [s]	0	Send to netw
Min. Send [s]	0	Limit network

6. To export the definitions of the created global connections, click the button **Export connections to disk** and choose the XML format.
7. To use those definitions, click the button **Import connections from disk** and choose an exported connections XML file. This creates the global connections structure with connection names but without any data points. Data points can then be added, for example, via drag-and-drop.

10.9 E-mail Templates

To make use of the E-Mail functions, mail templates can be created in the folder called *Mail Templates*. These are actually not data points, but are related to data points and are therefore managed in the same window. For each different type of mail, a template must be created and set up as required. The template may then either be used to send mail when one or more trigger data points reach a certain value or the template may be triggered by an action object (see the description of the action object in section 9.8).

In addition to the mail template setup, the mail account setup as described in section 11.6 must be completed to enable the actual sending of email.

To create a new mail template, select the folder called *Mail Templates*, open the context menu in the data point list and select the item *New E-Mail Template...*. A new dialog to set the properties of the new template is opened. This dialog has three tabs which are used to set up the new mail template.

10.9.1 Common E-Mail Properties

This tab is used to set the basic properties of the mail. First, enter a name for the new email template and the desired list of recipients. Separate multiple recipient addresses by comma or semicolon. The following types of recipient lists are provided:

- To: The mail is directly addressed to these persons.
- Cc: A carbon copy goes to the addresses listed here.
- Bcc: A blind carbon copy is sent to the recipients in this list. Recipients on this list will not be shown in the mails sent to other recipients.

Other properties to fill in are the desired subject of the mail and a rate limit, given in the maximum number of times this mail may be sent within a 24 hour period and the maximum number of times the mail may be sent in short succession.

In the top right area of the tab, you may attach data points to this mail. Every time this mail is sent, the current value of all attached data points is fetched and may be added to the mail text. Use the following procedure to embed the current value of a data point in an email:

- Press the *Add...* button to select the desired data point.
- The selected point will be added to the list of data sources.
- Select the data point from the source list and make sure the dropdown box below the list is set to *Selected Data Source Value*.
- Set the cursor to the position in the mail text at which the value of the data point should be inserted and press the *Paste to Text* button to add the required placeholder string to the mail text.

Alternatively, the placeholder string may be entered manually while typing the email text. The placeholder format is: `%{variable}` where variable is v1, v2, v3 and so on. The variable name assigned to a specific data source is shown in the source list.

NOTE:	<i>There are two special variable names which may be used in a placeholder string. The special variable name mailid will be replaced by the unique mail ID, the variable timestamp will be replaced by the date and time the email was sent. Both placeholder strings can be inserted automatically using the dropdown box and the Paste to Text button.</i>
--------------	--

10.9.2 E-Mail Trigger

This tab is used to configure automatic mail sending based on the value of certain data points. If nothing is configured here, the mail may still be sent using an action object in the L-VIS project (please refer to section 9.8 for a description of action objects).

To send the mail based on the value or state of a data point, follow these steps:

- Add the desired data point using the *Add...* button. Valid data points to use for mail triggers are normal analog, binary, and multistate values as well as alarm server and client points.
- Depending on the type of data point, select the conditions which should trigger this mail. Normal data points can trigger on the current value (zero or non-zero) as well as the state (invalid or offline), alarm data points can trigger the mail based on various alarm types and/or state transitions.

- Repeat for additional data points, if the mail should be triggered by more than one source. In case of multiple mail triggers, the mail is sent if any of the conditions is met (as opposed to all conditions).
- If required, select a single data point to be used as a master enable/disable switch below the trigger list. If a data point is selected for enable/disable, its value must be non-zero to allow mail sending. If no data point is selected, the mail template is considered enabled.

10.9.3 Attachments

This tab is used to select files which should be attached to the mail. This is used for example to send the current system log file or data log files together with the mail. The available files depend on the project and can be selected from the dropdown list. Use the *Add* and *Remove* buttons to manage the list of attached files.

NOTE:	<i>Data from trend log or data log controls can only be attached to a mail if the control has the option Preserve Data enabled. Otherwise, the data is not saved in a file and is therefore not accessible to the E-Mail system to include as a file attachment.</i>
--------------	---

NOTE:	<i>Attaching a number of large files will require a significant amount of free memory while the mail is being composed and sent. If the project itself consumes most of the available RAM, this should be taken into account. If necessary, split the information into multiple smaller mails and send them at different times to reduce the memory requirements.</i>
--------------	---

10.9.4 Limit E-mail Send Rate

The transmission of e-mails is triggered by the configured trigger conditions. It is not predictable, how often the trigger condition will cause the transmission of an e-mail. The e-mail template can be configured to limit the number of transmitted e-mails. This is done in the Configure E-mail Template dialog.

To configure an E-mail Rate Limit, configure the settings:

- **Max. E-mails per day:** This setting defines how many e-mails can be sent on average per day. The actual number of transmitted e-mails on a specific day may be slightly higher than this setting, depending on burst rates. The default is 100 e-mails per day. This results in an average interval of one e-mail per 14 minutes.
- **Send burst count:** This setting defines how many e-mails may be transmitted shortly after each other not limited by the above average interval. After the burst count, the average mails per day limit takes effect. The default is a maximum of 20 e-mails in a row.

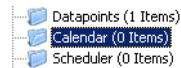
10.10 Local Schedule and Calendar

10.10.1 Create a local Calendar

As the first step, the required data points must be created. A calendar must be created, if the schedules shall work with exception days, such as 'Holidays'. If it suffices for schedules to define daily schedules for normal weekdays only, no calendar needs to be created.

To Create a Local Calendar

1. Under the port folder, select the **Calendar** sub-folder to create a calendar.



2. Right-click in the data point list view and select **New local Calendar**
3. In the Create New Calendar dialog box (as shown in Figure 25) enter Name and Description of the calendar. Note, that a BACnet calendar does not have the **Effective Period** settings.

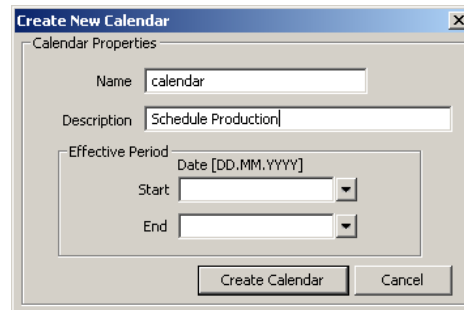


Figure 25: Create New Calendar dialog box.

4. Click **Create Calendar**. The calendar appears now in the data point list view.

10.10.2 Create Calendar Pattern

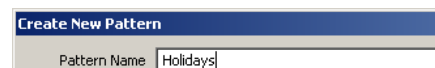
When a local calendar is used, it needs to be configured with calendar patterns. A calendar pattern represents a class of days such as 'Holidays'. The calendar patterns can then be used in a schedule to define daily schedules for exception days. The available calendar patterns should be created when the system configuration is engineered. The actual dates in the calendar pattern can be modified later at run-time.

To Create a Calendar Pattern

1. Select an existing calendar data point.

No.	Direction	Calendar Name
1	In	calendar

2. Right-click and select **Create Calendar Pattern...**
3. Enter a Pattern Name in the **Create Calendar Pattern** dialog



4. Click **Create Pattern**. The dialog closes and the calendar pattern appears beneath the calendar data point.

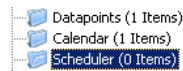
Local calendar objects							
	No.	Direction	Calendar Name	Index	Func. Block	Use	ID
	1	In	calendar			0	1030
	1.1		Holidays				1032

10.10.3 Create a Local Scheduler

For scheduling data points, a scheduler object must be created. Under each port folder, multiple local scheduler objects can be created. These local schedulers can then be configured to schedule data points.

To Create a Local Scheduler

1. Under the port folder, select the Scheduler sub-folder to create a scheduler.



2. Right-click in the data point list view and select **New Local Scheduler**
3. Enter a name for the schedule and a description. Note that the schedule automatically detects a corresponding calendar, if it has been created already.

4. Click **Create Schedule**. The new schedule appears in the data point list of the Scheduler sub-folder.

10.10.4 Configure Scheduled Data Points

Once the scheduler has been created, the data points it will be controlling must be defined. This is done by attaching data points to the scheduler. Note that there may be limits, on how many and which data points may be attached (see Section 10.5.2).

This configuration must be done as an initial setup. The scheduled data points and daily schedules can be changed later in the web UI or over the network.

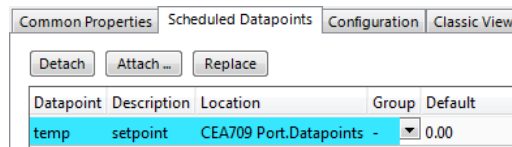
To Attach Data Points to a Scheduler

1. Select the scheduler data point in the Scheduler sub-folder.
2. Right-click and select **Configure Schedule** from the context menu. The same dialog which appears when a new scheduler is created is shown and allows configuring the scheduler. Of course, this step can also be done directly when the data point is created.
3. Select the tab **Scheduled Datapoints**.

4. Click the button **Attach ...**. This opens another data point selector window.
5. Select the data points to attach and click **OK**. For each of the attached data points, one or more lines appear in the list below the attach button. If the attached point is a structure, there will be one line for each element of the structure.

NOTE: *Data points can also be attached to a scheduler by choosing a data point from the data point list, drag the point onto a scheduler data point folder to open the list of scheduler data points, and drop it on the desired scheduler data point.*

6. Enter a short text in the **Description** field in the second column of each line. This text will serve as a label, which will be shown on the device's UI to identify the data point.



7. Add new value presets by entering a name and pressing the **Create** button next to the input field.

New Preset Name

NOTE: To generate presets automatically for multi-state data points, click the **Auto-Create** button. This button is available, if no other presets have been defined yet.

8. For each new preset, a new column will appear in the list. In this column, enter the desired value for each of the attached points, which will be set when this value template is scheduled. The user may later edit the values for each preset on the device but cannot add new value presets unless there is only one line (one value) in the list.

Datapoint	Description	Group	Default	day	night
NW_bac_lonCtrlInvo12_temp	temp	1	0.00	21.00	16.00

9. If there are multiple output values which belong together, they can be grouped in order to save space on the device. For each group, the entered value is stored only once, even if there are more data points in the same group.

Datapoint	Description	Group	Default	day	night
NW_bac_lonCtrlInvo12_temp	temp	1	0.00	21.00	16.00
NW_bac_lonCtrlInvo13_temp	temp	1	0.00	21.00	16.00

10. When done with the point and value setup, switch back to the **Configuration** tab or click **Save Changes** to leave the dialog.

NOTE: A shortcut for creating a scheduler object and attaching a data point is to select a data point in the data point manager, right-click on it and choose **Schedule Datapoint** from the context menu. This generates a scheduler and links that data point to it.

10.10.5 Configure Scheduled Events

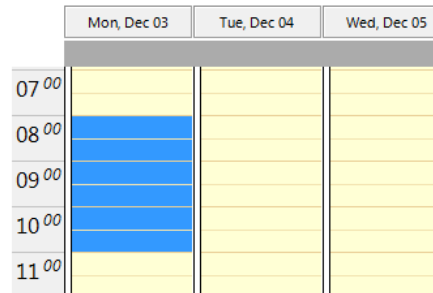
Once a scheduler is configured with attached data points and value presets, the schedule events can be defined. This can be done on the device or over the network at run-time, or also in the configuration software. A schedule consists of a number of scheduled events, each event starts at a defined time, has a scheduled value and ends at a defined within the same 24-hour period (starting at 00:00 and ending at 23:59 hours). The event can be configured to occur at a given date, for each weekday, or for a number of recurring dates.


In addition, scheduled events can be configured to occur on exception days from a calendar, such as "Holidays". An exception day always overrides a normal weekday. If more than one exception day is used, a priority must be assigned. This is necessary so that the system knows which schedule to follow on a day which matches more than one calendar pattern. Considering the priorities the calendar preview shows the effective schedule on a given day.

To Configure a Scheduled Event

1. Open the **Configure Schedule** dialog and click on the **Configuration** tab (see Section 10.10.4).

2. In the calendar view select the day for which to configure the scheduled event. Then select the event duration by clicking into the daily planner and dragging the mouse to the desired duration.



3. Then click on the button **Create new scheduled event**  and enter a **Name** for the scheduled event (note, in BACnet there is no name to be specified). Choose the scheduled **value** or enter a scheduled value. Modify the **Start** and **End** time to your needs. Optionally you should set a priority, if scheduled events overlap in the preview.

Event

Name

Value

Start End

Priority

Duration

4. Choose an **Event type**, which defines how the event is recurring.

Event type

☐ One-time

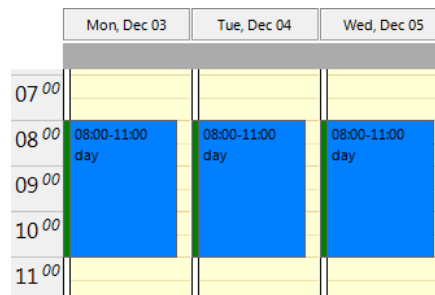
☒ Daily


☐ Weekly

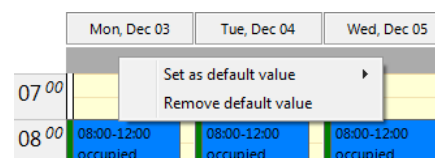
Start

End

5. The click **OK**. The new event appears in the daily planner.



6. For viewing more details on overlapping events you can change to the **day view** . In all views events can be extended or shrunk using the upper or lower boundary handles or moved by clicking and dragging.
7. Right-click on an event in order to edit it. You can choose to modify it, change its color, or delete it.
8. To set a default value, right-click onto the grey area right above the day planner. In the context menu choose a default value.



To Use Exception Days from a Calendar

1. For letting a scheduled event occur on exception days from a calendar, create a new scheduled event.
2. Select the event type **Calendar** and choose one of the defined **calendar date entries**.

3. Note, that if the scheduled event may overlap with other events. In this case edit the priorities of the scheduled events. For example, if a given calendar day falls in categories, “Holidays” and “Maintenance”, the scheduled event with the higher priority becomes effective. The highest available priority is marked **highest**. Note that the actual priority values depend on the technology (see Section 10.4.2).

NOTE: Choose different priorities for different overlapping scheduled events. If two scheduled events overlap and their priorities are equal, it is not determined which value is in effect.

10.10.6 Configure Exception Days

When a local calendar is used, its calendar patterns need to be configured with exception days (pattern entries). The calendar patterns can be configured in the configuration software, modified at run-time over the web UI or over the network. When configuring via the software, the current exception days can be uploaded from the device first, to work on the current configuration.

To Configure Exception Days in a Calendar Pattern

1. Select the **Calendar** folder and the calendar pattern which shall be configured.

	No.	Direction	Calendar Name	Index	Func. Block	Use	ID
📅	1	In	calendar			1	1030
	1.1		Holidays				1032

2. Right-click and select **Configure Pattern ...** in the context menu.
3. The **Configure Pattern** dialog appears as shown in Figure 26. Add dates to the calendar pattern by entering a Date Configuration. Then click **Add Entry**. The date appears in the Pattern Entries list on the right-hand side.
4. Edit an exception by selecting the pattern entry in the **Pattern Entries** list. Then modify the date configuration in the **Date Configuration** group box.

Figure 26: Configure Calendar Pattern Dialog.

5. Click **Save Changes** when all exception days have been entered.


NOTE: *If unsure how a date configuration affects the calendar days, click on a pattern in the **Pattern Entries** list and the affected days will be highlighted in the **Preview**.*

10.10.7 Configure Control Data Points

A scheduler object can be configured to use special control data points. These data points can control the scheduler and expose additional state information of the scheduler on the network. The following control data points are available:

- **Scheduler Enable/Disable Datapoint:** This data point can be used to enable or disable the scheduler depending on its value.
- **Enable/Disable Feedback Datapoint:** This data point is updated with the current enabled state of the scheduler. This also reflects a change in enabled state received from the network.
- **Scheduled Preset Name:** This data point can be attached to receive the name of the currently active preset. Naturally, only string data points can be attached.

To Configure Control Data Points

1. Open the **Configure Schedule** dialog to configure daily schedules as described in Section 10.10.4.
2. Go to the **Scheduled Datapoints** tab.
3. In the **Control Datapoints** group box, click the  button to add the desired control data point. A data point selection dialog opens.

4. Select a matching data point and click **OK**. For the preset name a string data point must be selected.
5. To remove an undesired control data point, click on the **Remove** button.

10.10.8 Using the SNVT_tod_event

On LOYTEC devices supporting CEA-709 networks, the SNVT_tod_event can be used in a schedule for implementing the next-state feature. The parts of this network variable are:

- Current state: This is the currently scheduled occupancy state.
- Next state: This is the next, future occupancy state in the schedule.
- Time to next state: This part reflects the time in minutes until the next state becomes active.

To Use a SNVT_tod_event

1. Create a SNVT_tod_event in the data point configuration.
2. Add the SNVT_tod_event to the scheduled data points of a scheduler as described in Section 10.10.4.
3. All three parts of the SNVT_tod_event are scheduled.

No.	OPC	Direction		Datapoint Name
1	<input checked="" type="checkbox"/>	Out		nvoTodEvent
1.1	<input checked="" type="checkbox"/>	Out		current_state
1.2	<input checked="" type="checkbox"/>	Out		next_state
1.3	<input checked="" type="checkbox"/>	Out		time_to_next_state

10.10.9 Schedule Configuration Data

It is important to note that the configuration of the scheduler and calendar objects consists of two parts:

One is the point setup, meaning the number of calendar patterns, number of scheduler objects, and which points are controlled by which scheduler. This information is part of the data point setup of the project, similar to the definition of local and remote network objects. It is always in effect when the project is downloaded and it will have an influence on the static interface of a CEA-709 device.

The other part is the dates entered in calendar patterns and the actual schedule of each scheduler (when the scheduler should output which value preset as well as the definition of the value presets themselves). This information is usually changed during the runtime of the device, that is, it is considered dynamic data, similar to the current value of a network variable or server object (this data is in fact the current value of such objects). Therefore, this data is not reset with every project download, since the user wants to keep the current values if possible. Only if the scheduler setup itself changed, the device will not be able to take over the current settings, since they do no longer apply.

NOTE: *The scheduler and calendar configuration entered in the configuration dialogs is stored together with the project. It can also be exported to XML format or imported from XML format. Once the project itself was downloaded to the device for the first time, the scheduler configuration should usually be downloaded as well to initialize the created objects with some data. Since this is normally not wanted in subsequent project downloads, the configuration software will normally ask the user if the scheduler and calendar objects should be initialized with the configuration data from the project or not. The behavior may be changed in the Project Settings Dialog between ask, always download, and never download. In any case, the configuration may be downloaded and also uploaded at any time from the Connection menu.*

10.10.10 System Resources

To actually implement the scheduler functionality, network technology dependant software modules are required on the device. The scheduler and calendar data points created here will be connected to these modules and serve as the link to exchange configuration data between the user and the actual scheduler unit.

NOTE: *It is important to understand that the scheduler data points used in the project are not actually the scheduler units but only serve as a link to exchange configuration information and thus control the actual scheduler, which may be located on a completely different device. The device containing the actual scheduler may even be a third party device which is compliant to the standards for scheduling defined for the respective network technology.*

In case of local scheduler objects, the required scheduler units are either automatically instantiated (BACnet devices automatically create the required server objects they need to operate the scheduler) or need to be defined by the user in a separate dialog.

For CEA-709 devices, the configuration software provides a *Network Settings* Dialog, where the required number of scheduler units may be instantiated and their capacity may be configured (how many time/value entries, value templates, bytes per value template, and so on). The dialog can be accessed through the *File* menu and contains the following options and settings which are relevant to calendar and scheduler functionality of the device:

- **Enable Calendar Object:** This checkbox enables a LonMark compliant calendar object on the device. It is automatically enabled together with local schedulers, since the two are always used together.
- **Enable Scheduler Objects:** This checkbox enables local LonMark compliant scheduler objects on the device. Checking this box will automatically enable the calendar as well.
- **Number of calendar entries:** Specifies the maximum number of different exception schedules (day classes like holiday, maintenance day) supported by this calendar object.
- **Total number of date entries:** Specifies the maximum number of date definitions which may be stored by the calendar. This is the sum of all date definitions from all calendar entries. A date definition is for example a single date, a date range, or a week and day pattern (every last Friday in April).
- **Number of local schedulers:** This is the number of local scheduler objects which should be available on the device. Each local scheduler data point created in the data point manager will connect to one of these scheduler objects. There may be more scheduler objects available on the device than are actually used at a certain time. It is a good idea to have some spare scheduler objects ready, in case another scheduler is needed.
- **Number of daily schedules:** This is the maximum number of schedules supported by each scheduler object. This number must be at least 7, since a

scheduler always needs to provide one schedule for each day of the week (default weekly schedule). For each special day defined by the calendar, an additional daily schedule is required to support it.

NOTE: *If the scheduler units are used with LWEB-900 or are displayed on a schedule control in extended UI mode, this value must be set to a high number, because one daily schedule will be required for each event that is entered.*

- **Entries in Time/Value table:** This is the total number of entries in each scheduler defining a value template that should apply on a specific day starting at a specific time (the time table).
- **Number of value templates:** This is the maximum number of value templates supported by each scheduler.
- **Data size per value template:** This specifies the buffer size reserved to hold the data for each value template. More data points or bigger data structures require a bigger value buffer.
- **Max. number of data point maps:** Specifies the maximum number of individual data points that this scheduler is able to control.

As can be seen from the above list, it is not easy to configure a LonMark scheduler object. There are many technical parameters which need to be set and which require some knowledge of how these scheduler objects work internally. Therefore, the configuration software provides the following mechanisms to help in choosing the right settings:

- **Resources required by the current project:** The absolute minimum settings required by the current project are shown in a table at the left side of the window. This data may be used to fill in the values at the right side, but some additional resources should be planned to allow for configuration changes which need more resources.
- **Auto-Set:** This button may be used to let the configuration software decide on the best settings to use, based on the current project. Since the current projects resource usage is taken as a starting point, all schedulers and calendar patterns in the project should first be configured as required before this button is used.
- **Set Defaults:** This button will choose standard values for all settings. In most cases, these settings will provide more resources than necessary.

NOTE: *It is possible to enter anything here, until the project is actually saved or downloaded. At this point in time, the software will check that the resources configured here are sufficient to support the projects configuration. If this is not the case, this dialog will automatically open so that the settings may be adjusted.*

10.10.11 Resource allocation

On BACnet devices, there is always a direct assignment between a local scheduler point and a server object, because the two are created and deleted together. Name and instance number of the created server object for the scheduler may be set in the same way as for any other server object.

For CEA-709 devices, resource allocation is more complicated. Once the required resources are configured in the *Network Settings* Dialog, the local scheduler data points will be assigned to the available scheduler objects of the device. Each new scheduler point will pick the first available scheduler object, until all objects are used. Scheduler points which cannot be assigned to a free scheduler will remain unassigned until enough schedulers are created or other scheduler points are deleted. The assignment is shown in the columns *Index*

and *Object* in the point list. These are the functional block index and name of the assigned scheduler.

Deleting a scheduler data point will detach the data point from the respective scheduler object and make this object available for other scheduler points which may still be unassigned due to missing resources or which may be created in the future. The assignment of other scheduler points to their scheduler objects is not changed.

10.10.12 Using the Scheduler

Once the above setup is done, the scheduler is basically operational. It will immediately start to work based on the configuration data downloaded through the configuration software. However, in almost all cases it will be required to add a control to the visible part of the project, through which the schedules may be viewed and edited. For this purpose, there is a specialized type of control available, the *Schedule Control*. To use it, create a control of this type on the page and connect one or more scheduler data points to it. Please see section 9.5.9 for more information about schedule control configuration and sections 7.2.1 and 7.2.3 for a guide on how to operate a schedule control on the device.

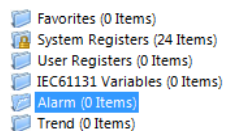
10.11 Local Alarming

10.11.1 Create an Alarm Server

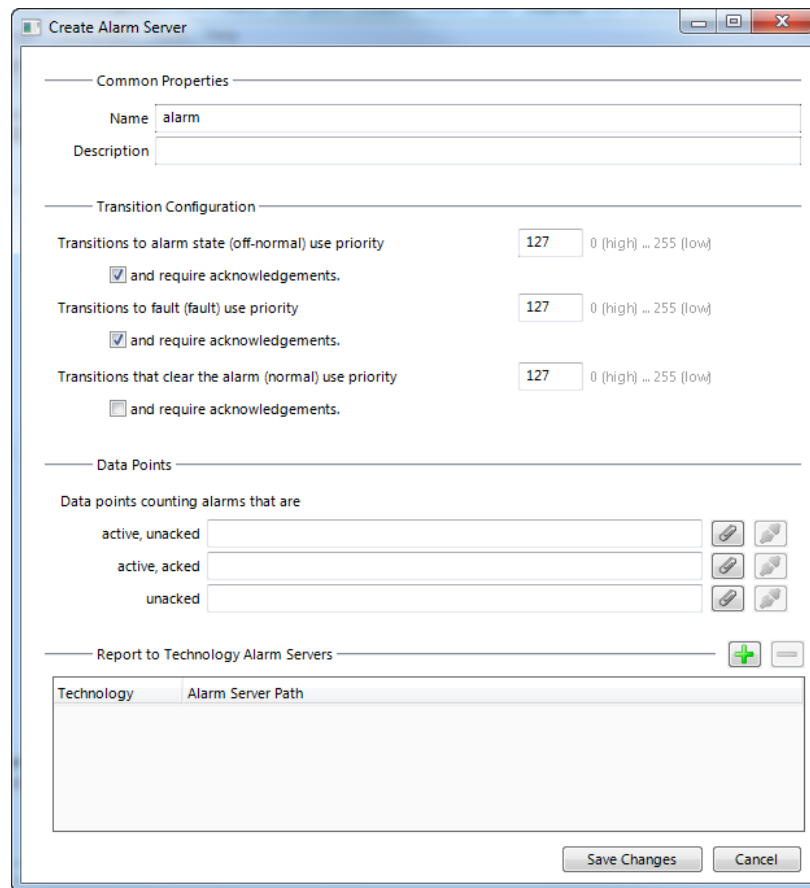
To generate local alarms, an alarm server needs to be created at first. The local alarm sources will report alarms to that alarm server. The alarm server is the interface to access local alarms.

To Create an Alarm Server

1. Under the device folder, select the **Alarm** sub-folder to create a generic alarm server. For a technology alarm server, select the **Alarm** sub-folder of the respective port.



2. Right-click in the data point list view and select **New Alarm Server**
3. In the **Create New Alarm Server** dialog box (as shown in Figure 27) enter **Name** and **Description** of the alarm server.




The 'Create Alarm Server' dialog box is divided into several sections:

- Common Properties:** Includes fields for 'Name' (set to 'alarm') and 'Description'.
- Transition Configuration:** Contains three rows for different transition types:
 - 'Transitions to alarm state (off-normal) use priority': Priority set to 127, with a checkbox 'and require acknowledgements' checked.
 - 'Transitions to fault (fault) use priority': Priority set to 127, with a checkbox 'and require acknowledgements' checked.
 - 'Transitions that clear the alarm (normal) use priority': Priority set to 127, with a checkbox 'and require acknowledgements' unchecked.
- Data Points:** A section titled 'Data points counting alarms that are' with three rows: 'active, unacked', 'active, acked', and 'unacked'. Each row has an input field and two small icons (a key and a lock).
- Report to Technology Alarm Servers:** A section with a green plus icon and a minus icon. Below it is a table with two columns: 'Technology' and 'Alarm Server Path'.

At the bottom right are 'Save Changes' and 'Cancel' buttons.

Figure 27: Create New Alarm Server dialog box.



4. For alarm transitions, you may define which ones require acknowledgement, and at which priority they are reported.
5. You may attach data points for storing alarm counters. These will be linked using the respective property relations (see Section 10.1.10).
6. When you create a generic alarm server, you may add technology alarm servers that shall be reported to. Click on  and choose one alarm server of each technology. If they don't exist, you need to create them under the technology port folders.
7. Click **Create**. The alarm server appears now in the data point list view.
8. For a BACnet technology alarm server, edit the instance number of the Notification Class object to fit your needs.

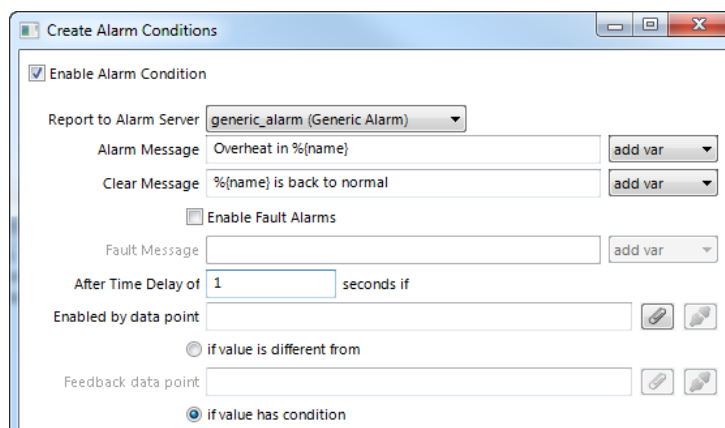
10.11.2 Create an Alarm Condition

To generate alarms from data points, intrinsic reporting is used. For each data point an alarm condition must be defined. This condition employs an intrinsic algorithm to generate alarms based on the data point's value or by evaluating a feedback value. Depending on the data point type (analog, binary, multi-state), different conditions are defined. The alarm is reported to the attached alarm server.

To Create an Intrinsic Alarm Condition

1. Select a data point.
2. Right-click and select **Create Alarm Condition...** from the context menu.

3. For the alarm condition, edit the following definitions which apply to all condition types as shown in Figure 28. Select the **Alarm Server** which the alarm shall be reported to. Typically, you will choose a generic alarm server.
4. Enter an **Alarm Message**. This is shown when the alarm becomes active. You may add variable placeholders to this message by selecting one from the drop-down box **add var** on the right-hand side. Enter a **Clear Message**. This is shown when the alarm clears.
5. Check the option **Enable Fault Alarms**, if fault conditions (offline, unreliable) shall generate fault alarms. If enabled, enter a **Fault Message**, which is displayed along with the fault alarm when it occurs.
6. Optionally, enter a **Time Delay**, for which the condition must persist before the alarm becomes active or is cleared again. The delay is entered in seconds.
7. By clicking  you may attach a data point, which is evaluated for enabling the alarm. This can also be done by editing the property relation 'enableAlarm' (see Section 10.6.9). Detach the data point by clicking .
8. Choose the option **value is different from** to define a feedback alarm. In this case the setpoint value of the alarmed data point is compared against the feedback value. A feedback data point can be attached for this purpose. This can also be done by editing the property relation 'feedbackRelation' (see Section 10.6.9).
9. Choose the option **value has condition** to define a value alarm. In this case the data point value is compared against the condition. Edit the condition in the box below this option.



Create Alarm Conditions

☒ Enable Alarm Condition

Report to Alarm Server: generic_alarm (Generic Alarm)

Alarm Message: Overheat in %(name) add var

Clear Message: %(name) is back to normal add var

☐ Enable Fault Alarms

Fault Message: add var

After Time Delay of: 1 seconds if

Enabled by data point: attach detach

☒ if value is different from

Feedback data point: attach detach

☐ if value has condition

Figure 28: Common settings for an alarm condition.

10. For an analog feedback condition, fill in the alarm condition as shown in Figure 29. A feedback alarm is generated, if the setpoint **differs by** – and **differs by** + value from the feedback value. Enter a **Deadband** to account for hysteresis. Attach or detach data points for those limits. This can also be done by editing the property relations 'lowLimit', 'highLimit', and 'deadband', respectively (see Section 10.6.9).

Figure 29: Condition for an Analog Feedback Alarm.

11. For an analog value condition, fill in the alarm condition as shown in Figure 30. Select **Low Limit** and **High Limit** and put check marks, if they shall be employed. Enter a **Deadband** to account for hysteresis. Attach or detach data points for those limits. This can also be done by editing the property relations 'lowLimit', 'highLimit', and 'deadband', respectively (see Section 10.6.9).

Figure 30: Alarm Condition for an Analog Value Condition.

12. For a binary data point, define an alarm value in the alarm condition as shown in Figure 31. Select the **Alarm Value** which triggers the alarm.

Figure 31: Alarm Condition for a Binary Data Point.

13. For a multi-state data point, define the alarm values in the alarm condition as shown in Figure 32. Select the states in the list **Not Alarmed** and move them to **Alarm on States** by clicking the arrow buttons.

Figure 32: Alarm Condition for a Multi-State Data Point.

14. Click on **Create**. In the alarm column, the alarm sign 🚨 will be added for those data points that have an alarm condition. If a sub-data point has been alarmed, the top-level data point will indicate this with the sign 🚨.

10.11.3 Deliver Alarms via E-mail

Updates in the alarm summary of an alarm object can be used as a trigger to send e-mail. For setting up e-mails, the account information has to be configured on the device, e.g., on the web UI. Then an e-mail template can be created and the alarm point attached as a

trigger. The e-mail template can be configured to contain certain alarm information in the text or subject field.

To Create an E-mail Template for Alarms

1. Create or configure an e-mail template as described in section 10.9.
2. Change to the **Mail Triggers** tab.
3. Click the **Add...** button and select an alarm data point.
4. In the Mail Triggers list select the added trigger data point.

Datapoint	Type	Condition
Critical	Alarm	-

5. In the **Manage Trigger Conditions** list put a check mark on alarm conditions that shall invoke the transmission of the e-mail.

Enabled Conditions	
<input checked="" type="checkbox"/>	State to Active
<input type="checkbox"/>	State to Ackd-Active
<input type="checkbox"/>	State to Ack-Pending
<input type="checkbox"/>	State to Ackd
<input type="checkbox"/>	State to Inactive
<input checked="" type="checkbox"/>	Offnormal-Alarms
<input checked="" type="checkbox"/>	Lowlimit-Alarms
<input checked="" type="checkbox"/>	Highlimit-Alarms
<input checked="" type="checkbox"/>	Fault-Alarms
<input checked="" type="checkbox"/>	Buffer-Alarms

6. Change to the **Common E-Mail Properties** tab.
7. Add the desired alarm data point as a data source.
8. Choose the desired alarm information from the fields in the drop-down **Selected Data Source Value** and paste a place holder into the e-mail text or subject field.


Datapoint	Var.
LINUX-120.CEA709 Port.Alarm.Critical	v1

Alarm value: to subject Paste

10.11.4 Multi-Edit Alarm Conditions

For editing a large number of alarm conditions, some multi-edit features are available to assist in this task. On a multi-selection of data points, the user can execute two options from the context menu:

- **Configure Alarm Conditions:** Use this option from the context menu on a multi-selection of data points. The alarm condition of all selected data points can be edited. If all selected data points are of the same class (e.g., all analog data points) the alarm condition can be fully specified. Note that these settings will be applied to all data points. In alarm messages use variable place holders. For assigning different limit and enable data points use the manage relations tab (see Section 10.6.9).
- **Configure Alarm Messages:** Use this option from the context menu for editing alarm messages (alarm, clear, fault) for all data points in the selection. A dialog

with a spreadsheet view appears as shown in Figure 33. Edit the messages directly in the spreadsheet. Alternatively, click on the export button  to export the spreadsheet as a CSV file for editing in Excel.

NOTE: Use the feature to include data points from sub-folders and filter expressions to expand the ability to perform multi-select.

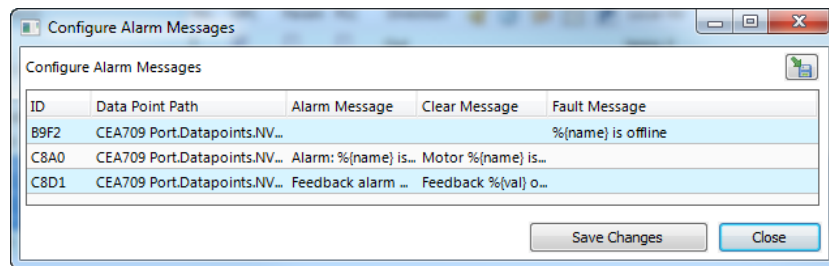


Figure 33: Spreadsheet for multi-edit of alarm messages.

10.12 Local Trending

10.12.1 Create a Local Trend

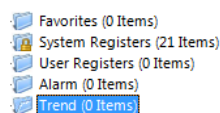
The value of a data point can be logged over time. This is referred to as trend data. To generate trend data a trend object has to be created. The trend data is stored in a data logger file.

Trend objects can generate trend logs for single and multiple data points and can be operated in one of the following modes:

- **Interval Mode:** In this mode a snapshot of all trended data points is logged into the data logger file. Aligned log intervals can be configured.
- **COV Mode:** In this mode, each of the trended data points is logged separately, if and only if its value changes. For analog data points, a specific COV increment can be configured in the data point configuration properties of the trended data point.
- **Trigger Mode:** In this mode a snapshot of all trended data points is logged each time a trigger condition fires. The trigger condition is applied to a trigger data point.

To Create a Trend Object

1. Select the **Trend** folder of the device.



2. In the data point list right-click and select **New Trend ...** from the context menu.
3. In the **Create New Trend Object** dialog (shown in Figure 34), enter a name and optionally a description for the trend log object.

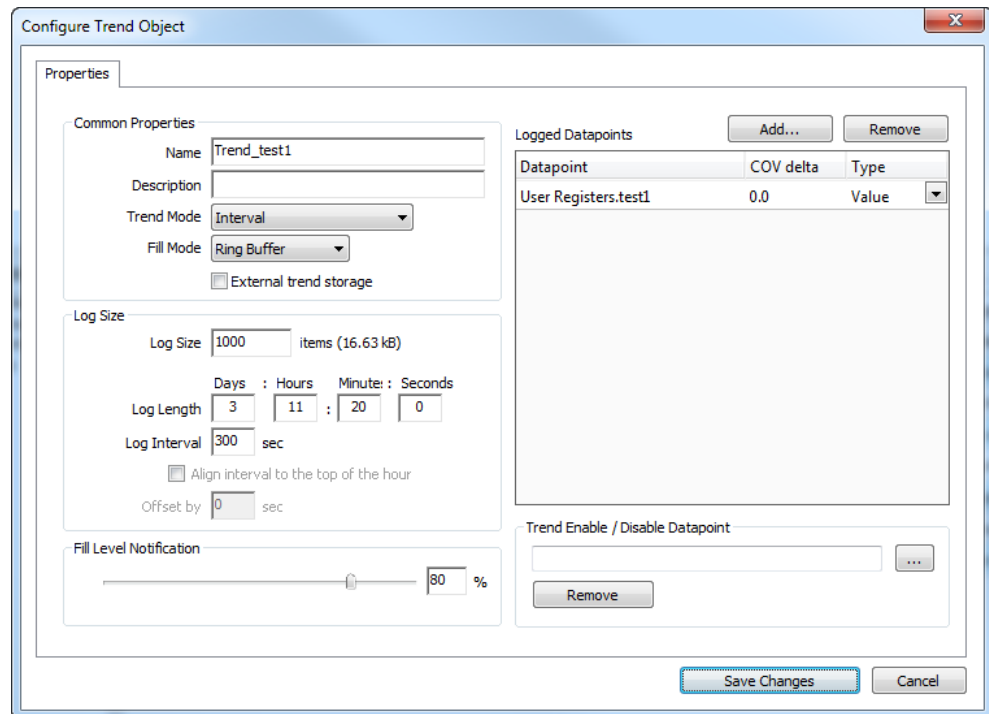


Figure 34: Basic Trend Object Configuration.

4. Select the desired **Trend Mode**.
5. On devices with SD cards, select **External trend storage**, if this trend log shall be backed up to an SD card. If doing so, also set the **Fill Level Notification**, which triggers when a backup is written to the external storage.
6. Select the **Log Size**. The display in the dialog will adapt the estimations for needed data logger file size in KB and duration of the trend log. Alternatively, for interval trends, the estimated log duration and log interval can be edited.
7. In the interval trend mode the **Align interval** option can be activated. Depending on the selected interval, the beginning of the interval is aligned to the wall time (e.g. every 15 minutes aligned to the top of the hour). An additional offset in seconds to that alignment can be specified (e.g. 5 seconds after those 15 minutes).
8. Select a **Fill Level Notification** percentage. This will decide at which fill-level trigger will fire. A fill-level trigger can be used to trigger the transmission of an e-mail (see Section 10.12.4) or a backup of log data to the SD card.
9. Click **Save changes** to store the basic configuration of the trend object. The new trend log object appears in the data point list of the Trend folder.

10.12.2 Configure Trended Data Points

When a local trend object has been created, it needs to be configured, which data points it shall log. This is done by attaching data points to the trend object. Only simple data points can be attached for trending, i.e., of class analog, binary, or multi-state. For trend log objects in the BACnet technology, single data points can be attached only.

The trending can be enabled and disabled on behalf of an *enable* data point. This data point should be of type *binary*. If the value of that enable data point is TRUE, the trend object logs data as defined by the trend mode. If the value of the enable is FALSE, trending is disabled. If no enable data point is configured, the trend log is always enabled.


To Attach Data Points for Trending

1. Select the trend object in the **Trend** sub-folder.

No.	Direction	Trend Name	Use	ID
1	Out	TestTrend	0	1014

2. Right-click and select **Configure Trend** from the context menu. The same dialog which appears when a new trend object is created is shown and allows configuring the trend object. Of course, this step can also be done directly when the object is created.
3. Add data points to be trended. Click on **Add ...** which opens a data point selector window.


Logged Datapoints			Add...	Remove
Datapoint	COV delta	Type		
abs_humid1	0.0	Value		

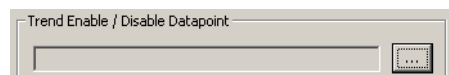
4. Select the data points and click **OK**. For each of the attached data points, a line appears in the list below the add button. The trended data points will also appear with the trend icon  in the data point manager.

NOTE: *Data points can also be attached to a trend by selecting a data point in the data point manager, drag it onto a trend object, and drop it on the trend object.*

5. Data points can be removed from the trend by clicking **Remove**.
6. If COV mode was selected, the COV increment is displayed in the **COV delta** column. This value can be increased to produce less trend data. Note that it cannot be lowered below the trended data point's own COV increment. Go to the data point configuration to change the COV increment of the data point in this case.
7. If the trended value of the data point shall be aggregated over the log interval, select the desired aggregation in the **Type** column. Available options are **Min**, **Max**, **Avg**.

NOTE: *To create multiple curves with minimum, average, and maximum values, add the same data point three times and select the different aggregation types.*

8. In addition, a special **Trend Enable** data point can be selected. If configured, the trend log will only log data when the value of this data point evaluates **true**, i.e., is not zero. Click the  button to select a data point.



9. To remove the enable data point, click the **Remove** button.
10. When done with the data point setup, click **Save Changes** to leave the dialog.

NOTE: *A shortcut for creating a trend log object and attaching a data point is to select a data point in the data point manager, right-click on it and choose **Trend Datapoint** from the context menu. This generates a trend log and links that data point to it.*

10.12.3 Trend Triggers

Local trend objects can be operated in *trigger mode*. In this mode, one or more trigger data points cause the generation of a snapshot containing the values of the trended data points at the time instant the trigger is activated. For a trend object, one or more trigger conditions

can be defined. Depending on the trigger data point type, the trigger conditions can be refined.

Note, that the behavior of the trigger data point is influenced by the COV properties of the data point. If the **Only notify on COV** property is checked, the data point triggers only if its value changes to the value of the trigger condition. If that property is not checked, the data point triggers on every write with a value that matches the trigger condition.

To Configure Trigger Data Points for Trending

1. Select the trend object in the **Trend** sub-folder.

No.	Direction	Trend Name	Use	ID
1	Out	TestTrend	0	1014

2. Right-click and select **Configure Trend** from the context menu.
3. Change to the **Triggers** tab.

NOTE: *Of course, you can also change directly to the **Triggers** tab when creating a trend object.*

4. Click the **Add...** button. A data point selection dialog opens.
5. Select one or more data points and click **OK**.
6. The triggers appear now in the **Trend Triggers** list.

Trend Triggers			Add...	Remove
Datapoint	Type	Condition		
state	Value Update	-		

7. In the **Manage Trigger Conditions** you can refine the trigger condition depending on the trigger data point class.
8. When done with the data point setup, click **Save Changes** to leave the dialog.

10.12.4 Deliver Trend Data via E-mail

Trend logs can be downloaded from the device via file transfer. This requires an active action by the user. Alternatively, trend data can be sent as an e-mail attachment. For doing that, an e-mail template has to be setup for the trend log to be transmitted. The fill-level condition in the trend object can be used as a trigger to send an e-mail with the trend's data logger CSV file as an attachment.

For setting up e-mails, the account information has to be configured on the device, e.g., on the web UI. Then an e-mail template can be created and the trend object attached as a trigger.

To Create an E-mail Template for Trends

1. Create or configure an e-mail template.
2. Change to the **Mail Triggers** tab.
3. Click the **Add...** button and select a trend object.
4. In the **Mail Triggers** list, the added trigger data point appears with the **Fill Level** condition.

E-Mail Triggers		
Datapoint	Type	Condition
TestTrend	Fill Level	

5. Change to the **Attachments** tab.
6. Select the trend log CSV file of the trend object in the **Attach File** drop-down box and click **Add**.

NOTE: *ZIP versions of the CSV files are also available. Select those to save transmission bandwidth and mailbox space.*

Attachments		Attach File	TestTrend_1014.csv	Add
Attachment	Device File Path	Add Datetime		Remove
TestTrend_1014.csv	/tmp/uid/trend/1014.csv	<input checked="" type="checkbox"/>		

7. Click **OK** to complete the e-mail template configuration.

10.12.5 Technology Trends

In the BACnet technology, trend logs can be exposed on the BACnet port via special BACnet Trend Log objects. To create a technology trend select the port folder (e.g., **BACnet Port**) and then the **Trend** folder underneath. Follow the same instructions as described in Section 10.12.1. Please note that certain restrictions apply to BACnet trends (see Section 10.5.3).

10.13 Trend Data Points and Trend Controls

Since trends may be recorded either directly by a trend control or independently by a trend data point, the project designer needs to decide on a case-by-case basis which of the two methods is to be used. Here are some of the aspects to consider:

10.13.1 Recorded by a Trend Control

- Trends recorded by the trend control itself are optimized for the graphical representation on the screen. The control records exactly the amount of data required to draw a smooth trend graph, given the resolution and size of the trend control on the screen.
- A trend control can record up to 8 different scalar data points at the same time, even on BACnet L-Vis devices. No Server object is required and there is no limit of one point per recording. This makes it easy to record for example the CPU load, which is a system register.
- The recorded data is tightly coupled with the trend control. In case the control is removed, the data is deleted as well. Also, if the graphical layout of the control changes, the data may need to be deleted as well since it no longer fits the controls size.
- The recorded data will not be accessible to remote devices on the network, even on BACnet L-Vis devices.

10.13.2 Recorded by a Trend Data Point

- Trends recorded by an independent trend data point will record data at intervals which are not optimized for a particular use, so that the graphical representation may not be as smooth and processing may not be as fast as with data recorded for a particular trend control.

- Trend objects are not related to a particular way of visualization and can operate even without any trend control showing the data on the local device. The advantage is that the visualization may be changed or even temporarily removed from the project, without also removing the trend recording.
- In cases where the same trend recording needs to be shown on different pages in the project, using an independent trend object to record and hold the data in one place may be preferable to having multiple independent trend controls record the same data multiple times.
- In case of BACnet L-Vis devices, data recorded by a trend object is accessible to remote devices on the network. However, this places some restrictions on the functionality of the trend object. Only one data source may be trended and it must be a BACnet server object located on the local device.

10.14 Remote AST Objects

10.14.1 Remote Scheduler and Calendar

Adding remote access to the configuration of a scheduler and calendar which is located on another device is done by creating remote scheduler and calendar points. These points may either be created from data obtained by a network scan, or can be created manually (CEA-709 only). Here are the steps to create the required points from a network scan:

- Execute a network scan, as described earlier in this document.
- From the points in the import folder, select the scheduler points you are interested in and use the command *Use on Device* to create suitable remote scheduler points in your project.
- Note that any existing calendar is automatically imported as well.
- Adjust the basic settings for the newly created points, which are now available in the folder *Remote Devices*, list the object name and description. The object name will be used as the name for the scheduler, as seen on the display later on.
- For BACnet, also adjust the poll cycle, which will be used to periodically fetch the current configuration in case the remote device does not support COV subscriptions.
- For CEA-709, a static NV is created to receive information from the remote device about changes to the scheduler configuration, so that the local device does not need to poll the remote device. Set a name for this NV (default is *nviSchedLink<number>*) and assign it to a suitable function block.

NOTE:	<i>Due to the static input NV which is required for a remote CEA-709 scheduler point, adding remote scheduler points will change the static interface of the device.</i>
--------------	--

To keep the project reusable in different networks, it is desirable to create the remote scheduler and calendar data point manually on CEA-709 devices and define the connection to the scheduler server by the network binding of the link NVs only. Instead of using data from a network scan, the required data points can be created as follows:

- Select the data point folder named *Calendar* and use *New Remote Calendar...* from the context menu in the data point list to bring up a dialog to create a new remote calendar point.

- Enter a name for the calendar point, a name for the input NV which will be used to connect the calendar from another device to this data point, and select the desired functional block in which the NV should be located.
- Press OK to create the calendar point (repeat for more calendar points if required). There should be one calendar point for each device from which schedule data is viewed, since the calendar point will carry the information about exception days used in the schedules.
- Select the data point folder named *Scheduler* and use the context menu to create a *New Remote Scheduler* point. Enter a name for the data point, a name for the input NV which will be used to connect the point to a scheduler server and select the functional block in which the NV should be located.
- Press OK to create the scheduler point and repeat for additional points as required. Each of the created scheduler points will later provide access to the configuration of the scheduler unit to which the scheduler link NV was bound in the network (see below).

On CEA-709 devices, there is one extra step to take before the new data points will be operational: The new static input NV representing the remote calendar on the local device (this NV is normally called *nviCalLink*) needs to be bound to the output NV called *nvoCalLink* located in the Calendar functional block of the remote device and the new static *nviSchedLink* NVs which were created for each remote scheduler point need to be bound to the respective *nvoSchedLink* variable located in the Scheduler functional block of the remote device. The binding between the *nvoSchedLink* variable on the remote device to the *nviSchedLink* variable on the local device defines which of the scheduler data points on the local device connect to which scheduler unit on the remote device.

On BACnet devices, the new data points can be used right away to exchange configuration data with the scheduler on the remote device. Just connect the new scheduler data point to a schedule control to view and edit the configuration of the remote devices scheduler.

10.14.2 Alarm Clients

Accessing alarm server objects on remote devices is done by creating remote alarm data points. These points may either be created from data obtained by a network scan, or can be created manually (CEA-709 only). Here are the steps to create the required points from a network scan:

- Execute a network scan, as described earlier in this document.
- From the points in the import folder, select the alarm server points you are interested in and use the command *Use on Device* to create suitable alarm client points in your project.
- For CEA-709, select the new alarm client point and adjust the name of the local NV (default name is *nviAlarm_2*). Also, you may assign this NV to one of the L-VIS functional blocks as required.

NOTE:	<i>Due to the static input NV which is required for a CEA-709 alarm client point, adding alarm clients will change the static interface of the device.</i>
--------------	--

To keep the project reusable in different networks, it is desirable to create the remote alarm data point manually on CEA-709 devices and define the connection to the alarm server by the network binding only. Instead of using data from a network scan, a remote alarm data point can be created as follows:

- Select the data point folder named *Alarm* and use *New Remote Alarm...* from the context menu in the data point list to bring up a dialog to create a new remote alarm point.
- Enter a name for the data point, a name for the input NV which will be used to connect the alarm server of another device to this data point, and select the desired functional block in which the NV should be located.

On BACnet devices, the new data points can be used right away to exchange alarm information with the alarm server on the remote device. Just connect the new alarm client data point to an alarm list control to view and acknowledge alarms reported by the associated alarm server.

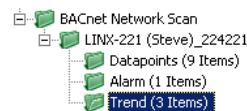
On CEA-709 devices, there is one extra step to take before the new data point will be operational: The new static input NV representing the alarm client on the local device need to be bound to the alarm output of the remote device from which alarms should be received. A CEA-709 device normally delivers alarms through an output NV of type SNVT_alarm_2 located in the node object of the device, therefore the new input NV on the local device must be bound to the alarm output NV of the remote devices node object.

10.14.3 Remote Trend Logs

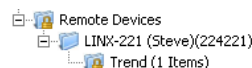
A remote trend log provides access to trend log data, which is actually generated and stored on another device. The remote trend log can load trend data from that device and supply it to the L-Vis application or a CSV file. Please note that remote trend logs are currently only supported by BACnet and OPC technology.

To Create a Remote BACnet Trend Log

1. Execute a network scan, as described earlier in this document. The scan folder is filled with available trend logs.



2. From the data points in the import folder, select the trend log objects you are interested in and select **Use on Device** from the context menu. This creates suitable remote trend logs in the **Remote Devices** folder.



To Create a Remote OPC Trend Log



Import the LINX configuration containing the trend log to be accessed. This is done by opening the OPC Server manager mentioned in section 10.6.11 and loading the required configuration file. This will create new OPC client data points, including any trend log points available on the imported LINX.

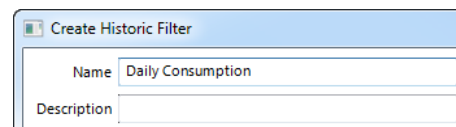
10.15 Historic Filters

10.15.1 Create Historic Filters


Historic filters are used to work with historic values of a base data point. These historic values are derived by defining historic filter functions. An historic filter template is a collection of such filter functions and can be assigned to any analog, binary, or multi-state base data point. For more general information on historic filters refer to Section 10.3.5.

To Create Historic Filters


1. Select one or more data points that shall serve as the base data points.
2. Right-click and select the **Configure Historic Filters ...** item in the context menu. As an alternative, click on  of the **Historic Filter** data point property.
3. The dialog **Select Historic Filter** opens. Click on  to create a new one.
4. In the **Create Historic Filter** dialog enter a name and optionally a description.






The dialog box has a title bar 'Create Historic Filter'. It contains two text input fields: 'Name' with the value 'Daily Consumption' and 'Description' which is empty.

5. To add a new filter function to the list click on the add button . Enter a filter entry **Name** and choose the desired period **Type**. The name will appear with the historic filter relation and helps identifying it. Depending on the type define the arguments **Day** and **Time**. Select how many **samples ago** the filter goes back. The most current sample is '0', the previous one is '1'.

Filter Entries

No.	Name	Type	Day	Time	samples ago	Delta to current
0	midnight_today	Value at hh:mm:ss of the day	N/A	00:00:00 h	0	










6. To duplicate an entry click on the duplicate button . On the duplicate modify the settings accordingly.

No.	Name	Type	Day	Time	samples ago	Delta to current
0	midnight_today	Value at hh:mm:ss of the day	N/A	00:00:00 h	0	
1	midnight_yesterday	Value at hh:mm:ss of the day	N/A	00:00:00 h	1	

7. For getting the difference of an historic value to the current value check the box **Delta to current**.





No.	Name	Type	Day	Time	samples ago	Delta to current
0	midnight_to_now	Value at hh:mm:ss of the day	N/A	00:00:00 h	0	<input checked="" type="checkbox"/>

8. Click on **Save Changes** and select the created filter template. For each filter entry defined, a historicFilter property relation is created under the base data point(s).

energyCount	1	<input checked="" type="checkbox"/>				Value	
historicFilter (midnight_to_now)	1.1					In	
historicFilter (midnight_today)	1.2					In	
historicFilter (midnight_yesterday)	1.3					In	

10.15.2 Managing Historic Filter Resources

Historic filters are stored as template types in the project resources. The configuration software provides a resource manager which can be used to view, edit, import, and export historic filter resources. When modifying an historic filter template, all instances which use it are updated.

Select the command *Historic Filters...* from the Edit menu to open the resource manager dialog. Select a filter type and click the edit button  to modify it. Use the plus button to add new filters and the minus button to delete selected filters. Click the duplicate button  to create a duplicate of the selected filter. Click the import button  to load historic filters from disk. When importing, filter definitions of the same name are overwritten. Click the export button  to store current filters to disk.

10.16 Data Points in L-Vis Projects

When referencing a data point of the data point configuration via an L-Vis data point object, the initial direction is taken over from the referenced data point, with value data points being used for read access by default. If needed, the data flow may be reversed by setting an option in the L-Vis data point object. This is important because the data flow direction determines how a data point will be used by other objects in the tree. For example, a mathematic object will assign variable names (v1, v2...) to all input data points and write the result of the calculation to all output data points. For controls, the direction of the attached data points determine if the control will be selectable (to input new data) or not.

NOTE:	<i>The direction is defined from the point of view of the L-VIS device in question, referring to the network side, not the application side. Therefore, an input data point will receive new data from the network and deliver this data to the L-Vis application, whereas an output data point will write new data from the L-VIS application to the network.</i>
--------------	--

Data points in an L-Vis project fall into one of the following categories:

- **Favorite:** A favorite is a symbolic link to another data point. It can be used to define a virtual data interface which can be connected to real data points during integration. It may be helpful to build reusable templates or collect data from different base folders into one favorite folder.
- **System Register:** System registers cannot be created by the user. They are provided by the device to modify certain internal system parameters, like the time zone, IP address, BACnet device instance number, or others, or read the current value of a system parameter like the current time, CPU load, free memory, and so on. A system register object is either a read-only or a write-only value, and is accessed via a system register data point of the appropriate direction.
- **User Register:** A user register is a place to store data internally, instead of communicating with other devices on a network. Registers are used to internally transfer data between objects, for example when a mathematic object calculates a value which should be displayed on a control, the result of the calculation is stored to a register and the control reads the value from the register for display. A register is therefore always represented either by a data point of type value (r/w) or by two separate data points. One to write a new value to the register and one by which the current value of the register can be read.
- **Local Network Object:** A local network object refers for example to a static or dynamic network variable (CEA-709 models) or a BACnet server object (BACnet models) which exists on the device. Most of these communication objects are either write-only (output NV, BACnet input object) or read-only (input NV, BACnet output object). They are therefore represented by one data point with the appropriate direction. One exception to this rule are BACnet value objects, which may be read and written and which are therefore represented like user registers (either one value data point or two separate read and write points).
- **Remote Network Object:** A remote network object is a reference to some object existing on a remote device on the network. Such a reference is called a *client mapping* on BACnet devices, since it maps the value of the existing data point on the L-VIS device to a communication object on a remote device. Depending on the direction of data flow, the mapping can be either read (poll or COV) or write. There may be more client mappings for the same remote object, for example to a remote BACnet value object, which can be read and written and is therefore represented by two client mappings. On CEA-709 devices, external network objects are called *external NVs* since they are references to NVs located on other devices. A remote network object is usually only constructed from data retrieved

by a network scan and will only work as long as the referenced object on the remote device does not change.

NOTE: *From the data which is retrieved by a network scan, it is possible to create a local network object from the client mapping. The user will decide if a client mapping or a local network object is the better solution, which depends on the network technology and the type of installation. For CEA-709, it is usually better to create local network objects and use a network management tool to define the bindings to other devices in the network, for BACnet it may be more convenient to use client mappings instead.*

- **OPC Client Objects:** An OPC client object is a data point on the LVIS device which represents an existing data point on a remote OPC server, for example a LINX device. Such data points are created by importing a LINX device configuration and providing the IP address of the corresponding device. The resulting data points will be available in a top level data point folder, next to the L-Vis folder. For each imported LINX device, a new top level folder will be created, holding all OPC-accessible data points of the remote device.

NOTE: *OPC client objects do not connect to the network side of the remote device, but to the application side. That is, the LVIS panel will function as a remote application for the LINX device, controlling the LINX data points in the same way as for example a logic program running on the LINX device or the Web UI of the LINX device. A local network output on the LINX, represented by a write data point, will also be represented by a write data point on the LVIS, just as if the LVIS application would be running on the LINX device. In contrast, remote network objects are always in reverse direction, since the data sent out by the remote device over the network will be received by the LVIS device and vice versa.*

- **Modbus Register:** A Modbus register object is similar to a user register, but with data types and sizes which are compatible to Modbus network technology. It can be created on Modbus interfaces which operate in slave mode and will be accessible via the Modbus interface by external Modbus masters. These data points are also called Modbus slave data points.
- **Modbus Remote Register:** A Modbus remote register object maps to a Modbus register on a remote Modbus slave device, which can be accessed via a Modbus interface on L-Vis operating in Master mode. This is the standard type of Modbus data point, since L-Vis will usually act as the Modbus Master, reading out register values from Modbus slave devices like energy meters.

New objects can either be created manually, or from template objects which were automatically created based on information provided by a file or a network scan. The various objects and the way to create them are explained in the following sections.

10.16.1 User Register

A new user register is very easy to create. Select the data point folder *User Registers*, right-click in a free area of the data point list, and select *New Data point...* from the context menu. A new dialog appears, asking for two names, a *Datapoint Name* and a *Register Name*. The first one is used for the name of the data points which will be created. The suffix *_Read* or *_Write* will be appended to this name, to identify the read and the write point. The second is the name of the register itself. You may leave either one or even both names empty, in which case a suitable default will be chosen.

For a description of the available register types which can be selected from a dropdown list, see Section 10.1.8 about user registers in general.

There are check boxes to define the *Direction* of data flow needed for the new register. Usually a local register will be read and written, so either select **Value** to create a single data point for read and write or uncheck the **Value** option and check both options **Read** and **Write** for two separate data points.

The optional check box to create the register as a parameter can be used to avoid the need to set this option in a separate step later on, if this requirement is already known. Note that this option can be changed later at any time.

10.16.2 Local Network Variable (CEA-709)

For CEA-709 models, local network data points refer to local network variables on the L-VIS device. If there are dynamic NVs present on the device, suitable data points will be created automatically. All manually created data points will therefore create additional static network variables on the device.

NOTE:	<i>Adding, removing, or otherwise changing the static network variables of a CEA-709 device may have an impact on the integration of the device in your network. The configuration software will automatically determine if your changes need special attention and warn you if the required modifications cannot be made automatically. Refer to section 3.2.1 for more information on this topic.</i>
--------------	---

To create a new static NV, select the folder named *Local NVs* and use the *New Data point...* command from the context menu. A new dialog will open and request the required information to create the new NV, which is:

- **Datapoint Name:** Enter a name for the data point created for the new NV. If this is left blank, the data point name will be derived from the programmatic name of the network variable.
- **Programmatic Name:** The programmatic name of the network variable on the device. There are certain restrictions to this name, which originate from the underlying network technology. For example, the name may not exceed 15 characters. Also, the name should start with the prefix *nvi* or *nvo*. If this is not the case, a suitable prefix will be added to the name automatically (depending on the selected direction).
- **Resource File:** This dropdown list contains all resource files currently installed on the system. Select the desired resource file, which contains the data type you want to create. The standard network variable types (*SNVT_xxx*) are defined in the resource file called *STANDARD*. If any additional resource files containing user defined type definitions (*UNVT/UCPT*) are installed on the system, they will be available in this drop down box for selection. In case the desired resource catalog is not listed in the drop down box, check the installation of your third party software.
- **Type:** The desired type of the network variable. The available types in this dropdown depend on the resource file selected before.
- **Direction:** Input or output, as seen from the network side.
- **Functional Block:** The functional block in which the NV should be located. L-VIS provides an array of eight functional blocks to organize your NVs and keep them separate from other NVs on the device which are managed by the device firmware.
- **NV Flags:** Flags for the newly created NV. You may leave these untouched. Most of the flags will be handled by the network management tool later on. The most important one which cannot be set by other tools is the polled flag. This is automatically set for newly created NVs as required.

10.16.3 Local Server-Object (BACnet)

On BACnet models, local server objects may be created to communicate with other devices on the network. The basic object types supported are analog, binary, and multi-state, each

available as input, output, or value object. Since a value object may be read and written, two data points will be created for each value server object when the project is designed for firmware before 4.6. For newer projects, a single value data point is created, unless the project settings indicate that this is an older project which is still using read/write data points. See Section 11.2.1 for a description of the relevant option.

Note that once created, a value data point is not automatically split into two separate points when changing the target firmware version from 4.6 or newer to 4.5 or older. Watch out for value data points before using a project designed for 4.6 or newer on an older device. In most cases, it is preferable to update the firmware of the old device instead.

NOTE:

BACnet defines the direction of data flow from the point of view of the application running on the L-VIS device. An analog input object therefore accepts input from the user and transmits this data out to other devices on the network. An analog output object receives data from the network and outputs it to the application (show data on the display). A BACnet input object is therefore represented by an output data point, whereas a BACnet output object is represented by an input data point.

To create a new server object and the required data point(s) to represent it, select the folder named Server Objects and use the context menu to add a new object. A dialog will open, requesting the required information to create the new server object. The following data may be entered:

- **Datapoint Name:** This is the name of the data point representing the BACnet server object. You may leave this empty, in which case the server object name will be used.
- **Object Name:** The name of the server object to be created. This is the name which will be seen by other devices on the network.
- **Object Type:** Select the required object type here. Keep in mind that ‘input’ and ‘output’ here are according to BACnet naming rules for objects, as explained above.
- **Engineering Units:** Optionally, specify the engineering units for your server object. This will be available to other devices on the network as a property of your server object.
- **Description:** As above, optionally specify a description for the server object. In contrast to the data point description, the description given here will be available as a property of your server object for other devices or system integrators to read.
- **Device Type:** Optional property for your object, as above.

Once all the required information has been filled in, the button *Create Server Object* is used to create the new server object and the required data point(s) to represent it. As explained before, you will get either two separate read and write data points or a single value data point, if you create a server object of type ‘value’.

10.16.3.1 Map other Properties than Present_Value

When creating a BACnet server object, the *Present_Value* property is mapped by the created data point. That means writing and reading on the data point reads or writes the *Present_Value*. If other properties shall be accessed, they must be added to the BACnet server object’s data point.

To Add other BACnet Properties

1. Select the BACnet server object for adding properties.

2. Right-click on the data point and select **Add/Remove BACnet properties ...** from the context menu. The dialog appears as shown below.

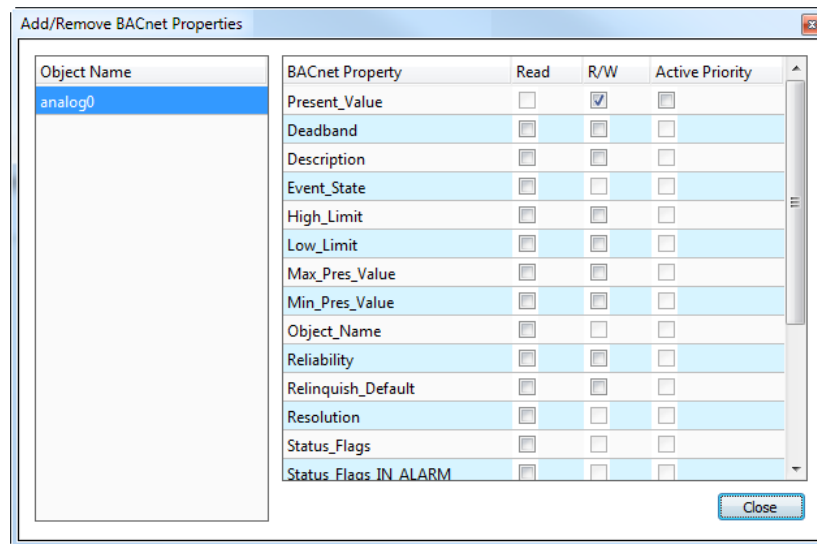


Figure 35: Dialog for adding/removing BACnet properties.

3. Check the additional properties. Checking the **Read** box will add an input data point, checking the **R/W** box will add a value data point.
4. Click **Close**. The selected data point can now be expanded with the plus icon and show its additional properties as sub-data points.

[-] AI1	1	<input type="checkbox"/>	<input type="checkbox"/>	Out	AI1	Analog Input
[-] High_Limit	1.1	<input type="checkbox"/>	<input type="checkbox"/>	Value	AI1	Analog Input
[-] Low_Limit	1.2	<input type="checkbox"/>	<input type="checkbox"/>	Value	AI1	Analog Input

5. To remove properties perform the same steps and uncheck the corresponding checkboxes. Alternatively, select the property (or more) and press the *Delete* key.

10.16.3.2 Read the Active Priority

In BACnet objects that possess a priority array the effective value depends on the used write priority slots. A special data point on the device can be added that allows reading out the active priority of such an object, giving a value between 1 and 16.

To Read the Active Priority of a Local Object

1. Select the BACnet server object for adding properties.
2. Right-click on the data point and select **Add/Remove BACnet properties ...** from the context menu. The dialog for mapping BACnet properties to data points appears.
3. For *the Present_Value* select the additional box Active Priority.

BACnet Property	Read	Write	Active Priority
Present_Value	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

To Read the Active Priority of a Feedback Value

1. Select a write client mapping.
2. Right-click on the data point and select **Create Priority Feedback Point** from the context menu.

3. A new data point is created, which contains a special client mapping to read the active priority out of the remote object.

10.16.3.3 Write with Priority

In BACnet objects containing a priority array, the effective value depends on the values written to the individual priority slots. To access these slots, additional priority output data points can be added for writing to the BACnet object at different priorities. A write priority between 1 and 16 can be specified which each of the output points.

To Create a Priority Write Data Point

1. Select the commandable BACnet server object.
2. Right-click and select **Create Priority Write Point...** from the context menu.
3. A dialog pops up and prompts for the write priority. Note that write priority can also be changed later.
4. The new priority write data point appears below the original BACnet server object data point.

Datapoint Name	No.	OPC	Param	PLC	Direction
AO1	1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	In
AO1_pri4_Write	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Out

10.16.4 Remote Network Object

Since there is a lot of information required to create a mapping for a remote network object, such objects are usually only created from existing templates. The first step therefore is to either scan the network for remote network objects or import a file, to get a list of available template objects from which new mappings may be created.

10.16.4.1 CSV / EDE Import

Use the context menu of the file import folder and select the command *Import File*. Browse to the CSV file you want to import and select it. For BACnet models, standard EDE files may be imported through this method. For CEA-709 models, a similar file format was specified so that tools may be developed which are able to output data in a format the configuration software is able to read. An example CSV file for CEA-709 is installed together with the configuration software.

Once opened, the contents of the file will be parsed and suitable data point templates will be created in the file import folder. These templates are then used in a later step to create client mappings for use on the L-VIS device (see section 10.16.4.3 below).

10.16.4.2 Network Scan

Scanning an existing network is another way to generate template data points. Depending on the type of device and the current operating mode of the configuration software, there are different ways to scan the network:

- **BACnet, no connection to a device:** In this mode, there is currently no way to scan the BACnet network. To execute the scan, a connection to a BACnet device must be established first.
- **BACnet, connected to a device:** The network can be scanned by the L-VIS device, which reports the results back to the configuration software. To start the

scan, use the context menu of the *BACnet Network Scan* folder and select the menu item *Scan BACnet Network*.

- **CEA-709, no connection to a device:** In this mode, network scanning is currently not possible. Connect to a device first.
- **CEA-709, direct connection (not LNS mode):** In this mode, network scan is currently not possible, because the scanner engine on the device can only be operated over a TCP/IP link.
- **CEA-709, connected to a device via TCP/IP:** In this mode, the device is able to execute a network scan and report the results back to the configuration software, similar to the BACnet scan. Since the device scans the physical network, the results can only include information which is stored on the devices in the field, not information which may be present in an LNS database, like a device name, functional block names, or NV display names. To start the scan, use the context menu of the scan folder and select *Scan CEA-709/852 Network*.
- **CEA-709, connected to a device in Plug-In mode:** In this mode, scanning is done on the LNS database. The devices do not need to be physically present. The information gathered from this type of scan will usually include more information, because there is more information about the devices available in the LNS database. However, finding a device through this type of scan does not necessarily mean that the device will actually be available on the physical network, whereas the results from a direct network scan will reflect the network as currently seen by the L-VIS device running the scan. To start the scan, use the context menu of the scan folder and select *Scan LNS Database*.

A new window will open, which is used to conduct the network scan. Depending on the type of scan, the window will contain different controls and options to conduct the operation. A few things to note are:

- **BACnet Scan:** As the first step, the current list of devices should be updated using the button *Refresh Device List*. After a while, the list of available devices on the network will be shown in the device list. Now one or more devices may be selected from this list and the objects available on the selected devices can be scanned by pressing the button *Scan Objects*. This step may be repeated for all required devices. For each device, a new sub-folder will be created and the objects found on the device will be created in this sub-folder.
- **CEA-709 Scan:** Note that the device you are connected to must be commissioned in order to execute the scan. If it does not yet have a valid network address, a suitable address may be assigned in the area called *Connected Scanner Device*. If you want to see a list of all reachable devices, use the button *Discover Devices*, alternatively start scanning for service pin messages by pressing the button *Discover on Service Pin*, then go through all required devices and press the service button, which will add the device to the list of available devices, and stop the process using the *Done* button when finished. Once the list of devices is ready, use the button *Scan Device* to get a list of objects available on the device.
- **LNS Database Scan:** The current network will be selected from the database automatically. Select the desired channel from the list of channels at the right to see a list of devices on that channel. Now either scan all devices on the channel using the button *Scan Channel*, or select individual devices from the list and use *Scan Device* for individual device scan.

When you are done with all the required devices, close the scanner window and return to the data point management window. The scan folder should now have a sub-folder for each of the scanned devices, containing data points representing the communication objects

available on these devices. Now the required mappings may be created from these template data points (see next section).

10.16.4.3 Create Mapping

Browse through the available data points in the import folder and select the ones for which mappings should be created. Multi selection within the same folder is possible. Open the context menu and select *Use on Device*. For all selected points, a suitable remote data point will be created.

NOTE:	<i>The data direction of the points created in the import folder are set from the point of view of the scanned device, since they represent communication objects which are available on the remote device (for an input network variable on a remote device, you will see an input data point in the import folder). Once the point is used to create a remote network data point on the L-VIS device, the direction will change, since an output network variable will be required to send data to the remote devices input network variable.</i>
--------------	---

To simplify the selection and configuration of remote network data points for CEA-709 devices, it is possible to create device configuration templates for the different program-IDs. The device template defines a set of network variables and their configuration, which can be used as a template when a device of the specified type is added to the project. To add new network variables to the device template, select the desired imported data points from the import folder and select the context menu command *Use as Template* to add them to the respective device template, which will be automatically created for each device in the *CEA-709 Templates* folder. On these templates, you can now configure a few settings like the desired data point name, or the type of data point (external or static NV). These settings will be applied automatically when a data point matching the template is added.

When the templates are ready, it is possible to use all data points from all scanned objects which match one of the available templates, and have the default settings applied to the newly created data points. To do this, select the command *Use on device and apply all templates* on the top level scan folder.

NOTE:	<i>A remote network data point refers to another device in the network, which had a certain configuration at the time the scan was executed. If the configuration of the remote device changes later on, the remote network data points may stop working until the network is rescanned or the mapping to the remote device is changed manually to reflect the new configuration. To avoid this situation and to keep a project reusable for different networks or different subsystems within the same network, it is therefore desirable to create local network data points instead when using a CEA-709 device, and use network variable bindings to define the communication between the L-VIS device and other devices. This way, the same project file may be used in different rooms, communicating with different devices and not be specific to a certain room.</i>
--------------	---

11 Project Settings

Using the command *Project Settings* from the *File* menu, a number of advanced options can be controlled which apply to the entire project. The dialog is organized in a number of property pages, which are explained below.

11.1 Basic Settings

On this property page, the basic setup is performed. The areas of the property page are explained in more detail below.

11.1.1 Page Protection

In this area, all options related to access control can be configured. Access control is described in detail in section 15.1. The following data can be set:

- **Setup PIN Codes:** This button opens a new dialog where the 16 available access levels can be configured. For each level, a name and a PIN code can be entered, except for level 0 (the lowest level) which cannot be protected.
- **Next download sets PIN codes:** If this checkbox is selected, the next project download will reset the PIN codes stored on the device with the codes entered in this project. Otherwise the PIN codes of the devices are not modified. PIN codes can be changed on the device during runtime by writing new PIN codes to the PIN code Set system registers.
- **Lock open pages after:** This sets a timeout after which the system resets the current access level back to 0, if no user input was detected (auto-logout).
- **Go to default page after:** This sets a timeout after which the system will switch back to the page marked as the default page, if no user input was detected.

11.1.2 Timeout Values

This section is used to define various time limits. The following timeout values are currently available:

- **Leave edit mode after:** The amount of time after which a control which is currently in edit mode (accepting user input) and is not being used will cancel the input operation, possibly restoring the value which was in effect before the user input was started.
- **Popup menu after:** The amount of time the user has to touch the screen on an empty area to pull up the navigation menu. In projects where navigation is usually accomplished by custom buttons on the screen, this timeout can be set to a high value.

- **Hide popup after:** The amount of idle time after which an open navigation menu or other popup is closed automatically, without changing to a different page.
- **Hide selection after:** The amount of idle time after which the selection frame of the last used control is removed.
- **Signal long press after:** The amount of time a control has to be touched to trigger a long press action. If the control is released within a shorter period, a short press will be triggered.
- **Dim backlight after:** The amount of idle time after which the backlight is dimmed down and the device enters standby mode. This may have effects on other functions, for example a schedule control may return to the weekly overview page and show the current date.
- **Backlight OFF after:** The amount of idle time after which the backlight is completely turned off. Note that this time has a minimum value to protect the CCFL lamp from turning off and on too often, reducing the life time of the lamp. Restarting the lamp after it was turned off typically consumes an equivalent lifetime of 10 minutes in normal operation.

11.1.3 Sound

This section controls the sounds produced when touching the display and entering values. The frequency of a low pitch and a high pitch sound may be entered and the sound output can be turned on and off.

11.1.4 OEM Bitmaps

In this section, bitmaps can be specified to customize the about page (shown during system boot) and the PIN code entry page (shown when a protected page is accessed and the current access level is not sufficient). The bitmaps must match the resolution of the display (for example 320x240). Suitable template bitmaps are installed together with the configuration software. For newer firmware versions, it is recommended to design dedicated pages as part of the project and declare them as *Login Page* or *About Page*. See section 9.4.1 for more information about this feature.

11.1.5 System Strings

Pressing the button *Setup System Strings* opens another dialog, where a number of strings can be translated to match the requirements of the project. These strings are used by the specialized controls on the device to edit schedules, calendars or display alarm lists. A set of translations may be given a name in the *Preset* field, which can then be saved using the *Add* button. The translation is then stored on the system and will be available in the Preset dropdown list for re-use in other projects.

11.2 Extended Settings

On this property page, some less frequently used options are available.

11.2.1 System Flags

The following system flags are currently available:

- **Lock pages in setup menu:** If this option is checked, the pages of the built-in system setup menu will be protected at level 15, otherwise they will be public. If the default setup menu is not used, this option has no effect.

- **Hide setup menu:** This option instructs the device not to add a link to the built-in setup menu, so that the built-in setup is not accessible. The user may provide customized setup pages instead, matching the local language and the design of the other pages. See section 7.3 for details on this topic.
- **Disable touch gestures:** This option causes the touch gestures to flip pages and open the setup menu to be turned off.
- **Ignore data point value state:** This is a compatibility option which will be checked automatically when an older project is loaded. Turn it off to enable the new feature of tracking the state of a data point in addition to its value. For example if an input NV has a receive timeout set and does not receive a value, the point will go OFFLINE and the display will indicate that the value shown is no longer considered valid. Note that using invalid values in a formula will automatically invalidate the result; therefore it is possible to turn state tracking off for individual data points (see the description of the data point object). The checkbox here is a master switch to turn state tracking off for the entire project.
- **Create separate data points for read/write:** This is a compatibility option, which is automatically set for projects created with older configuration software. It makes sure that new data points for bidirectional data transfer are still created as separate read and write points, instead of a single value data point. To upgrade an old project to use value data points instead of read/write pairs, convert the existing data points as needed using the context menu command *Convert to value* and uncheck this option for future data points to be created as value data points.
- **Enable deploy to LROC device:** Used by the L-ROC automation system to determine if this project should be compiled and deployed automatically.
- **Enable all settings:** Can be used in LWEB projects to enable all project settings, even those which are not used by LWEB clients. This can be helpful with a workflow where the primary design is done on an LWEB project and this project is later converted and downloaded to LVIS devices. In LVIS projects, this setting may enable features which are otherwise hidden because they are LVIS specific (not supported by LWEB) and will therefore break compatibility of the LVIS project with LWEB clients, which can be a problem in some use cases.

11.2.2 System Settings

The settings which can be made in this section are:

- **Device Name:** A device name can be given using this field. It will be used as the host name when the device requests an IP address at a DHCP server and it will be used as a server name by the built-in VNC server. The device name will also be available on the device in a system register, so it can be displayed or connected to other data points so that it can be seen on the network.
- **Time Source:** This dropdown list allows to set a specific source for time synchronization. This avoids possible conflicts if there is a time signal coming in from more than one source and the time is not the same. Select the desired source for time synchronization from the list.
- **Update Scheduler Config:** This dropdown list is used to set the behavior of the configuration software when a project was downloaded to the device or uploaded from the device and there are local schedulers configured. Depending on the setting here, the configuration software will either automatically synchronize the scheduler configuration data, ask the user if the configuration should be updated, or not update the configuration, in which case the user may manually update using the commands from the connection menu.

- **Create data points during paste:** Set the behavior of the configuration software when objects are pasted from a different project and not all of the referenced data points can be found in the destination project. Missing data points can either be created automatically as needed (classic behavior), not created at all (will break the reference in the pasted object), or the user can be asked what to do (new default behavior).
- **Keypad Button Size:** Specifies the desired size of the keypad buttons on the display (width x height) in millimeters.

11.2.3 System Colors

This section is used to adjust the general system colors. They will be used on the device by elements which are created dynamically, for example the keypad, and will be used by the configuration software as default colors for new objects when no other defaults are set by the user (using the button *Set as Default*) on the **Common Properties** page.

11.2.4 Unit System

This section is used to set the unit system to use for engineering the project. It influences the unit in which default values, limit values, COV increment, or other data point properties specified in data point units, are displayed and entered in the configuration software. The effective unit is also reflected in the LCD preview, when numeric controls display the unit of the connected data point.

If the option **Set unit system** is set, the device will take over the configured unit system setting whenever the project is downloaded to the device. Otherwise, the unit system selected for project engineering is independent of the unit system selected on a specific device for operation. The setting can be changed on the Web interface of the device, via a system register, or on the device commands page of the built-in setup menu.

11.2.5 Time Zone and DST

At the bottom of the page, a checkbox can be set to enable automatic control of the UTC offset, based on a region selected from the dropdown list. If enabled, the device will automatically control the UTC offset register and also automatically switch between standard time and daylight saving time, if applicable for the selected region.

11.3 Localization

This page contains options and settings related to project localization. It is used to define the list of project languages and set the desired locale for each language, if supported by the selected device model.

11.3.1 Project Languages

The *Project Languages* area contains a few buttons to add and remove languages and change their order. Below the buttons is the list of currently available languages. Selecting an entry from the list switches the project editor to the corresponding language. Any modifications to the locale settings (see below) or text and font settings of individual objects will apply to the selected language.

The buttons from left to right:

- **Add** a new language, based on the currently selected one. All text and font settings as well as the locale settings for the new language will be initialized from the currently selected one. Based on this copy, the user may adjust the properties of the new language and start translation of text and font settings.

- **Move** the currently selected language **up** by one position. The position of a language in the list is important for run time language selection, because languages are identified by numbers from **0** up to the total number of available languages, with language **0** being the first language in the list.
- **Move** the currently selected language **down** by one position (see above).
- **Delete** the selected language.

11.3.2 Locale Settings

For each language, the desired locale can be configured in the *Locale Settings* area:

- **Language:** Select the desired language of the project from a list of more than 190 available languages.
- **Region:** In case the selected language is spoken in different regions of the world, refine the selection by picking the appropriate region. This will change details like the used decimal separator, grouping character, or currency of the locale to match the selected region. It will also influence the selection of available scripts and calendar systems (see below).
- **Script:** In some regions, more than one script is commonly used. For example, Hong Kong may use either simplified or traditional Chinese. In most cases, however, the script will be automatically defined by the selected language and region, without any options to choose from.
- **Calendar:** The calendar system to use. While most regions exclusively use the Gregorian calendar, there are some regions in which multiple different calendar systems are in use. For these regions, the calendar selector becomes available and lists the commonly used calendar systems for the selected region, for example Gregorian, Hebrew, and Islamic for region Israel. The selection of the calendar system has a big impact on how dates are displayed in a locale aware date control.
- **Numbers:** The number system to use. The setting **Default** selects the number system commonly used for the selected combination of language and region and is the preferred setting.

Whenever the current language is changed during run time by writing a different language number to the **Language** system register, the corresponding locale will be put into effect automatically. In addition to the selected language number, the currently active locale is also available in a system register, called **Locale ID**. However, there is no need to manually set the locale when project languages are used. For single language projects that still want to change the locale, see Section 0 regarding manual locale control.

11.3.3 Options

There is currently one option, which was moved from the **Extended Settings** page to the **Localization** page:

- **Support complex writing systems:** This option enables support for complex writing systems such as Arabic, Devanagari, Bengali, Thai, Hebrew and many others belonging to the family of Abjad or Abugida scripts. These scripts are often bidirectional and require complex text shaping and layout engines to display correctly. It is available as an option for LVIS-3MEx devices and always turned on for LWEB-80x projects. Note that this option will also improve text layout in western scripts by using ligatures and other advanced layout features, as far as provided by the font. If an existing project is converted to the new layout

engine by turning this option on, the project must be re-checked for possible problems due to changes in text layout.

NOTE: *This option causes **significant overhead** during text layout. Especially on pages with a lot of individual short pieces of text, like a page with a large number of text or number controls, it will cause noticeable delays when switching to a different page, or updating a complex control with a large amount of text. In projects which do not require support for complex writing systems, consider to turn this option off for maximum performance.*

11.3.4 Manual Locale Control

The current locale can be changed manually during run time by writing a so-called locale ID string to the **Locale ID** system register. The following sections contain detailed information regarding manual control of the current locale.

NOTE: *In general, it should **not** be necessary to set the locale ID string directly, since a different locale will only be required when changing the project language. However, switching the active language will automatically set the locale configured for that language, so there is no need to manually change the locale, except in a single language project with a requirement for different locale settings within one language.*

When the locale ID is changed manually during run time in a single language project, the following algorithm applies:

- When a new project is downloaded, the device checks if the locale setting in the new project is different from the old project. If yes, the settings from the new project are taken over (activated on the device). Otherwise, the current device locale remains unchanged, even if it is different from the project settings.

This ensures that changing the locale settings of the project in an otherwise static system (no run time changes) will take effect after download, and at the same time keep any run time selection of a different locale persistent across project downloads, as long as the locale setting of the project remains untouched.

NOTE: *It is **not recommended** to mix run time language selection and manual control of the locale ID. The results may be unexpected. The device does **not** store a run time modified locale ID for each of the available languages. Selection of a language will always install the locale defined for that language in the project settings, overriding any manual modifications.*

11.3.4.1 Locale ID Format

In case the locale for the selected language is going to be changed during run time, it is necessary to understand the format of the locale specifier string in order to construct valid input for the system register. A locale ID string consists of one or more pieces of ordered information:

- **Language Code:** The first part of any locale string is the language identifier. Languages are specified using a two-letter lowercase code for a particular language. For example, Spanish is *es*, English is *en* and French is *fr*. The two-letter language code follows the ISO-639 standard.
- **Script Code:** The optional four-letter script code follows the language code. If specified, it should be a valid script code as listed on the Unicode ISO-15924 Registry, for example *Latn* for Latin, *Hans* for simplified Chinese, or *Jpan* for Japanese.
- **Country Code (Region):** There are often different language conventions within the same language. For example, Spanish is spoken in many countries in Central and South America but the currencies are different in each country. To allow for these differences among specific geographical, political, or cultural regions,

locales are specified by two-letter, uppercase codes. For example, *ES* represents Spain and *MX* represents Mexico. The two letter country code uses the ISO-3166 standard.

- **Keywords:** The final element of a locale is an optional list of keywords together with their values. Keywords must be unique. Their order is not significant. Unknown keywords are ignored. The following keywords may be useful to override the default settings of the selected locale:
 - **calendar:** A calendar specifier such as *gregorian*, *islamic*, *chinese*, *hebrew*, *japanese*, or *buddhist*. See Section 11.3.4.2 below for a list of valid calendar names.
 - **currency:** A standard three-letter currency code, such as *USD* or *JPY*. Can be used to override the default currency of the selected locale and is used by the currency format of number controls.
 - **numbers:** A numbering system specifier such as for example *latn*, *arab*, *deva*, *hans*, or *thai*. The numbering system defines the set of digits used for decimal formatting, such as *latn* for western (ASCII) digits, or *thai* for Thai digits. The numbering system may also define complex algorithms for number formatting, such as *hansfin* for simplified Chinese numerals using financial ideographs. A list of valid numbering system names is given in Section 14.3.3.

NOTE:	<i>The numbers keyword can be used to globally override the default numbering system specified by the selected locale. To use specific numbering systems for individual controls, override the numbering system directly in the controls properties. The global setting only affects controls which inherit their numbering system from the locale (default setting).</i>
--------------	--

The basic syntax to combine the individual pieces of information and form a valid locale specifier, requires that *language*, *script*, and *country* codes are separated by underscore. If keywords are specified, an @ sign must be used to separate the base locale from the keyword section. Keywords are given using a *key=value* syntax, where multiple keywords are separated by semicolons:

language_script_country@keyword1=value1;keyword2=value2....

Anything except the language code is optional, so it is valid to just select a language by writing *en* (English), *de* (German), *jp* (Japanese), or any other ISO-639 language code to the system register. However, in most cases a language and a country code should be given to fully specify a locale, since the country code defines how numbers are formatted (decimal separator, grouping, and currency).

Example: Japanese for Japan with native calendar and numbering system:

ja_JP@calendar=japanese;numbers=jpan

Example: Traditional Chinese for Hong Kong with Traditional Chinese calendar:

zh_Hant_HK@calendar=chinese

11.3.4.2 Calendar Names

This section lists the valid calendar names for use with the **calendar** keyword:

- **buddhist:** Thai Buddhist calendar
- **chinese:** Traditional Chinese calendar
- **coptic:** Coptic calendar
- **dangi:** Traditional Korean calendar

- **ethioaa:** Ethiopic calendar, Amete Alem (epoch approx. 5493 B.C.E)
- **ethiopic:** Ethiopic calendar, Amete Mihret (epoch approx, 8 C.E.)
- **gregorian:** Gregorian calendar
- **hebrew:** Traditional Hebrew calendar
- **indian:** Indian calendar
- **islamic:** Islamic calendar
- **iso8601:** ISO calendar (Gregorian calendar using ISO 8601 calendar week rules)
- **japanese:** Japanese Imperial calendar
- **persian:** Persian calendar
- **roc:** Republic of China calendar

The following calendar systems are currently not supported by LWEB-802/803, but can be used on LVIS-3ME devices:

- **islamic-umalqura:** Islamic calendar, Umm al-Qura
- **islamic-tbla:** Islamic calendar, tabular (intercalary years [2,5,7,10,13,16,18,21,24,26,29] - astronomical epoch)
- **islamic-civil:** Islamic calendar, tabular (intercalary years [2,5,7,10,13,16,18,21,24,26,29] - civil epoch)
- **islamic-rgsa:** Islamic calendar, Saudi Arabia sighting

11.3.4.3 Numbering Systems

This section lists the valid numbering systems for use with the **numbers** keyword. The following systems are supported on every client:

- **latn:** Latin digits
- **arab:** Arabic-Indic digits
- **arabext:** Extended Arabic-Indic digits
- **bali:** Balinese digits
- **beng:** Bengali digits
- **deva:** Devanagari digits
- **gujr:** Gujarati digits
- **guru:** Gurmukhi digits
- **hanidec:** Positional decimal system using Chinese number ideographs
- **khmr:** Khmer digits
- **knda:** Kannada digits
- **laoo:** Lao digits
- **mlym:** Malayalam digits
- **mong:** Mongolian digits
- **mymr:** Myanmar digits
- **tamldec:** Modern Tamil decimal digits
- **telu:** Telugu digits
- **thai:** Thai digits
- **tibt:** Tibetan digits

The following numbering systems are currently not supported by LWEB-802/803, but can be used on LVIS-3ME devices. These are mostly algorithmic systems, or rarely used decimal systems:

- **native:** Native digits (defined by the selected locale)
- **cyril:** Cyrillic numerals — algorithmic
- **ethi:** Ethiopic numerals — algorithmic
- **geor:** Georgian numerals — algorithmic
- **grek:** Greek upper case numerals — algorithmic
- **greklow:** Greek lower case numerals — algorithmic
- **hans:** Simplified Chinese numerals — algorithmic
- **hant:** Traditional Chinese numerals — algorithmic
- **hebr:** Hebrew numerals — algorithmic
- **java:** Javanese digits

- **jpan:** Japanese numerals — algorithmic
- **osma:** Osmanya digits
- **roman:** Roman upper case numerals — algorithmic
- **romanlow:** Roman lowercase numerals — algorithmic
- **sund:** Sundanese digits
- **taml:** Tamil numerals — algorithmic

11.4 Display

This page defines the display properties of the device and project specific properties of the configuration software regarding the LCD preview.

11.4.1 LCD Backlight

This slider is used to set the brightness of the LCD backlight when the unit is active. Good settings are between normal and bright, where the bright setting consumes more power. The life time of the backlight is not significantly affected by the brightness setting.

11.4.2 Display Properties

For L-VIS devices, which have fixed display hardware, this area shows the properties of the display for the selected device model. Devices which can support configurable resolution and/or color depth allow the user to change these settings here. This is currently supported for L-WEB models. Since the L-WEB is actually a piece of software running on a PC, it can support any resolution, although the color depth is currently still fixed at VGA color.

11.4.3 Editor Properties

This section contains some project specific settings related to the graphical editing in the LCD preview.

- **Grid Size:** Defines the spacing of grid points, used when the snap to grid function is enabled. See also Section 8.3.7 about snap to grid.
- **Grid Offset X:** Defines a horizontal offset for the grid.
- **Grid Offset Y:** Defines a vertical offset for the grid.

11.5 Connectivity

This page collects settings related to the communication between the device and a PC.

11.5.1 Remote Display Access

The option **Enable VNC Server** may be checked to enable a built-in VNC server on the device, if supported by the firmware (version 3.0.0 or newer). To use it, a TCP/IP connection to the device must be possible, since VNC is a TCP based protocol. The remaining options are used to set the TCP port on which the VNC server should listen for incoming connections, the maximum number of simultaneous connections that the server should accept and an optional password to protect access to the device.

A device with a firmware supporting VNC which has no project loaded will automatically enable a VNC server on port 5900 and accept one connection without password. This can be used to simplify device installation. If a fresh project is started, the VNC option is turned off by default.

To connect to the VNC server, install one of the available VNC clients on your PC or other device (VNC clients are available for PDA and even some cell phones) and connect to the device. You can then remotely access the LCD display of your device and use the mouse to operate the touch screen.

If the connection fails, the most common reasons are:

- The maximum number of connections is reached.
- There are other clients connected and the new connection requests exclusive access. Check the settings of your client and disable exclusive access in this case.
- There is one other client already connected and holds an exclusive connection.

If more than one user is connected to the same device, everyone will see the same display, just like VNC used on a PC.

NOTE:	<i>LVIS-3E100 and LVIS-ME200 devices use a VGA color map (256 colors). For these devices, it is therefore most efficient to transfer the color map to the client and then only transmit the color index for each pixel (one byte per pixel). If the VNC client supports this mode (color map mode), the VNC server on the L-VIS device will tell the client that this is the preferred display mode to optimize speed and bandwidth usage while keeping a fully accurate rendering of the project colors. However, many clients only support true color modes, where either two or more bytes per pixel need to be transferred or the color resolution will be dramatically reduced. A client which supports color maps is RealVNC.</i>
--------------	---

11.5.2 Remote Data Access (FTP)

To access trend and event logs created by the respective controls, a dedicated FTP access is available. For firmware versions 4.0 and older, the user name and password specified here can be used to log in using an FTP client and download the required data files. Starting with firmware version 4.0, the operator account is used for this purpose. The operator password can be set on the Web-UI of the device.

11.5.3 Data Point Update

This area defines how network input data points are updated with new values. There are two basic modes available, which are controlled by the option **Use dynamic polling**:

- **Enabled:** The device will distinguish between active and inactive data points. An active data point is one which is either currently displayed by a control on the LCD screen, displayed on the web interface, or a data point which is connected to a trigger, action, trend, or formula and thus must be monitored all the time.
- **Disabled:** All data points are treated the same, no matter if and where their values are currently used. Active polling continues for all data points as specified in the properties of each point.

With dynamic polling enabled, the device will only fetch values for currently active data points. Monitoring will still be done at the rate specified for each individual data point, but all inactive data points will stop any polling or subscription activity in order to reduce the network load as much as possible.

To avoid stale data in data points which are not subscribed for extended periods of time, the option **Enable background polling** can be used. The device will go through all otherwise inactive data points and poll the current value at the specified poll rate. This ensures that in cases where a data point becomes active after a long period of inactivity, it already has a reasonable value which is not several days or even weeks old. This can especially be important when the desired data cannot be retrieved at the moment it is requested, due to a communication problem with the server.

NOTE: *Depending on the underlying network technology, inactive data points may still receive event driven updates from other nodes, so dynamic polling has no effect on bound input network variables on a CEA-709 network, for example.*

11.5.4 OPC-XML-DA

For input data points which are using OPC-XML/DA to retrieve new data from the server, there are two modes of operation available:

- **Polling:** Data is requested at a regular interval, which can be specified in the input field called **OPC Polycle**.
- **Delayed Response:** An advanced algorithm to optimize latency and minimize network traffic is applied. For this mode, a **Hold Time** and a **Wait Time** have to be specified. The hold time specifies how long the server may hold back the response and wait for additional value changes. This limits the maximum number of responses, even if there are more value changes. The wait time comes into effect when there is no value change to report during the hold time. It defines how long the server should keep the request and send a delayed response if a value change is detected. If there is no value change even during the wait time, the request times out and an empty response is sent.

11.5.5 Auto-Convert to LWEB

This area is used to set up automatic generation of an LWEB project matching the current L-Vis project. It can optionally be downloaded to the device together with the L-Vis project, where it can be used by LWEB-800/802 clients to display the same project (as currently running on the L-Vis device) also on the PC. The following settings are available:

- **Automatically save LWEB project:** This option must be enabled to make use of any of the following options. When enabled, instructs the configuration software to build and save an LWEB version of the project, whenever the L-Vis version is saved. To do this automatically, the user needs to specify host name and port number of the device which will be used by LWEB to fetch data point values:
 - **Host:** In most cases, you can leave this at the default of *localhost*. This will build an LWEB project which fetches the data points from the same device from which the LWEB project was loaded, which is usually what you want. Only in cases where the generated project will be loaded from a central server instead of the L-Vis device, enter the name or address of the L-Vis device which will run the original L-Vis project.
 - **Port:** The port number for OPC access. Default is port 80.
- **Download together with L-Vis project:** Checking this option will instruct the configuration software to download the automatically generated LWEB project together with the L-Vis project, whenever a project download is executed. This will make sure that the LWEB project on the device always matches the currently running L-Vis project, which is important for the LWEB project to work.
- **Use .lweb2 file format:** This option will use the new .lweb2 file format instead of the traditional .lweb format when building the LWEB project. You may turn this option on if your LWEB-800 client already supports .lweb2 or if you are testing the new LWEB-802 client software, which can only read .lweb2 files.

11.6 Modbus

This page is used to configure the desired mode of operation (Master or Slave) for the available Modbus interfaces, as well as the desired Modbus address ranges for slave mode. There are two sections:

- **Modbus TCP:** Configures operation of the Modbus TCP interface.
- **Modbus RTU:** Configures operation of the Modbus RTU interface, if available.

In each section, the following settings can be made:

- **Interface Mode:** Sets the desired mode of operation for the given interface. In almost all cases, the desired mode will be **MASTER**.
- **Slave Address Ranges:** These input fields become available when slave mode is selected. They define a valid address range for each type of Modbus register. Modbus slave data points created on the respective interface need to use addresses within the ranges specified here.

See also Section 10.6.6 about special properties of Modbus data points.

11.7 Mail Account

This page allows configuring the SMTP mail account used to deliver E-Mails. To use the mail feature, the following conditions must be met:

- The device must be connected to a TCP/IP network and must be able to communicate with the desired SMTP server. In many cases, this server will be outside of the current subnet. Therefore it is important that the gateway address and the network mask settings on the L-VIS are correct.
- At least one DNS server address must be set on the L-VIS device if the SMTP server is given as a DNS name (which is usually the case).
- In the mail account configuration dialog, the enable checkbox must be set and a name or IP address of an SMTP server must be entered.
- It is desirable to use some form of time synchronization on the L-VIS device, in order to provide correct time stamps when building and delivering mails. Some mail clients tend to sort incoming mails by origin date (not date of reception), which could cause confusion if the date and time of the device are wrong.

If the SMTP server requires authentication, set the option **Enable user authentication** and enter a user name and password. A secure connection via STARTTLS is automatically used if supported by the server. To use the older, deprecated SMTPS standard (plain SMTP over SSL/TLS, usually on port 465), check the option **Use SSL secure connection**. Standard ISP mail service often authenticates the user just by IP address, since the address was assigned by the ISP and therefore identifies the user as being a customer of the ISP.

NOTE: *While communication between mail servers usually happens on port 25, mail clients generally don't use this port anymore, since most ISPs are now blocking outgoing connections on port 25 from end customers as an anti-spam measurement. Instead, the submission port 587 (as defined by RFC 6409), is used to submit new mails for delivery.*

Additional configuration data to fill out includes the sender address which is used in all mails originating from this device, an optional reply to address, the maximum number of delivery attempts per mail, and the time between delivery attempts.

If the test receiver address is filled out, the system command to send a test mail may be used to perform a basic check of the mail system.

NOTE: *All failed mail delivery attempts are logged to the system log. Upload the system log through the configuration software and check for errors if a mail delivery attempt fails.*

11.8 Info

On this page, some additional information related to the project can be entered. Currently available is a field for the name of the project designer, some kind of job reference number or purchase order, and a large area to place notes such as changes between revisions, special instructions for installing this project, or general project documentation. This information is not used by the device in any way, but it will be stored together with all other project data on the device and remain available even after uploading a project from the device.

Also available on this page are two fields which display information as soon as the project was saved for the first time using a recent version of the configuration software (4.2.0 or newer). Currently shown are the local timestamp when the project was saved for the last time, as well as the version of the configuration software which created the project file.

12 Standard Procedures (CEA-709)

12.1 Device Integration

How a new device is integrated into your network depends on the network management tool in use and the way the configuration software is run. The available choices are detailed in the following sections.

12.1.1 New Device

Integration of a new device normally starts with an empty device (no project loaded). Integration of a pre-programmed device is explained further down.

NOTE:	<i>To follow the procedures in this section, make sure there is no project loaded in your device. This can be done by setting the CLEAR jumper, starting the device, and removing the jumper again. Also, if the device was previously used in another network, it is a good idea to also reset the network configuration data by pressing and holding the service button while the device boots up. This has to be done in an extra step, when the CLEAR jumper is not set.</i>
--------------	--

The integration procedure may vary depending on the type of network variables used in the project. The following sections detail the three possible variants.

12.1.1.1 Dynamic NVs

In case your project uses dynamic NVs only, there must be a network database system which provides support for dynamic NVs. The configuration software may be run in plug-in mode as well as in standalone mode, since no complicated database updates are required in this scenario. The followings steps will be required to integrate a new device.

- Create a new device in your database using the provided device template for the L-VIS device you are using (FT-10 or IP-10L).
- Create all dynamic network variables on the device which are required for your project.
- Commission the device, so that the dynamic NVs are created and the configuration software can communicate with the device.
- Run the configuration software either as a plug-in on the newly created device or standalone using a TCP/IP or CEA-709 connection. If everything worked OK, the configuration software will be able to contact the device and determine model number and firmware. Suitable data points will be automatically created for all

dynamic NVs present on the device. If you have communication problems, an error message will be displayed and the procedure cannot continue until communication is possible.

- Load the project which was already prepared or design a new project. If a prepared project is loaded, the contained dynamic NV data points will be synchronized with the existing data points, based on the NV name, direction and type. If there are any data points which cannot be assigned automatically, a dialog for manual assignment will be displayed.
- Now download the project to the device. The device will restart and activate the new project. It is now ready to use.

12.1.1.2 Static NVs

Projects using static NVs have the advantage that they do not require the network management to support dynamic NVs. Also, it is possible for such projects to run in a special mode, where the device is compatible to the older CEA-709.1-A network management commands, compared to the extended network management defined in the CEA-709.1-B standard.

To integrate a device using static NVs, the configuration software may be run either as a plug-in or standalone. The procedure when run as a plug-in is fairly automatic:

- Create a new device in your database using the provided device template for the L-VIS device you are using (FT-10 or IP-10L).
- Commission the device, so that the configuration software can communicate with it.
- Run the configuration software as a plug-in on the newly created device and load the prepared project or design a new project.
- Download the project to the device. A warning will appear, notifying the user of the required database update. The update and the selection of a new model number are done automatically.
- After the device restarted and the database was updated, the device is re-commissioned and ready to use. Keep in mind that during the download, a new model number was selected and the project should be saved **after** the download, so that the model number in the file on the PC matches the one in the device.

If the network management does not support plug-ins or the project is downloaded via TCP/IP to reduce the download time, the device is usually first programmed standalone and then integrated using an XIF file:

- Connect to the device using TCP/IP or CEA709. The configuration software will determine the device model and firmware.
- Load or design the project, similar to the above procedures.
- Open the **Network Settings...** dialog and set the *Program-ID Model Number* to the desired value. This will be the last byte of the resulting program ID of the device. Make sure there are no other devices in the same network using the same model number but a different set of static network variables. Also, zero is reserved for a device without static NVs, so the valid range is 1 to 254.
- If required, activate the *CEA-709.1-A restricted mode*.

- Download the project to the device. A warning message will appear, notifying the user that this operation will now change the static interface of the device and that the configuration software will not be able to automatically upgrade an existing device in a network database, since it is run in standalone mode. Since the device will be integrated for the first time, this is OK.
- The device can now be integrated into the network. This is usually done using an XIF file, or, if available, an ‘upload from device’ method, in which the network management reads the static interface directly from the device and thus requires no XIF file. If an XIF file is required, generate one using the **Export XIF...** function from the *Tools* menu and use it to integrate and commission your device.
- If the network management tool does support plug-ins and the newly created device should be configurable in plug-in mode in the future, re-run the plug-in registration for the L-VIS configuration software. The new device will be identified as an L-VIS device and the configuration software will register itself as a configuration plug-in automatically.

12.1.1.3 Static and Dynamic NVs

This is the most complicated setup, which is not available under all circumstances. Projects using static and dynamic NVs at the same time can only be integrated and maintained when the configuration software is run in **plug-in mode**. Due to limitations of the network database system, the presence of dynamic NVs on the device will prevent manual device integration if there are static NVs to be created, even if you don’t actively use the dynamic NV points.

NOTE:

*If possible, try to decide on one method and use either only dynamic or only static NVs, since this greatly simplifies the integration and upgrade process. If necessary, a mixed operation is possible, but due to the required operations on the underlying database is only suggested for advanced users. If used, you are advised to **have a recent backup of your database ready**.*

The integration procedure for such a project is a combination of the procedures used for a static NV project and a dynamic NV project in plug-in mode. The dynamic NVs must be created on the device in a first step, the data points in a loaded project are synchronized with the available dynamic NVs on the device and the static NVs defined in the project will be available on the device after a database update.

The difference is that during the upgrade of the static interface all dynamic NVs will be temporarily removed from the device and the database, so that the upgrade can be executed. The removed NVs will later be recreated automatically, but they may get a different NV index assigned, so it is necessary to update the drawing of your device. Existing bindings will be saved and restored as well. If something goes wrong, restore the database from the backup or read the section about device recovery.

12.1.2 Pre-Programmed Device

In some cases, it may be necessary to integrate an already programmed device into a network. This can only be done with static NV projects, since dynamic NVs are stored in the network database and not the L-VIS project and are therefore not transportable from one network to another.

The procedure is basically the same as integrating a new device with static NVs using the configuration software in standalone mode. The only difference is that the first few steps were already executed by somebody and the device already contains the final project.

When pre-programming a device in this way, an XIF file should always be generated as well after the device was programmed, and this XIF file should be delivered together with

the device. If this was not the case and the network management tool needs an XIF file, it may be created after the device was programmed using the following procedure:

- Run the configuration software in standalone mode and connect to the pre-programmed device for which you need an XIF file.
- Press the **Upload Configuration** button from the tool bar to upload the project which is currently stored in the device.
- Use the **Export XIF...** command from the *Tools* menu to output the required XIF file.

Now integrate the device like any other CEA-709 node, using the XIF file.

If the network management tool supports plug-ins and the new device should be configurable in plug-in mode, re-run the plug-in registration for the L-VIS configuration software, so that it registers itself on the new device.

12.2 Device Replacement

This section describes the steps necessary to replace an already integrated device with a new device using the same project as the old device. Depending on the situation, this may not be a trivial task. The procedures for the possible situations are detailed in the following sections.

NOTE:	<i>The following procedures assume that the L-VIS project file is available to the PC which runs the configuration software. If this is not the case, the project must first be uploaded from the old device and saved to a file.</i>
--------------	---

12.2.1 Dynamic NV Projects

Devices using dynamic NVs only are relatively easy to replace, since the device may physically be replaced right away and the configuration software may then be run in either mode. The recommended procedure to replace a device using dynamic NVs is as follows:

- Physically replace the device in your network, the new device still being empty, that is, no project is loaded yet.
- Execute the standard replacement procedure of your network management tool and use the already existing device template which was installed by the configuration software. This is the same template as used to originally integrate the device, since the device does not change its static interface in such a setup.
- After the replacement, make sure that the device is commissioned, so that the dynamic NVs are created on the device.
- Run the configuration software either in plug-in mode or in standalone mode, load the project from the file and program it into the device.
- The device restarts and is ready to use.

12.2.2 Static NV Projects

A device with a project using static NVs only is easy to replace if it is possible to connect the new device to an IP network for programming purposes. Here is the recommended procedure:

- Connect to the new device in standalone mode, using a TCP/IP connection (requires to set up the IP settings for the new device).
- Make sure the device is empty and remove any existing project if required. This can be done through the configuration software using the command **Remove Configuration...** from the *File* menu.
- Load the project from the file and download it to the device. A warning will appear, stating that the static interface will change and that an already integrated device cannot be upgraded. This is OK, since we are changing the interface so that it will match the already integrated device.
- Once the device restarted, it is ready to use. Connect it at its final location (replacing the old device) and use the standard procedure of your network management tool to replace the device. If an XIF file is required for this operation, you may use the one which was originally used to integrate the device, or create a new one from within the configuration software.

In case it is not possible to program the new device using a standalone connection, the procedure becomes more complicated, because the device needs to be commissioned in the network to be programmed, but the device currently in the database does not match the empty device and cannot be used for this task. The following procedure is recommended in this case:

- Create a new database or network project in your network management tool, adding the PC's network interface and a new L-VIS device and connect them using a suitable channel.
- Connect the new device physically to the network interface of your PC, or otherwise make sure that the PC can communicate with the device in the new network project.
- Commission the new device.
- Run the configuration software on the empty device and program the device with the required project, as explained in the section about integrating a new device using static NVs and plug-in mode.
- The device will be programmed, the database will be updated and the device will be commissioned in your small network project.
- Decommission the device in your helper network, close it, and open the final target network, in which the device should be replaced.
- Connect the device at its original location in the network and execute the standard procedure of your network management tool to replace a device.

NOTE:

*In both cases, make sure that the stored project which is programmed into the replacement device has the correct model number set. This can be checked in the **Network Settings...** dialog. The number must match the last byte of the program ID of the device to be replaced. If this is not the case and the model number is not adjusted before the download, the replacement of the device may fail with an error stating that the new program interface does not match the previous program interface.*

12.2.3 Mixed Projects (static and dynamic)

These projects are the hardest to replace, since it takes a two-step process to complete the replacement. First, the new device needs to be loaded with the project to match the static interface of the original device, then it may be commissioned in the network to get the

dynamic NVs created and then the project needs to be loaded again, so that the dynamic NVs which were not available the first time can be used. The recommended procedure is a combination of the above two procedures.

- Follow the procedure to replace a device with static NVs to prepare the new device with the original project. When the project is loaded from the file, a dialog will appear and list all dynamic data points in the project, stating that they could not be found on the device.
- Finish the dialog by removing all data points listed. Since the device does not yet contain any dynamic NVs, these data points cannot be used for the moment.
- Download the modified project into the device, but **do not** save this project over your original project. Remember that it has all dynamic NV data points stripped. You may save the project to a different file, but this is not necessary.
- Now take the device and execute the procedure for replacing a device using dynamic NVs, that is, physically replace the device and continue with the procedure. If an XIF file or a device template is required for the replacement, use the same XIF which was used to initially integrate the device or use the same device template as the device to be replaced. Since the device already has a project loaded which matches the final static interface, it will be possible to replace the device without changing the static interface in the database. Commissioning the device will create all the dynamic NVs stored in the database.
- The original project must now be loaded a second time, but this time the dynamic NV data points can be assigned to the existing NVs and the project should load without errors.

12.3 Configuration Change

This section describes the necessary procedures to change the configuration of an already integrated device. It depends on the existing project and the kind of changes done to the project, how the update may be performed.

NOTE:	<i>For all configuration changes it is strongly recommended to first execute the download procedure and then store the new project on the PC (usually overwriting the last stored project), since this guarantees that the model number stored in the file on the PC matches the model number which was automatically selected during the upgrade procedure and that in cases where the upgrade must be aborted, the device may easily be restored to the last working project, since it is still intact on the PC and not already overwritten by the new project.</i>
--------------	---

12.3.1 Same NV Configuration

This is the easiest of all configuration changes. As long as there are no changes to either the static or dynamic NVs of the device, there are no special requirements and no special procedures to follow. The new project may be downloaded to the device through any of the available methods, where a TCP/IP connection is usually preferred.

The device will restart after the download and will be ready to operate without any further steps.

12.3.2 Dynamic NV Change

If the project change includes the addition or removal of dynamic NVs **without any changes to static NVs**, the following procedure is recommended:

- Execute all required changes to the dynamic NV interface using your network management software. This includes adding new NVs and removing old NVs as required.
- Once the required changes were made, start the configuration software or, if it is already running, use the **Update Data Points** button from the tool bar to read the new set of available dynamic NVs from the device. The configuration software may run in either mode for this type of configuration change.
- Load the project and modify it as required. If the project still uses some of the dynamic NVs which were removed in the first step, these will be reported and may be replaced by other data points or may be deleted all together. Do not replace them with static NVs, since this would mean a change of the static interface, which implies a different procedure.
- Download the modified project (TCP/IP is preferred due to its speed). The device will restart and be ready to use.

12.3.3 Static NV Change

For project changes which result in a change of the static NV configuration, certain restrictions apply regarding the way such changes may be performed. There are two possible ways to perform such a change. The automatic and recommended way is to run the configuration software in plug-in mode, as detailed below.

- If there are any changes required for the dynamic NVs on the device, perform them as the first step using your network management tool.
- Run the configuration software **in plug-in mode** on the device which is to be changed. This will automatically update the dynamic data points according to the changes which were made to the dynamic NVs on the device.
- Load the current project from a file or from the device and modify it as required, adding or removing static NV data points. During the load, the data points used in the project are updated according to the new set of dynamic NV points. If dynamic NVs were removed or renamed, the affected data points need to be manually assigned to other NVs or may be deleted from the project.
- Download the new project to the device. A warning appears, stating that this will change the static interface of the device and that the database needs to be updated to reflect these changes.
- The configuration software will execute all required steps automatically, temporarily removing all dynamic NVs and all bindings to guarantee a correct update of the device. Once the device restarted and the database was updated, the NVs and the bindings are restored.
- The device is then commissioned and ready to operate.

WARNING:

*It is dangerous to change the static interface while using dynamic NVs, due to the fact that all dynamic NVs must be removed in order to upgrade the static interface. If something goes wrong during this process, all dynamic NVs on the device may be lost and must be recovered using another procedure. You are advised to **have a recent backup of the database** ready, in case the recover mechanism built into the configuration software fails as well.*

For network management systems which do not support plug-ins, the new project must be loaded in standalone mode and a new XIF file must be generated after the download completed. The network management tool itself must then provide a way to upgrade the

static interface of the already integrated device and take over the existing bindings. The recommended procedure is as follows:

- Run the configuration software in standalone mode and connect to the device, preferably using a TCP/IP connection.
- Load the current project from a file or from the device and modify it as required, adding or removing static NV data points.
- Download the new project to the device. A warning appears, stating that this operation will change the static interface and that the configuration software will not be able to automatically upgrade your device in the network.
- In case you are connected over CEA-709, another warning will tell you that the connection will be lost after the download, because the device will go to un-configured state due to the static interface change. It is therefore recommended to use TCP/IP for the connection.
- After the device restarted, use the **Export XIF...** function from the *Tools* menu to generate a new XIF file, matching the new configuration of the device.
- Execute the standard procedure of your network management tool to replace the old device with a new device and use the new XIF file to define the new interface.

NOTE:	<i>The above procedure assumes that there are no dynamic NVs present on the device. Manually changing the static interface of a device is not supported by any of the common network management tools, when there are dynamic NVs present at the same time. Therefore, updates to the static interface in mixed projects can only be done in plug-in mode.</i>
--------------	---

12.4 Configuration Recovery

This section describes procedures to recover from situations where the current configuration of the device does not match the current definition of the device in the network. This may happen as a result of a failed upgrade procedure or because the definition of the device in the network changed because the network had to be restored from an older backup.

Basically, there are two ways to resolve any such conflict:

- Change the configuration of the device to match the current definition in the network.
- Change the definition of the device in the network to match the current configuration.

The decision depends on the desired end result, that is, if the current network definition or the current device configuration is closer to the desired end result.

12.4.1 Change Device Configuration

In case the current definition of the device in the network represents the desired end result, the configuration of the device should be set back to match the definition in the database. This of course is only possible if a matching project is still available as a file on a PC. Assuming that a suitable project file is available, the device may be easily restored by following the appropriate procedure for a device replacement, as outlined earlier in this document.

To determine if a project is suitable or not, it is not necessary to execute the whole replacement procedure just to find out that the result still does not match the definition of the device in the database. Instead, check the following things upfront, before starting the download:

- Run the configuration software standalone, without connection to any L-VIS device and load the project which is considered to match the current device in the network.
- Check the program ID of the device definition in the network.
- If the ID starts with 9000a9... then the interface matches a standard device, without any static NVs. In this case, only consider projects which do not contain any static NV data points. The loaded project must have a Program-ID model number of 0.
- If the program ID starts with 9000d7... then the device has static NVs created and therefore uses a non-standard interface. In this case, check the last byte of the program ID. It is the program ID model number and it must match the model number of the project which you consider for download.
- Once a project with matching model number was identified, this will most likely be compatible as far as the static interface is concerned. To double check, you may go through the list of static NVs defined in the project and compare it with the NVs that should be available on the device according to the devices definition in the network. If the static interface seems to match, the replace procedure may be started.
- During the replace procedure, if the selected project causes data points to be reported during project load, which cannot be assigned to any of the existing dynamic NVs, then this project assumes a different set of dynamic NVs and is most likely not the right version of the project. Repeat the process with other projects. The only time when such reports are to be expected is during the first download in a two-stage replace procedure, as required for mixed projects (static and dynamic NVs).

12.4.2 Change Network Configuration

There are a few cases where the network definition of the device is changed to match the current configuration of the device. Such changes are normally handled by the network management program, but may be supported by functions of the configuration software as well.

While changing the device configuration requires one of the device replacement procedures to be executed, changing the network configuration usually means to restore a database backup which matches the configuration of the current device.

If the device uses static NVs only and the bindings which were originally used together with this project were saved and are available in a file, the network configuration may be adjusted by completely removing the old device definition (if any), executing the integration procedure for a pre-programmed new device, followed by a binding import to restore the original bindings.

12.4.3 Recover from a failed Update

When a project update including a change of the static interface fails, there may be some extra steps necessary to restore the last working setup or to resolve the original problem and finish the upgrade. Which one to use, depends on how far the upgrade procedure ran.

12.4.3.1 Failure before Database Update

If the procedure failed before the database update, that is, the device template in the database is still the old version, the device must be rolled back to the last project as well, so that the database and the device match again. The recommended procedure is detailed below:

- Follow the standard procedure for device replacement as appropriate for your project, but **do not run the configuration software in plug-in mode at any time**. Only use the software standalone, including any other configuration tasks you may be doing to other devices at the same time. If the project used static and dynamic NVs and the upgrade required a change of the static interface, your dynamic NVs may be lost and the replace procedure may report data points which cannot be assigned. Remove them from the project and continue.
- Once the device is replaced and commissioned, run the configuration software in plug-in mode.
- If the device lost its dynamic NVs, use the function **Recover DynNVs...** from the *Tools* menu now, to recover the set of dynamic NVs which were removed during the last upgrade.
- Most likely, the bindings will be lost as well. Use the function **Recover Bindings...** from the *Tools* menu to recreate the bindings as they were recorded before the upgrade procedure started.
- If the replace procedure reported missing dynamic NVs before, the project has to be loaded again. Clear the current project data from the editor and re-open it from the project file.
- Program the device to restore the system to the last working state.

12.4.3.2 Failure after Database Update

If the procedure failed after the database update, that is, the device successfully loaded the new project and the database was updated to match the new static interface, the only things which could still be missing are dynamic NVs and bindings which would need to be restored after the database update.

There are two possible options from here:

- Try to find the reason why the NVs or the bindings could not be restored, possibly checking the **Binding Report...** which is available from the *View* menu for errors, then try to use the **Recover DynNVs...** and **Recover Bindings...** functions from the *Tools* menu to first restore the dynamic NVs (if any) and then restore the bindings.
- Revert back to the last working setup by removing the device from the network database and roll back the device configuration by following one of the procedures for integration of a new device, but using the last working project. While doing this, **do not run the configuration software in plug-in mode at any time**. Also, skip the step of manually creating dynamic NVs, if instructed to do so by the standard procedure. If the old project actually used dynamic NV data points, these will be reported as unusable. Delete them from the project and continue. Once the device is re-integrated, recover the dynamic NVs (if any) and the bindings as described above. If dynamic NVs were recovered, the project needs to be loaded again so that the NVs can be used.

NOTE:	<i>Reverting back to the last working setup is much easier if a backup of the database is available. In this case, first restore the database from the backup and then simply follow the appropriate procedure for device replacement, using the last working project file. In this case, the configuration software may also be used in plug-in mode during the replace procedure, since the recover functions are not required.</i>
--------------	---

13 Special Functions (CEA-709)

13.1 Binding Management

When the configuration software is run in plug-in mode, additional functions to export, import, and remove the network bindings of the device are available from the *Tools* menu.

The **Export Bindings...** function scans the database and outputs all current bindings into a text file in CSV format. This file may be viewed in a text editor or spreadsheet program and used for documentation purposes or as a security backup before major changes are done to the device.

The **Import Bindings...** function reads a file which was previously created by the export function or manually created in a text editor and creates all the bindings which are listed in the file.

NOTE:	<i>If you use a network management tool with a graphic representation of your network bindings, you will need to update the drawing based on the modified database, so that the new bindings created by the configuration software become visible. This is normally not required for object oriented tools, where the new bindings immediately appear in the object tree.</i>
--------------	---

The **Remove Bindings...** function is not normally used, but may be helpful if the network bindings of the device should be reworked completely without deleting the device itself.

NOTE:	<i>This function is also useful for the rare cases where the static interface of a device needs to be upgraded manually and the network management tool has difficulties to take over the existing bindings during the upgrade. In this case, export the bindings, remove them, then do the manual upgrade and as the last step restore the bindings from the file.</i>
--------------	---

13.2 XIF Export

When the selected model is a CEA-709 model and the current project uses at least one static network variable, the menu item **Export XIF...** becomes available in the *Tools* menu. This can be used to create an XIF file for the current project, for later device integration. This is normally used when the network management software cannot run the configuration software in plug-in mode and the device must therefore be integrated manually, after it was programmed outside the network management with the configuration software running in standalone mode.

To generate an XIF that matches the actual device, there are two options:

13.2.1 Offline

A project may be designed completely offline, without any connection to a physical device. The target device type and firmware version are selected from the menu. In this scenario, the following extra steps are necessary:

- **Select Model Number:** Since the project is never downloaded to a device, the model number is not set automatically. Before exporting the XIF file, open the **Network Settings...** Dialog from the *File* menu and enter the desired model number for your device.
- **Select Transceiver:** Since no device is connected, the configuration software cannot automatically determine the transceiver used. Therefore, a dialog will appear and ask for the transceiver type which will be used by the device.

When the project for which this XIF file was created is later actually downloaded into a device, make sure to use the same model number as you entered before the XIF was exported.

NOTE:	<i>If the device already contains a project using the same model number, but the static interface of this project does not match the new project, the configuration software will automatically change the model number and the result would no longer match your XIF file. In this case, remove the incompatible project first and then download the new project.</i>
--------------	--

13.2.2 Online

In cases where the project is designed online (while connected to a device), all required data can be determined automatically. To make sure that the model number in the XIF file matches the model number which is automatically selected during project download, first download the finalized project to your device and then export the XIF file.

13.3 Remote Network Interface

On newer hardware of the CEA-709 model, there is support for remote network interface functionality, similar to that of a NIC-IP. The function is only available if the device is operated in FT-10 mode (not IP-852 mode) and only if the hardware can support it. If active, the device accepts connections from a PC on the TCP/IP interface and can act as a remote network interface, providing access to the FT-10 channel to which the device is connected. It also provides functionality to run the LPA software on this interface to monitor the FT-10 channel.

NOTE:	<i>Using this feature will cause additional load on the device, which should be considered at the time of project design. If the project already requires a lot of CPU time to run, it may not be feasible to use the device as a remote network interface at the same time.</i>
--------------	--

The current state of the RNI module is available in a system register called *RNI State*. It may be used to make a running RNI connection visible on the display, as a warning message or other element to alert the user.

13.4 CEA-709.1-A Operation

The CEA-709 models normally support the CEA-709.1-B standard, which means that the device accepts the extended network management commands used to configure the extra resources available on such devices (for example up to 512 address table entries on L-VIS). However, since the standard requires devices which were once configured using extended network management commands to no longer accept any of the older commands. Such a

device would no longer be usable together with older management software, which does not provide support for the extended command set.

To guarantee that the device will be usable together with older tools, the **Network Settings** dialog, which is accessible from the **File** menu, has a checkbox to enable a so called *CEA-709.1-A restricted mode*. In this mode, the device will not accept any of the extended commands and react according to the old standard. This also means that the device is limited to 15 address table entries.

13.5 Feedback Data Points

Feedback data points allow reading back the value written out over an output data point. In LONMARK systems, getting a feedback value is normally accomplished by creating a dedicated feedback NV on the remote device which can be bound back to the devices that are interested in the currently active value. This is especially important if multiple independent devices write to the same input of one device, since none of the writers may assume that the value he last wrote is still in effect.

Some nodes, however, do not possess such feedback NVs for certain functions. To support getting feedback values on such nodes, it is possible to create feedback data points based on existing output data points. This is especially interesting for bound output NVs (static and dynamic alike). The corresponding feedback data point is an input, which uses the original output NV for polling the target NV. In case the binding is changed, the new target is polled automatically. No additional input NV needs to be created for the feedback value, if the feedback data point feature is used.

To Create a Feedback Data Point

1. Select an output data point in the data point list of the **CEA-709 Port** folder, e.g. 'nvoHumid101'.
2. Right-click and choose **Create Feedback-Point** from the context menu.
3. A new input data point is created, having '_fb' appended to the original name, e.g., 'nvoHumid101_fb'. Note, that the feedback data point maps to the same NV index as the original output data point.
4. Choose an appropriate poll cycle in the data point properties for the feedback data point.

13.6 Configuration Properties

Configuration properties (CPs) are supported by the LNS network scan and the online network scan. They can be selected and used on the device in a similar way as NVs. There is a notable difference to NVs: CPs are part of files on the remote nodes. Reading and writing CPs on the device results in a file transfer.

The device supports both, the LONMARK file transfer and the simpler direct memory read/write method. In both cases however, one has to keep in mind that a file transfer incurs more overhead than a simple NV read/write. Therefore, polling CPs should be done at a much slower rate than polling NVs (e.g., every 10 minutes).

Another aspect is how CPs are handled by network management tools. Formerly, those tools were the only instance that could modify CPs in devices. Therefore, most tools do not automatically read back CPs from the devices when browsing them. This can result in inconsistencies between the actual CP contents on the device and their copy in the network

management tool. It is recommended to synchronize the CPs from the device into the LNS database before editing and writing them back.

To Synchronize CPs in NL220

1. Double-click on the device object in the device tree
2. Press the **Upload** button on the Configuration tab of the device properties (see Figure 36).

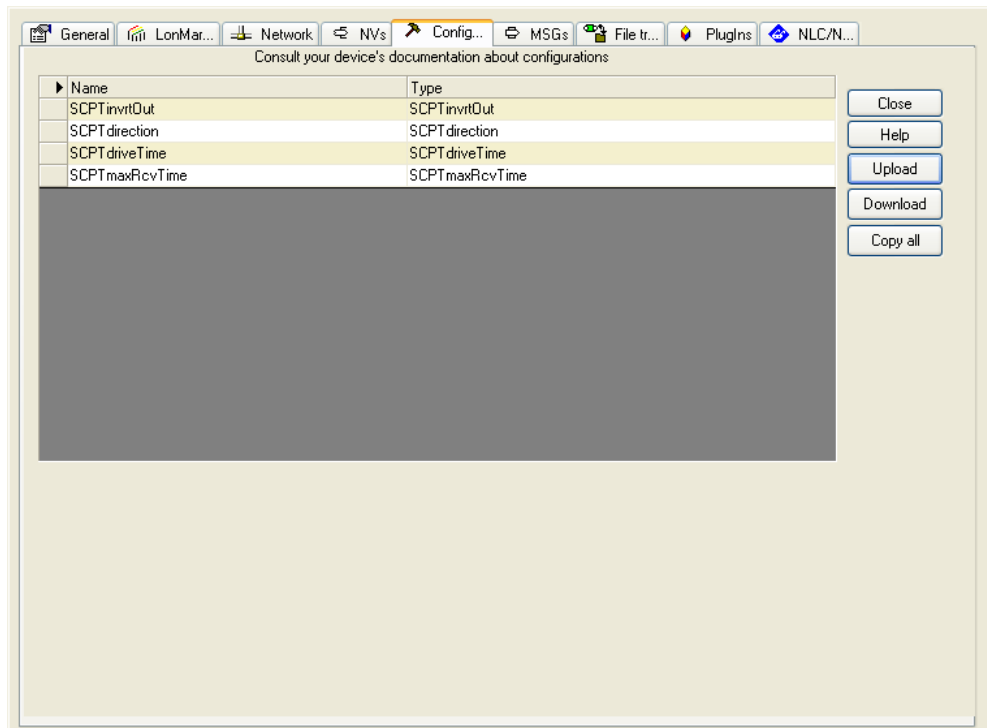


Figure 36: Configuration Tab for Configuration Properties in NL220.

To Synchronize CPs in LonMaker TE

1. Right-click on a device object and select **Commissioning → Resync CPs...** from the context menu.
2. This opens the dialog shown in Figure 37.

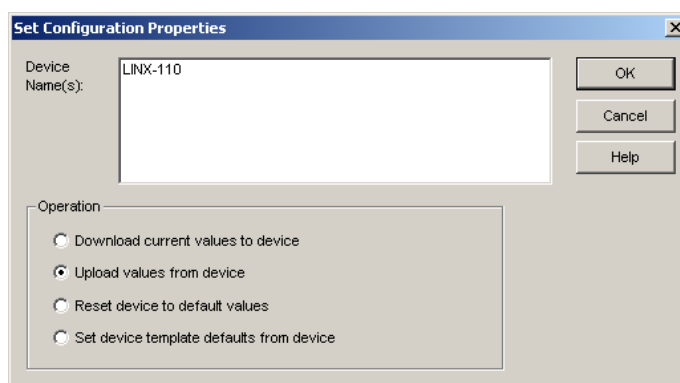


Figure 37: Set Configuration Properties in LonMaker TE.

3. In this dialog select the radio button **Upload values from device** in the **Operation** group box. To use the current settings of the device as default values for new devices, select **Set device template defaults from device**.
4. Execute the operation by clicking the **OK** button.

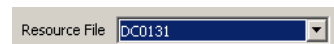
13.7 UNVTs and UCPTs

This device supports user-defined type, including user-defined network variable types (UNVTs) and user-defined configuration property types (UCPTs). In order to interpret the contents of user-defined types, the *device resource files* supplied by the vendor must be added to the resource catalog on your PC.

Once the resource files are installed, the CEA-709 network scan and the LNS scan will display the user-defined types from the resource files. Those data points can be used on the device like regular, standard-type data points. Also manual creation of UNVTs can be performed.

To Manually Create a Static UNVT

1. Perform the steps to manually create a static NV as described in Section 10.16.2.
2. When the **Create New NV** dialog appears, change the resource file from 'STANDARD' in the **Resource File** drop-down box to the desired, user-defined resource file



3. Then select the desired UNVT from the **Type** drop-down list below. This list will display the types of the selected resource file only.
4. Click **Create Static NV** to create the UNVT on the device.

13.8 Dynamic Network Variables

Some of the installation scenarios for CEA-709 L-VIS devices make use of dynamic network variables. This section gives a short description of how dynamic network variables are created in different network management tools.

13.8.1 LonMaker® Network Management Tool

To create a dynamic network variable, create a functional block (e.g. L-VIS1) for the device as shown in Figure 38.

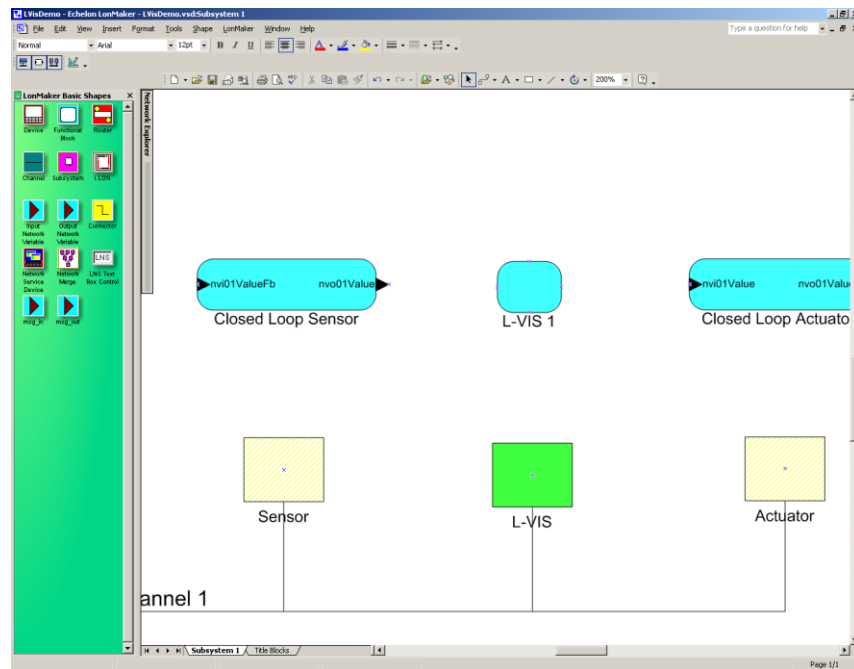


Figure 38: Create a function block

Drag an input or output network variable shape from the LonMaker® network management tool shape sheet and drop it on the function block. The dialog in Figure 39 is shown.

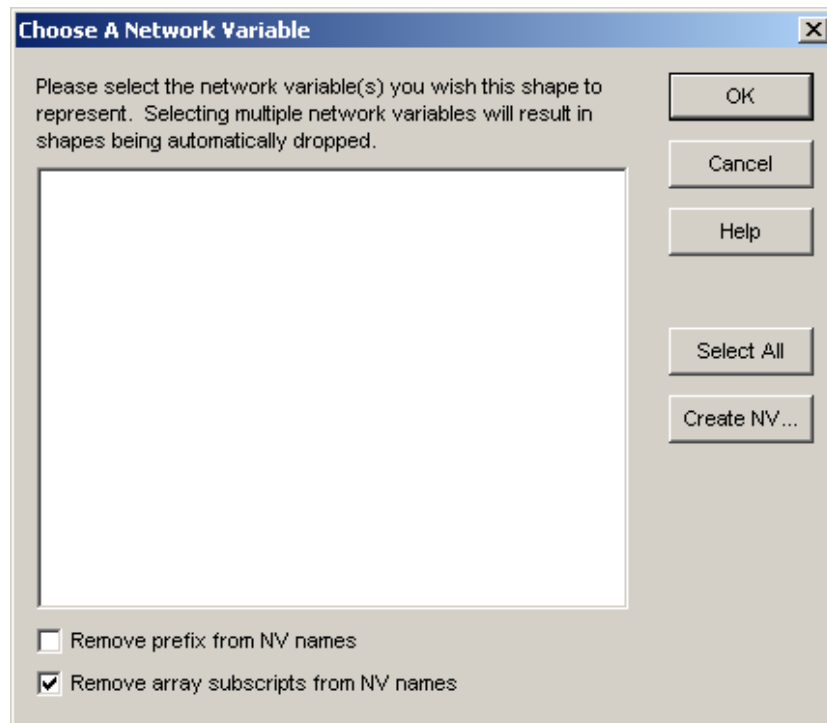


Figure 39: Choose network variable dialog

Select 'Create NV' and in the displayed dialog (Figure 40), enter the name for the new network variable and click on 'Browse'.

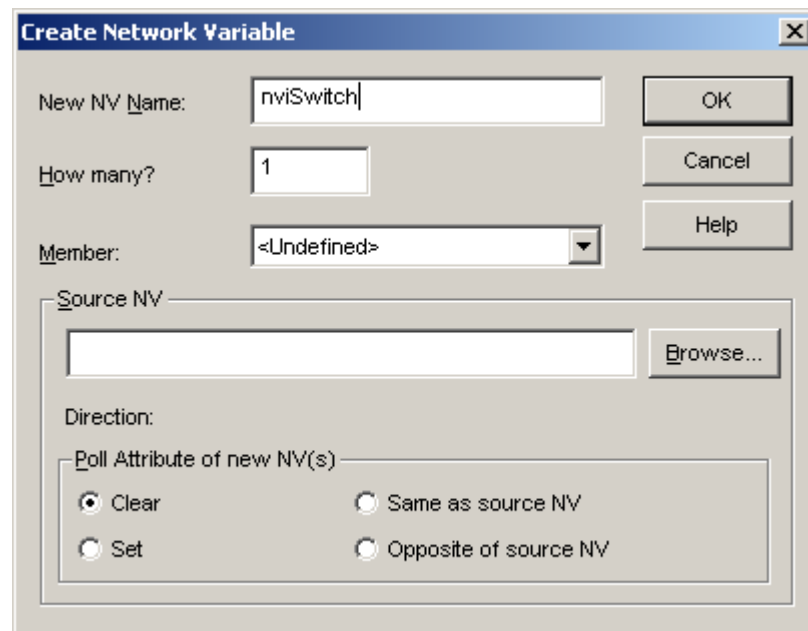


Figure 40: Create network variable dialog

In the dialog (Figure 41), select the network variable on the node to which the new dynamic network variable will be connected afterwards. The dynamic network variable will get some attributes of the selected variable, like the SNVT type, assigned. Click on OK to confirm your choice.

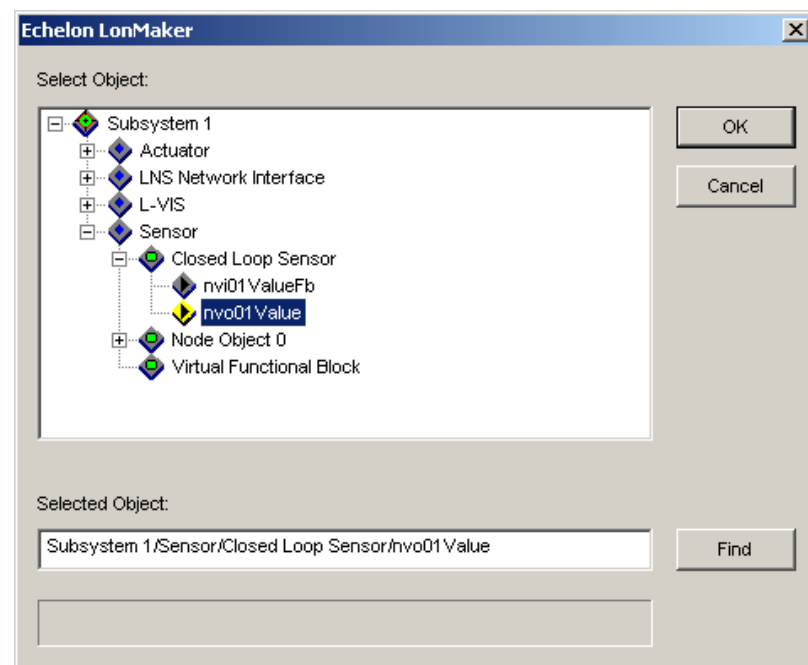


Figure 41: Select complementary variable dialog

Now the source NV name in the Create Network Variable dialog is filled in (Figure 42).

NOTE: Make sure to set the poll attribute of the NV to 'Set' if you are creating an input NV, in order to allow the device to support the receive timeout and poll on system start features.

When Finished, press the OK button and also confirm the Choose a Network Variable dialog with OK.

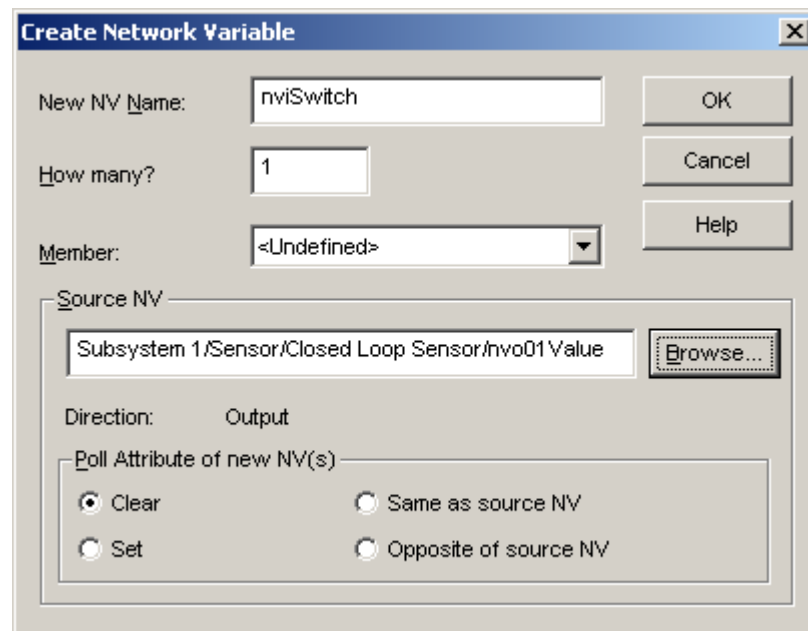


Figure 42: Create network variable dialog

This completes the creation of the new dynamic network variable (Figure 43). The dynamic network variable can now be bound to other network variables like a normal network variable and can also be used as a data point source or data point sink in the L-VIS configuration software.

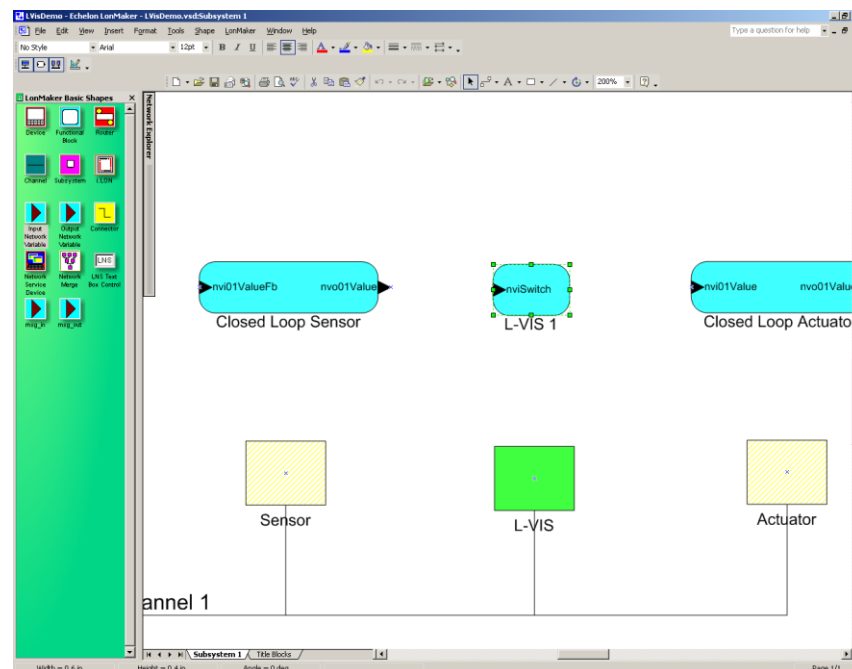


Figure 43: New dynamic network variable

13.8.2 NL220

To create dynamic network variables in NL220, first add the L-VIS device to your project. Then browse to the node which holds the complementary network variable of the new dynamic network variable to be created. Drag the complementary network variable and drop it on the desired L-VIS[...] LonMark object of the L-VIS device. In the dialog (Figure

44), enter a name for the new dynamic network variable and set the direction and the poll attribute as required. When done, click on *OK*.

NOTE: *Make sure to set the poll attribute of the NV to 'Set' if you are creating an input NV, in order to allow the device to support the receive timeout and poll on system start features.*

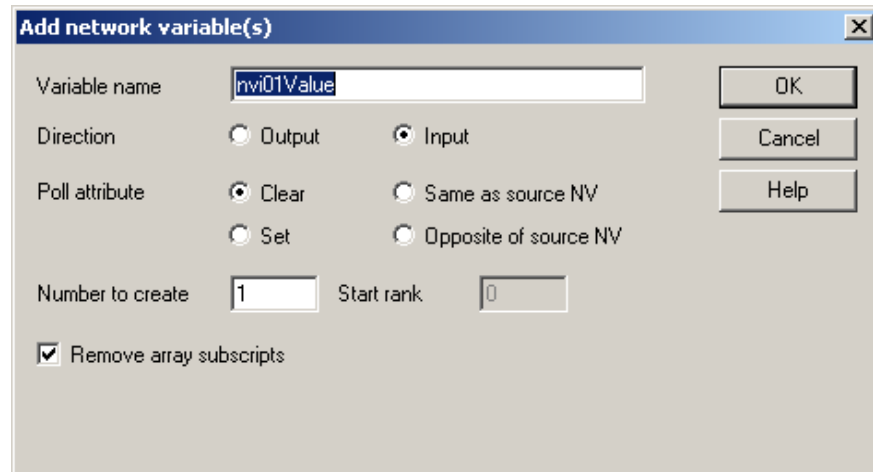


Figure 44: Add network variable dialog

This finished the creation of the dynamic network variable and shows the window which allows to bind the dynamic network variable to the complementary network variable (Figure 45).

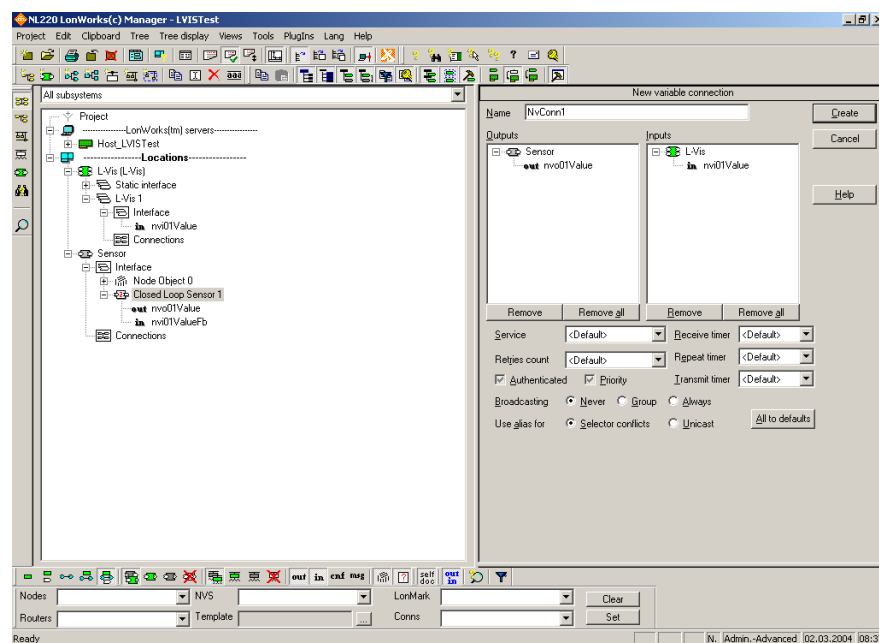


Figure 45: Network variable connection window

13.8.3 Alex

To create dynamic network variables in Alex, add the L-VIS device to your project. Select **Virtuelle Funktionseinheiten** and click on *Neue Virtuelle Funktionseinheit* (Figure 46).

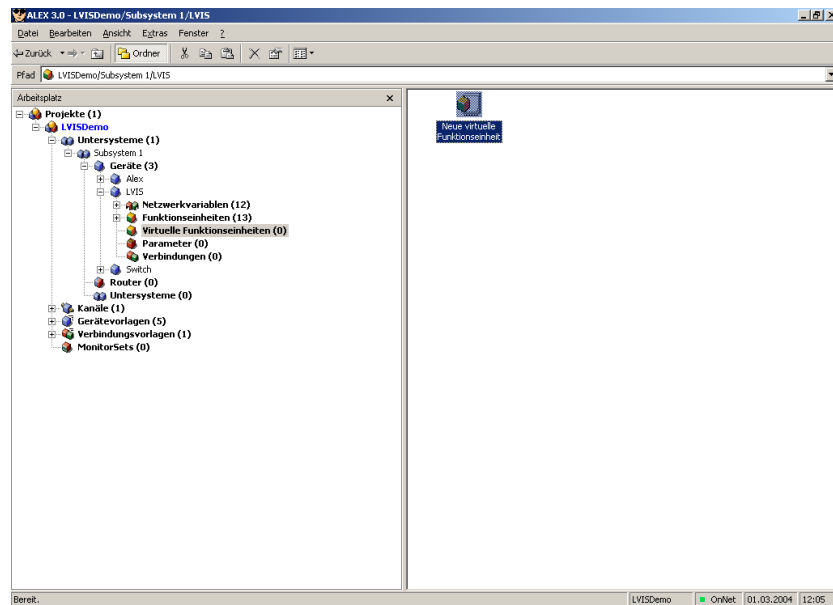


Figure 46: Alex main window

Enter a name in the dialog (Figure 47) and press *OK*.

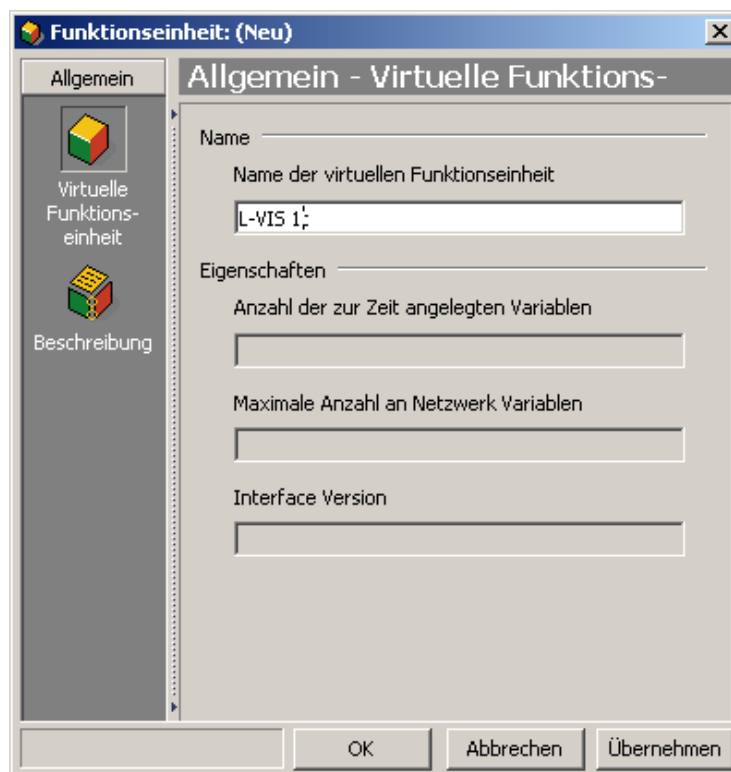


Figure 47: Virtuelle Funktionseinheit dialog

Open the Netzwerkvariablen collection in the tree view and select *Neue Netzwerkvariable* (Figure 48).

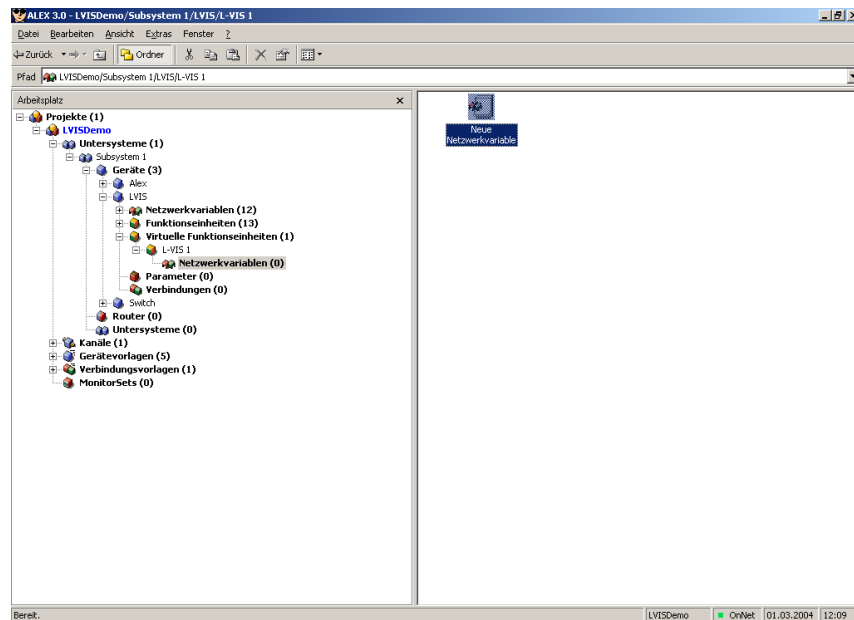


Figure 48: Create new network variable

In the Netzwerk Variable dialog (Figure 49), select the device template (Gerätevorlage) and the network variable to which the new dynamic network variable will be connected.

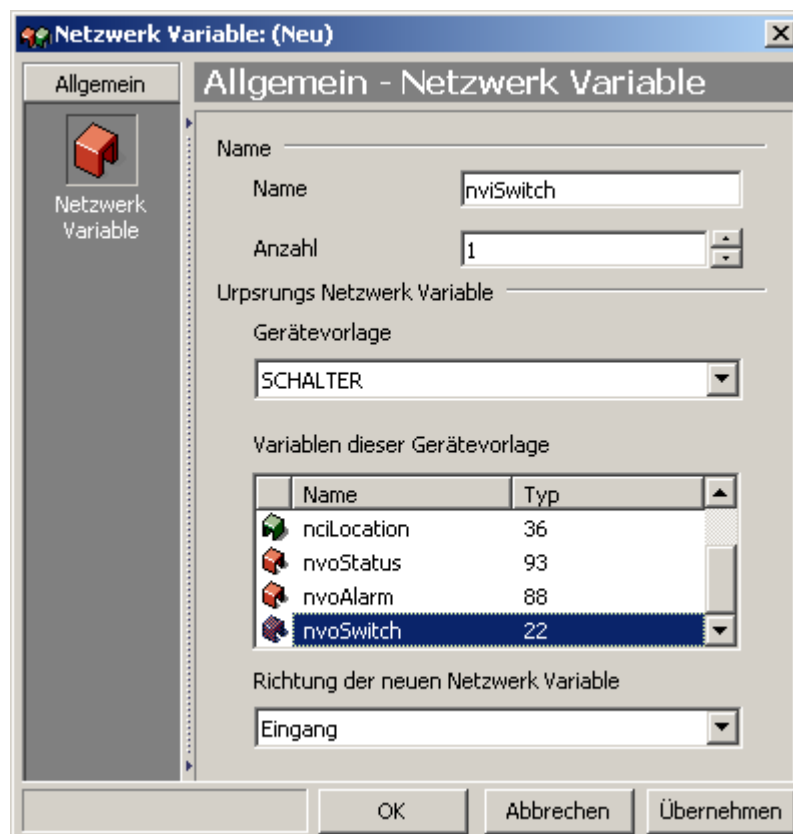


Figure 49: New network variable dialog

The dynamic network variable will get attributes like the SNVT type from this network variable. Select the direction of the new dynamic network variable and enter a name for the network variable. Finally, click on *OK* to confirm the creation of the new network variable (Figure 50).

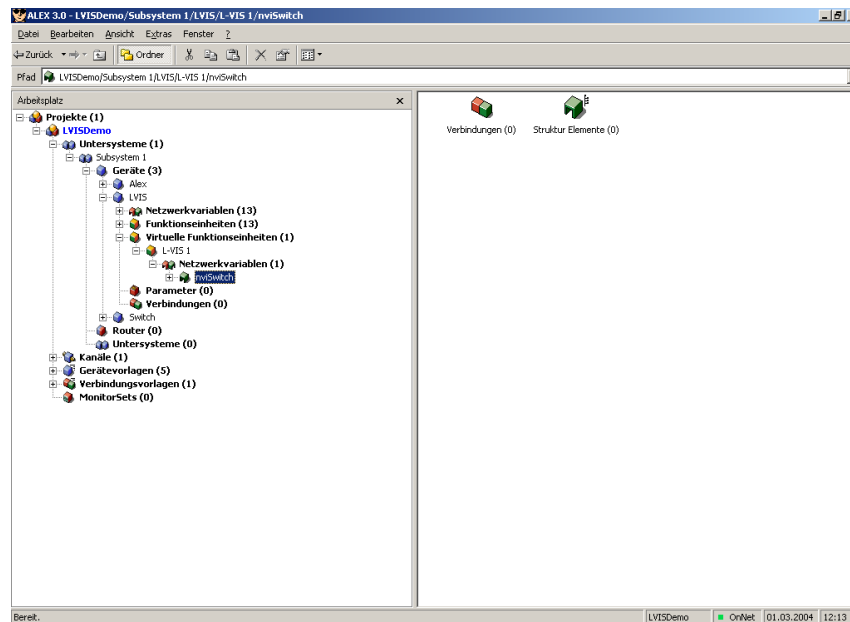


Figure 50: Finished creation of new network variable

13.9 Built-in LONMARK® Objects

The CEA-709 models of L-VIS provide a number of static function blocks which are available for use, even when no project is loaded. These blocks are described in the following sections.

13.9.1 Timer Object

The timer object implements a LONMARK® Real Time Keeper object according to the LONMARK® Functional Profile #3300. It sends out the current time in the nvoDateTime network variable of type SNVT_time_stamp. The object is configured using the standard configuration properties:

- SCPTmanualAllowed
- SCPTmasterSlave
- SCPTobjMajVer
- SCPTobjMinVer
- SCPTsummerTime
- SCPTupdateRate
- SCPTwinterTime.

The configuration properties are defined in the LONMARK® functional Profile #3300 document.

13.9.2 Switch Object

The switch object implements a LONMARK® Switch object according to the LONMARK® Function Profile #3200. It sends out an nvoSwitch network variable of type SNVT_switch

and receives a feedback over nviSwitchFb. The object is configured using the configuration properties

- SCPTmaxOut
- SCPTmaxSendTime
- SCPTminSendTime
- SCPTstepValue
- SCPTtimeout.

The configuration properties are described in detail in the LONMARK® Function Profile documentation.

On L-VIS, the switch object is controlled by a switch or push button connected to the *Switch* terminal IN0 of the device. The object can operate in 3 different modes:

1. **Switch mode:** Use this mode when a two position switch is connected to the input terminals. To select switch mode, set SCPTstepValue and SCPTtimeout to zero.
2. **Push button mode:** Use this mode when a push button is connected to the terminals and you want to toggle the light on and off with each push of the button. To select push button mode, set SCPTstepValue to zero and SCPTtimeout to a value greater than zero
3. **Dimmer mode:** Use this mode when a push button is connected to the input terminals and you want to use the button to toggle the light on and off with each short press of the button and ramp the light intensity up and down when holding down the button. To select and configure dimmer mode, set SCPTstepValue to the amount of change wanted for each step when ramping up or down and set SCPTtimeout to the desired maximum push time, which is detected as a short press (the time to wait before starting to ramp the light intensity up or down). The speed at which value updates are sent out during ramping is controlled by SCPTminSendTime.

13.9.3 Relay / LCD Backlight control

The relay object implements a LONMARK® Lamp Actuator object according to the LONMARK® Functional Profile #3040. It receives the input value of type SNVT_switch in the variable nviSwitch and sends out a feedback of the current state in nviSwitchFb. The output network variable reflects the state of the relay input of the old LVIS-3ECTB device. The newer touch screen models do not provide a relay and the lamp actuator object on these models is used to control the LCD backlight.

To overrule the value which is adjusted in the L-VIS Plugin Software, the state part of the nviSwitch network variable has to be set to 1. In that case, the value part of the network variable controls the brightness of the display from 0 to 100%. If the state part of the variable is set back to 0 again, the backlight automatically uses the values which are adjusted in the L-VIS Plugin software. This feature can be used to adjust the backlight in relation to the ambient brightness, which could be measured with a sensor.

On newer firmware revisions, the backlight may also be controlled via an internal system register, which is available as a data point and may be used freely across the project. This is the preferred way of controlling the backlight.

13.9.4 HVAC Temperature Sensor

The temperature object implements a LONMARK® Temperature Sensor object according to the LONMARK® Functional Profile #1040. It sends out the current temperature in the nvoHVACTemp network variable of type SNVT_temp_p. The object is configured over the configuration properties

- SCPTmaxSendTime
- SCPTminDeltaTemp
- SCPTminSendTime
- SCPToffsetTemp.

The configuration properties are described in detail in the LONMARK® Functional Profile specification.

On the L-VIS device, the temperature sensor object reads the current temperature value from an external temperature sensor connected to the input terminals labeled 'TEMP' on the device.

The same temperature is also available via a system register, which is the preferred way to read the temperature if it is used directly on the L-VIS device.

14 Special Functions (BACnet)

14.1 EDE Export of BACnet Objects

The current BACnet server object configuration of BACnet L-VIS models (LVIS-ME200) is accessible as a set of CSV files following the EDE format convention. They can be downloaded via FTP from the directory `/data/ede` on the L-VIS device. The files are:

- **lvvis.csv**: This is the main EDE sheet with the list of BACnet objects.
- **lvvis-states.csv**: This is the state text sheet. For each state text reference in the main sheet, a line contains the state texts for this multi-state object.
- **lvvis-types.csv**: This is the object types text sheet. The file contains a line for each object type number. Note, that lines for standard object types can be omitted.
- **lvvis-units.csv**: This is the unit text sheet. The file contains a line for each engineering unit enumerator value. Note that lines for standard units can be omitted.

NOTE: *First download the project to the device, wait for the device to restart, and then access the EDE files. The device will create files matching the loaded configuration when it starts.*

14.2 BACnet Network Settings

For BACnet models, selecting the command **Network Settings...** from the *File* menu of the main window opens a small dialog to configure BACnet specific settings. The following settings are available:

- **Character Set:** Allows changing the character set used for string values on the BACnet network. This setting affects encoding of all string properties of server objects on the device. It does not limit the ability of the device to receive and process string values from other devices in other encodings. See also the note below for more background information.
- **Enable unsolicited COV:** Send out value changes of local server objects as a broadcast to all other devices.
- **Use global instead of subnet wide broadcast:** Broadcast messages are not directed to the local subnet but instead are sent to the global broadcast address. This is not conforming to the standard, but some third party devices need this to work around problems in their software. If needed, it can be turned on here.

- **Enable periodic I-Am broadcast:** The device sends out I-Am messages on a regular basis to notify others of its presence.
- **Keep OWS values in device:** A parameter download, for example via LWEB-900, does not overwrite any persistent property values of BACnet server objects. It is assumed, that these are managed by a BACnet OWS and are written via the BACnet network.
- **I-Am Interval:** The interval used for periodic I-Am messages, if enabled.
- **Default Polycle:** The poll cycle to set for newly created client mappings.
- **Default COV Expiry:** The default expiration time for COV subscriptions.
- **Default Write Priority:** Write priority for all client mappings, which are set to use the default write priority.

NOTE:	<i>Since all strings in an L-VIS project are Unicode strings (ISO-10646), the default setting is to use UTF-8 encoding for BACnet string properties. When using UCS-2, be aware that there are many tool vendors which do not support UCS-2 strings, and are completely unable to display such strings in their tools, even if the string itself contains only ASCII characters. If the device is to be used together with such systems, the character set must be set to either ISO-8859-1 (Latin-1), which is a subset of ISO-10646 and is sufficient for most western languages, or to UTF-8, which is backwards compatible to ANSI X3.4 (US-ASCII) as long as no foreign characters are used. If the character set is reduced to ISO-8859-1, the user must ensure that no characters outside this character set are used for server object names and descriptions. Unprintable characters will be replaced by a question mark.</i>
--------------	--

15 Advanced Topics

This chapter discusses some of the more advanced topics.

15.1 Access Control

Access control is organized in 16 levels. Pages and input controls can be assigned a minimum access level required to view the page or enter data on an input control. Any access level higher than the specified minimum level automatically grants access to the protected object.

To reach a specific access level, there are a number of possible options:

- Switching to a page which cannot be viewed at the current access level will automatically bring up a system page where a PIN code can be entered in order to increase the access level and access the page. See also the note below.
- On a normal project page, or a dedicated login page, a numeric input control can be placed to allow the user to write a PIN code value to the system register called *PIN Code Enter*. Any value written to this register will be interpreted as a PIN code and the system will try to log the user in using this code. If none of the defined access levels has a matching PIN code, the access level is set to 0 (log out). To construct a suitable control, make sure to set the input value range to 0000-9999 and check the *Password Input Keypad* option.
- Using a data point connector, a PIN code can be received from the network and forwarded to the *PIN Code Enter* register to allow remote login using a PIN code.
- The desired access level may be set directly by writing a value between 0 and 15 to the system register called *Access Level*. This is less useful if allowed from a control on a project page, but may be convenient if there is an external access controller in the network, where the user was already authenticated (for example using an RFID card). The access controller can communicate the successful authentication over the network to the L-VIS device, where a suitable access level can be set directly, without using a PIN code.

NOTE:

*If the user was automatically redirected to the login page, due to insufficient access rights for the page he wanted to view, any write to the **PIN Code Enter** register will not only evaluate the PIN and adjust the access level, but will also either forward the user to the desired page (access granted), or return to the page he came from (access denied). This ensures that the user does not get stuck on the login page, in case he is not able to enter a valid PIN code.*

For security reasons, it is possible to change the PIN code for each level by writing the new PIN code to the respective *PIN code Set* register. There is one register for each of the access

levels 1 through 15. The new PIN codes will be stored on the device and remain in effect until a new project is loaded with a special option set, telling the device to reset its PIN codes to the values stored in the project.

15.2 Multi-Language Projects

Starting with LVIS firmware 7.0 and LWEB-802 version 3.4, it is possible to build projects containing text and font resources for multiple different languages. The user can switch between languages during run time without reloading the project, just by writing the desired language number to the **Language** system register.

NOTE:

*For a simple language selector, just connect the **Language** read and write system registers to a text control. A suitable mapping table with names for all defined languages and their language numbers will be built automatically. Make sure to update the mapping table after changes to the project languages. The **Reset** button on the mapping table property page can be used to rebuild the table automatically.*

Using multi-language projects has a number of advantages compared to other approaches which use copies of every control, one for each language, and show the correct set of controls using visibility triggers or separate pages (one page for each language). Since pages and controls do not need to be duplicated for every language, it is much easier to apply changes later on. Also, since the number of controls is not multiplied, the project uses less resources and is easier to handle during development. Multiple layers of controls on top of each other to support multiple languages are difficult to handle in the LCD preview.

Another benefit of multi-language projects is the ability to translate the project externally. All text resources and font selections for objects in the project can be exported to a file, which can be given to an external translator. Translated resources can then be imported back, and checked in the LCD preview for layout problems resulting from different space requirements for different languages.

15.2.1 Configuration

Language setup and configuration is done on the **Localization** page in the project settings dialog. Languages can be added by copying resources and settings from an already existing language, the order (numbering) of languages can be defined, and languages no longer needed can be removed from the project. See Section 11.3 for instructions how to get started with a multi-language project.

15.2.2 Editing multi-language Projects

There are a few things to consider when working on multi-language projects. First of all, any **text** entered in the text input field of a control, or mapping table, as well as any **font** selection, usually applies to the **currently selected language** only. This may be confusing when languages are added to the project early in the project development cycle. Sometimes the wrong language will be selected, causing entered input to end up not where the user intended it to be.

Adding languages near the end of the development can minimize that risk, but tests regarding the multi-language features, such as switching between languages, will have to wait until the languages are available.

15.2.2.1 Format Patterns

Format patterns for localized number or date controls may be the same for all languages, but if the project already contains multiple languages, they still need to be set for each language, either separately or together using language multi-selection. After all, they may be different if additional text is added around the format specifier.

As explained in Section 8.12.3 regarding the **Editor** property page, it is possible to set text or a font for a number of languages at the same time, however, it is usually better to start out with the primary language only, and add other languages to the project when most of the content is already in place. This way, any format patterns and font selections which are the same for all languages, are automatically taken over from the primary language.

15.2.2.2 Control Size

Controls maintain only **one common position and size** for all languages. This means that page layout is independent of the selected project language. The same layout needs to accommodate all expected content for all languages, so make sure to leave enough room for wider text, and consider alignment of labels and numbers to handle dynamic label width. For numbers, it is usually a good idea to use right aligned content, with padding for the unit string such that the actual numbers of multiple lines turn out vertically aligned.

Using **auto-size** for text based controls is usually not possible in multi-language projects. The configuration software does consider all available resource settings (text and font selection for all languages) to calculate the required size, but in practice, not all translations will be ready at the time the page layout is done, and in case the control uses localized date or number formatting, it may be hard or even impossible for the software to determine the exact maximum space requirement. As a consequence, control dimensions should be set manually to accommodate expected space requirements for all languages.

15.2.2.3 Images

Since images are usually not language dependent, images assigned to a bitmap control, or images in a mapping table, are used equally for all languages. There is no mechanism to use different images for different languages, other than using individual controls and showing the desired control for each language, using visibility triggers.

15.2.3 Translation

Translation can be done directly in the configuration software, or externally in a text editor or any translation software supporting translation of XML files.

To edit a certain language or review an imported translation, select the desired language from the *View / Language* menu. Alternatively, select the language from the list of available languages on the **Editor** property page. The LCD preview will show the project in the selected language, and any changes to text or font selection will modify this language only. Since there is visual feedback via the LCD preview, it may be easier to pick the correct translation based on context and available space, compared to external translation. On the other hand, the translator needs to have the configuration software installed and be familiar with its use, which may not be the case.

15.2.3.1 Language Export

To translate a project externally, prepare the required XML files first. A number of different ways to export string resources is available from the *Tools / Export Strings* menu:

- **Default Language:** A single language XML file is generated, containing all strings and font settings from the default language (the first language in the list).
- **Current Language:** Same as above, but generates an XML file for the currently active language. This can be useful to generate an updated language file, after translations were added from multiple different sources.
- **All in one file:** Generates a multi-language XML file, containing translations for all languages for every string or font resource in the project.

- **All in separate files:** Similar to a single language export, but automatically generates one file for every language. The given file name will be annotated with the locale specification to generate a unique file name for every file.

Which of the above files are needed depends on the translation workflow and the external tools involved in the process. It will be required to understand the capabilities of external translation tools, external translators, and the XML file format. See the next section for a description of the single language and multi-language XML file.

15.2.3.2 XML File Format

The configuration software can generate two types of language XML file. A single language format and a multi-language format. The difference is small: The single language file contains a **lang** attribute at the top level **resource** element, specifying the language for all strings in the file. The multi-language file contains this attribute with every **string** element instead, since a resource item will contain multiple strings for different languages.

The basic structure is the same for both formats. The root element contains one **<resource>** element (with or without language attribute). This resource element contains one **<item>** element per string or font resource. Each item has a unique **id** attribute, to identify the resource item throughout its life time and from one export to the next. As long as the string exists in the project, it will maintain the same unique ID, such that it can be used as a resource key by external translation software to identify the resource.

If the item is related to a visible object in the object tree, it will also contain a **path** attribute, stating the location of the object to which the resource applies. This provides additional context information for an external translator, and allows someone with access to the configuration software to look up the object in the LCD preview. Here is an example for an item element:

```
<item id="0x00ae09c8" path="Root Menu / Menu Item / Page 1 / Title">
```

Below the item will be one or more **<string>** elements, containing the actual data for the resource. In a multi-language file, there will be one string per language, and each string element will have a **lang** attribute, identifying the language to which it applies. A single language project will just contain one string element without language identifier. The following is an example for a multi-language file:

```
<item id="0x00000070">
  <string lang="en-US">Global Event List</string>
  <string lang="de-AT">Alle Schaltzeitpunkte</string>
  <string lang="ja-JP">グローバルイベントリスト</string>
</item>
```

In case the resource is a font selection, the string will contain the name of the font to use for the given language. The path attribute will have **Font** appended to the object path, to denote the resource as being a font resource. Here is an example:

```
<item id="0x18c5481c" path="Menu / Item1 / Page1 / Schedule : Font">
  <string lang="en-US">12x16-ROM-Fixed-R</string>
  <string lang="de-AT">18pt-Candara-Italic</string>
  <string lang="ja-JP">18pt-MS PGothic-Regular</string>
</item>
```

Note that during external translation, the translator needs a list of fonts that will be available in the project to choose from. The configuration software makes no attempt to add a new font based on a translated font resource name during import of translations.

When the exported XML file is imported into an external tool capable of translating a given XML to different languages, the user is usually expected to denote the elements and attributes to translate, and identify key elements or attributes for sorting and tracking of items. Some tools could also use the locale specification of the language attribute to

automatically select the correct language or generate the attribute for the translated string resources in a multi-language output file.

When an external tool is used, one would usually export the default language and import it into the tool. Then add target languages as required, translate, and generate translated single language XML output files from the tool, for import into the configuration software. Whenever objects are added or removed, the default language XML would be re-exported from the configuration software and re-imported into the external software, to update the external translation project. Due to the unique IDs assigned to each resource entry, the tool will be able to identify items that were removed, items that were added and items for which the default content was changed (the translation needs to be revisited).

15.2.3.3 Language Import

Translated XML resource files can be imported from the *Tools / Import Strings* menu. Browse to the XML file and select it for import. Based on the language and unique ID attributes in the file, the configuration software will try to match strings from the file with existing resources, and update the project accordingly. Resources for languages which do not exist in the project are ignored, so make sure to add a suitable language definition to the project before importing the translation.

15.2.4 Copy/Paste

It is possible to use copy/paste between different multi-language projects. During copy, all available information from the source project regarding languages and translations for the copied objects will be stored in the clipboard. During paste, the available data will be used to take over translations and fonts for as many of the destination languages as possible. Text and font selection for languages which exist in the destination project but not in the source project will be initialized from the default language of the source (the first language in the source project). See also Section 8.5.3 about copy/paste in general, and 8.5.3.3 specifically regarding multi-language projects.

15.2.5 Compatibility

In case a multi-language project is downloaded to a device running firmware version 6.2 or older, or the project is saved for such a firmware version, only the currently active language will be saved. A warning will be displayed, stating that not all information can be stored and data will be lost. However, the project will retain all languages in the editor, such that a different language may be selected and the project may be saved a second time under a different name. This process can be repeated until all languages were saved to individual project files.

In case a multi-language LWEB project is loaded by an LWEB-802 version older than 3.4, the project will load and show pages and controls, but text based controls, like text, number, or date controls will be missing all text and use a default font, since the older client will not be able to read the language specific data.

15.3 Project Statistics

A dialog showing information about the number of currently defined objects and their memory requirements can be opened via the command **Project Statistics...** which is available in the *File* menu of the main program window.

The dialog shows two object lists. The first list shows the different object types, the total number of objects used of each type, as well as the largest object in terms of memory usage. For object types which require persistent storage in the file system in addition to runtime memory, two separate lines are shown to indicate the object with the largest amount of RAM usage and the object with the largest amount of FLASH usage, which may in fact be different objects.

The second list shows individual objects sorted by accumulated size, starting with the largest object. Note that the indicated size is the total size of the object itself plus all of its descendants, so it corresponds to the amount of memory which would be freed if the object would be removed from the project. Selecting an item from this list will select and highlight the corresponding item in the main window.

This dialog may be helpful to scan the project for items which cause undesired memory usage and should be optimized to improve the projects memory requirements.

15.4 Automatic Project Builder

The configuration software provides a function to automatically create a project or parts of a project from a selected set of data points. For each data point folder, it creates a menu item and adds pages and controls to show the values of all data points in that folder. In case of sub folders, a sub menu is created to hold the menu items for the sub folders and so on.

The function is called from the context menu of a menu object or a menu item object, as explained in sections 8.5.10 and 8.5.11. Calling the function opens a configuration dialog to select the desired base folder for which the project should be generated, as well as some additional options.

The desired base folder is selected from the data point folder tree on the left. Note that in case the function was called on a menu object, any data points located directly in the selected folder will be ignored, since a menu object cannot have pages attached to it. The selected folder will only be scanned for sub folders, and menu items will be created for each sub folder. When the function was called on a menu item object, pages will be created for the data points located in the selected folder, and a sub menu with items and pages will be created for the sub folders (if any).

15.4.1 Options

Additional options are available to the right of the folder tree.

- **Basic page layout:** Select one of the available layouts. Currently available are simple one to four column layouts.
- **Element indentation:** This option sets the amount of indentation for elements of structured data points. In case a structured data point has nested sub structures, structure members will be indented based on the position of their parent element.
- **Page margins:** Any of the automatically created content on a page will be limited to the area defined here. Set the margin to zero to allow the generator to use the entire page. In case you use a page template to add additional content to each of the auto generated pages, set the page margins to restrict the generated content to the area reserved for it.
- **Allow data entry for input data points:** Creates input controls for input data points, such that their value may be modified. This can be useful for OPC client data points.
- **Allow data entry for output data points:** Creates input controls for output data points. This is on by default but may be turned off to create a project where the user can see the current value of output data points but may not change their value.
- **Add data point folder path headline:** This option adds one line at the top to show the current location within the data point folder hierarchy. A “show menu”

action is attached to this line, allowing the user to open the menu when touching the headline.

- **Add page numbering:** This option adds a page numbering (current/total) at the right end of the headline. If the folder path headline is not activated, the headline will be enabled without the path (just the page numbering).
- **Add links to sub-folders:** This option places the names of all sub folders at the beginning of the page and then continues with the actual data points. Each of the folder names has an action attached to it which links to the first page of the respective sub folder, such that navigation to sub folders is possible without opening the menu.
- **Add page action to move one level up:** This option adds an action to each generated page which links to the first page of the parent data point hierarchy. This action triggers whenever the user touches the page somewhere outside an input control, for example on a data point name or an unused area of the page. This allows navigation back to the previous level without opening the menu.
- **Use collections for label / value pairs:** When enabled, each pair of text label and value field will be placed in a collection for easier manipulation, in case the generated layout is going to be modified manually later on.
- **Use data point description as value label:** If available, the text label will display the data point description instead of the data point name.
- **Do not add auto-created page content:** This option can be used in special cases, where the page generator is only used to generate one page for each data point folder and instantiate a given template on the page. See Section 15.4.4 for a discussion of custom pages.
- **Add template instance to each page:** This option instructs the page generator to add an instance of a template to each of the generated pages. The name of the desired template is entered in the text field below and defaults to 'Auto-Page'. If a template by this name is found, it is used. Otherwise a new, empty template is generated and instantiated on each page. Such a template can be used to add a navigation bar or other static design elements to the automatically created pages.

Once the desired options are set, the project builder can be started by pressing the **Start** button. Note that this button will be available only when the current settings would actually generate content. As long as no data point folder is selected or a folder without sub folders is selected and the generator was called on a menu object, the button will be deactivated.

15.4.2 Page Design

Any new object created by the project builder will be created with the currently defined defaults for the respective object. In most cases, it will be required to change these defaults from the built-in ones before content is created. To do this, an object of the desired type must be selected and configured to match the requirements (fonts, colors, format strings and so on). Once the object looks OK, open the *Common Properties* tab and press the button **Set as Default** to store the properties of the selected object as defaults for all new objects of the same type.

Objects which may be generated by the project builder and which should have their defaults configured are:

- **Menu:** Used for sub menus. Configure the desired width of the menu container, as well as colors, scroll bar, and the font for the menu header.

- **Menu Item:** Used for data point folders. Select colors and font and make sure the menu container is wide enough to hold the longest data point folder name.
- **Page:** Pages are created to hold the controls to visualize the data points. Set the desired page background color or background bitmap.
- **Text Control:** Text controls are used for the page header (path and page number) as well as data point names and textual data (from string, binary, or multi state data points).
- **Number Control:** Number controls are used to visualize and enter numerical data. Note that the format string can be set to modify the display of numbers. Since the same number control will be used for a wide range of different values, it is often useful to use the %g format instead of the fixed point %f.
- **Trend Control:** Trend controls are created when trend data points need to be visualized. They are always instantiated full-screen, that is, one control per page.
- **Scheduler Control:** Used to visualize schedule data points. These controls are always instantiated full-screen, similar to trend controls.
- **Alarm List Control:** These controls are instantiated full-screen to visualize alarm server/client data points. Note that the control will not use the full page height, since it will be followed by three lines of extra information related to alarm server and client data points.

Note that the configured defaults are stored with the project, such that when you open the project the next time, the same defaults still apply.

15.4.3 Page Template

To further customize the generated pages, the project builder can be instructed to instantiate a given template on each of the generated pages. This template may be used to define additional content like a navigation bar or other static page content like a company logo. In case the auto generated page header is activated, the template may be used to place a colored bar where the header will be located.

NOTE:	<i>Do not forget to set the page margins when generating content, such that areas occupied by objects from your template are not also used by the project builder.</i>
--------------	--

The most common objects used in such a template would be buttons for navigation such that the user does not need to open the menu. The following actions would be used:

- **Prev Page** to navigate to the page before the current page (flip through the data points of one folder, spread out on multiple pages).
- **Next Page** to navigate to the next page with data points of the same folder.
- **Show Page** with a relative path of `../../$P1` to navigate to the first page of the parent data point folder. To create this action, set the action type to *Show Page*, press the button *Select...*, choose the option *Internal Link* and enter the above path as the path to the target page.

Since the contents of the page template are not generated or modified by the project builder, the template must be designed manually. This can be done before or after the project was generated. Changing the contents of the template after project build is possible. All generated pages will be updated with the new template contents automatically.

NOTE

Changing the template such that it requires more space on the page or occupies different areas of the page will require a rebuild of the pages, since the margin settings will need to be adjusted. Also, if the default settings of a control or other object should be changed, a rebuild of the project will be required to take over the new settings.

15.4.4 Custom Pages

The page template mechanism can be used to auto-create pages using a custom layout.

Consider the case where a large number of remote devices of the same type were scanned and data points for each device were created such that there is one folder for each device. Also, a complete page was created to visualize one such device. This page already includes all graphics and other display elements like bars, numbers, analog meters, and so on.

In this scenario, one would define the page as a template and instantiate it once for every device which should be visualized. While this is not a very complicated task, it may still be faster to do it using the page wizard. Here are the steps how to take advantage of the page wizard to create all the necessary menu items, pages, and template instances automatically:

1. Create the custom page as a template, and use **the first template base path** to specify the data point folder of the device for which the template is designed. See Section 9.15 for more information about templates.
2. Start the page wizard on the menu which should be filled with the auto-created content. This can either be the root menu or a sub menu of the project.
3. In the project builder dialog, select the data point folder which contains the device folders. The selected folder will be scanned for sub folders and a new item and page will be created for each sub folder found.
4. From the list of available options, select **Do not add auto-created page content** and **Add template instance to each page**. Specify the name of your custom page template.
5. Click **Start** to create the content.

The wizard will create a new menu item for each of the available sub folders, add one page to each item, instantiate the custom page template, and adjust the first base path of the template instance to point to the data point folder of the respective device.

15.5 Local Data Point Capacity

The number of local BACnet server objects is limited to a maximum number of 512 on the LVIS-ME200, and 1000 on all other models supporting BACnet. In most cases, the L-VIS device operates as a client, retrieving and controlling the value of server objects on other devices. Therefore, there are usually not many server objects required on the L-VIS device itself. The number of client connections to other devices is limited only by available system resources and network bandwidth.

Models which support CEA-709 allow the user to set the capacity for local network variables (including static and dynamic NVs, excluding NVs which the firmware creates automatically) in the *Network Settings* dialog, which is accessible from the *File* menu of the main window. The default setting, which matches the configuration of an empty L-VIS device, is 512 and must not be changed if the static interface of the device should stay at default configuration. If the static interface is allowed to change, the capacity may be increased to support up to 1000 NVs using the input field *Max. Number of NVs*. Keep in mind, that an increased NV capacity requires system resources which may otherwise be used for the project, so this setting should not be set significantly higher than necessary.

See also section 15.11 regarding device resource limits.

15.6 Avoiding Update Loops

Especially when a lot of data point connectors and mathematical objects are used in a project, the danger of accidentally creating update loops increases and the user needs to be aware of this problem.

An update loop means that a single update of one data point causes a chain of following updates which does not end, because one of the following updates again triggers the original update. For example, register A is used as input for a mathematical object which writes its result to register B, while at the same time the value of register B is copied into register A using a data point connector.

The configuration software is able to check for such loops when they are caused entirely by internal objects. In such a case, the project cannot be downloaded into the device and a warning message appears, showing the objects which cause the infinite update loop.

However, the configuration software cannot check loops which are closed via the network, because it does not have enough information to do this. For example, if a data point connector is used to connect a switch input SI to a switch output SO, this in itself does not constitute a loop at all. But when the update of SO on the network is sent to a lamp actuator and the lamp actuators feedback output is connected to the switch input SI, the result may be an infinite update loop between the L-VIS device and the lamp actuator, depending on how the actuator behaves. Once the L-VIS sends out an update on SO, it will be received by the lamp actuator. This may cause the actuator to send out an update on the feedback output, which will be received by L-VIS on the input SI, which in turn will be copied by the data point connector to the output SO and so on. The result can have adverse effects on the network performance and the performance of the devices in the loop, so special care should be taken to avoid such setups.

15.7 XML Support

Similar to the XML import/export function for individual objects, as explained in section 8.5.4, it is possible to read and write the entire project file in XML format. The easiest way to do this is to select XML format from the file format dropdown list in the file save dialog, which opens when the command **Save As...** is selected from the *File* menu of the main window. Alternatively, use the command **Export XML > Project...** from the *File* menu.

Using the sub-menus **Export XML** and **Import XML** from the *File* menu, it is also possible to read or write only parts of the project in XML format. The following parts may be imported and exported separately:

- **Project:** As explained above, reads or writes an entire project.
- **Font Set:** Reads or writes the current set of user loaded fonts. This can be used to save a standard font set in XML format and import it as a starting point when a new project is started. If the font set is imported while there are already controls in the project referencing user fonts, care should be taken that the changed font set is suitable for the existing controls. User fonts are referenced by index number, so that the changed font set may cause controls to be drawn in a font of different size as before. Check the project for problem areas if such a switch is done.
- **Project Settings:** Reads or writes the current project settings, as configured in the project settings dialog. This can also be used to store a default project setup in XML format and import it when a new project is started.

- **Data Point Setup:** Reads or writes the data point configuration, as configured in the data point selection and management dialog. Note that all controls and other objects reference data points through their unique ID. If the data point configuration is replaced while some of the old data points are used by objects in the project, these references may be broken or point to wrong data points. This function should therefore be used with care, for example if both a data point configuration and suitable project pages are imported together.
- **Calendar Config:** Configuration data for a local calendar. Only available if the project contains a local calendar point.
- **Scheduler Config:** Configuration data for all local scheduler units. Only available if the project contains at least one local scheduler.

The main use of XML import and export is to enable external tools to automatically create parts of a project or even a full project, based on XML templates prepared using the export functions. Such a tool could for example browse a network or a database, pick out some devices which should be visualized in an L-VIS project, select suitable templates from an XML template database, create the required data point configuration to allow communication with the desired devices and complete the templates by linking the data point references to the automatically created data points. The resulting XML file could represent an entire L-VIS project, which can be loaded into the configuration software and written to the L-VIS device.

The complete XML schema specification for the individual parts of the configuration, which is required to develop external tools, may be obtained from LOYTEC on request.

15.8 Format strings

This chapter describes the various types of format strings, as well as regular expressions, which are used by the configuration software.

15.8.1 Number format strings

The format string used in text controls and **non-localized** numeric controls is equivalent to the format strings used by the ANSI-C printf function. L-VIS will always use the given format string to format exactly **one argument**, which will be a string, an integer, or a double precision floating point number, depending on the type of control and its configuration. Therefore, there should always be exactly **one conversion specifier** in an L-VIS format string and its type needs to match the argument provided by the control. For example, it will not work to use a %d specifier for a text control or a %s specifier for a numeric control.

The format string is composed of zero or more directives: ordinary characters (not %), which are copied unchanged to the output stream; and conversion specifications, each of which results in fetching zero or more subsequent arguments. Each conversion specification is introduced by the character %, and ends with a *conversion specifier*. In between there may be (in this order) zero or more *flags*, an optional minimum *field width*, an optional *precision* and an optional *length modifier*. For some numeric conversions a radix character ('decimal point') or thousands' grouping character is used. The actual character used depends on the implementation. Non-localized number controls always use '.' as radix character, and do not use grouping.

15.8.1.1 Flags

The character % is followed by zero or more of the following flags:

The value should be converted to an "alternate form". For **o** conversions, the first character of the output string is made zero (by prefixing a 0 if it was not zero already). For

x and **X** conversions, a nonzero result has the string '0x' (or '0X' for **X** conversions) prepended to it. For **a**, **A**, **e**, **E**, **f**, **F**, **g**, and **G** conversions, the result will always contain a decimal point, even if no digits follow it (normally, a decimal point appears in the results of those conversions only if a digit follows). For **g** and **G** conversions, trailing zeros are not removed from the result as they would otherwise be. For other conversions, the result is undefined.

0 The value should be zero padded. For **d**, **i**, **o**, **u**, **x**, **X**, **a**, **A**, **e**, **E**, **f**, **F**, **g**, and **G** conversions, the converted value is padded on the left with zeros rather than blanks. If the **0** and **-** flags both appear, the **0** flag is ignored. If a precision is given with a numeric conversion (**d**, **i**, **o**, **u**, **x**, and **X**), the **0** flag is ignored. For other conversions, the behavior is undefined.

- The converted value is to be left adjusted on the field boundary. (The default is right justification.) Except for **n** conversions, the converted value is padded on the right with blanks, rather than on the left with blanks or zeros. A **-** overrides a **0** if both are given.

' ' (a space) A blank should be left before a positive number (or empty string) produced by a signed conversion.

+ A sign (+ or -) always be placed before a number produced by a signed conversion. By default a sign is used only for negative numbers. A **+** overrides a space if both are used.

15.8.1.2 Field Width

An optional decimal digit string (with nonzero first digit) specifying a minimum field width. If the converted value has fewer characters than the field width, it will be padded with spaces on the left (or right, if the left-adjustment flag has been given). A negative field width is taken as a **'-'** flag followed by a positive field width. In no case does a non-existent or small field width cause truncation of a field; if the result of a conversion is wider than the field width, the field is expanded to contain the conversion result.

15.8.1.3 Precision

An optional precision, in the form of a period ('.') followed by an optional decimal digit string. If the precision is given as just '.', or the precision is negative, the precision is taken to be zero. This gives the minimum number of digits to appear for **d**, **i**, **o**, **u**, **x**, and **X** conversions, the number of digits to appear after the radix character for **a**, **A**, **e**, **E**, **f**, and **F** conversions, the maximum number of significant digits for **g** and **G** conversions, or the maximum number of characters to be printed from a string for **s** and **S** conversions.

15.8.1.4 Length Modifier

The conversion specifier may normally be preceded by a length modifier, defining the physical size of the arguments data type. In the L-VIS implementation, this modifier is not necessary and may be omitted in all cases.

15.8.1.5 Conversion Specifier

This is the character that specifies the type of conversion to be applied. The available conversion specifiers and their meanings are:

d,i The *int* argument is converted to signed decimal notation. The precision, if any, gives the minimum number of digits that must appear; if the converted value requires fewer digits, it is padded on the left with zeros. The default precision is 1. When 0 is printed with an explicit precision 0, the output is empty.

o,u,x,X The *unsigned int* argument is converted to unsigned octal (**o**), unsigned decimal (**u**), or unsigned hexadecimal (**x** and **X**) notation. The letters **abcdef** are used for **x** conversions; the letters **ABCDEF** are used for **X** conversions. The precision, if any, gives the minimum number of digits that must appear; if the converted value

requires fewer digits, it is padded on the left with zeros. The default precision is 1. When 0 is printed with an explicit precision 0, the output is empty.

- e,E** The *double* argument is rounded and converted in the style `[-]d.ddde±dd` where there is one digit before the decimal point character and the number of digits after it is equal to the precision; if the precision is missing, it is taken as 6; if the precision is zero, no decimal point character appears. An **E** conversion uses the letter **E** (rather than **e**) to introduce the exponent. The exponent always contains at least two digits; if the value is zero, the exponent is 00.
- f,F** The *double* argument is rounded and converted to decimal notation in the style `[-]ddd.ddd`, where the number of digits after the decimal point character is equal to the precision specification. If the precision is missing, it is taken as 6; if the precision is explicitly zero, no decimal point character appears. If a decimal point appears, at least one digit appears before it.
- g,G** The *double* argument is converted in style **f** or **e** (or **F** or **E** for **G** conversions). The precision specifies the number of significant digits. If the precision is missing, 6 digits are given; if the precision is zero, it is treated as 1. Style **e** is used if the exponent from its conversion is less than -4 or greater than or equal to the precision. Trailing zeros are removed from the fractional part of the result; a decimal point appears only if it is followed by at least one digit.
- s** The argument is expected to be a text string. If a precision is specified, no more characters than the number specified are written.
- %** A `'%'` is written. No argument is converted in the process.

15.8.2 Date format strings

The format string used by **non-localized** date controls is equivalent to the format string used by the ANSI-C function `strftime`. The locale setting for the purpose of formatting dates is fixed to U.S. in this case, therefore the names of weekdays and months will be in English language only. Use localized formatting (if available), to show localized dates.

Conversion specifiers are introduced by a `'%'` character, and are replaced as follows:

- %a** The abbreviated weekday name.
- %A** The full weekday name.
- %b** The abbreviated month name.
- %B** The full month name.
- %c** The date and time in American writing.
- %C** The century number (year/100) as a 2-digit integer. (SU)
- %C** The century number (the year divided by 100 and truncated to an integer).
- %d** The day of the month as a decimal number (range 01 to 31).
- %D** Equivalent to `%m/%d/%y`.
- %e** Like `%d`, the day of the month as a decimal number, but a leading zero is replaced by a space.
- %E** Modifier: use alternative format, see below.

- %G** The ISO 8601 year with century as a decimal number. The 4-digit year corresponding to the ISO week number (see %V). This has the same format and value as %y, except that if the ISO week number belongs to the previous or next year, that year is used instead.
- %g** Like %G, but without century, i.e., with a 2-digit year (00-99).
- %h** Equivalent to %b.
- %H** The hour as a decimal number using a 24-hour clock (range 00 to 23).
- %I** The hour as a decimal number using a 12-hour clock (range 01 to 12).
- %j** The day of the year as a decimal number (range 001 to 366).
- %k** The hour (24-hour clock) as a decimal number (range 0 to 23); single digits are preceded by a blank. (See also %H.)
- %l** The hour (12-hour clock) as a decimal number (range 1 to 12); single digits are preceded by a blank. (See also %I.)
- %m** The month as a decimal number (range 01 to 12).
- %M** The minute as a decimal number (range 00 to 59).
- %n** A newline character..
- %O** Modifier: use alternative format, see below.
- %p** Either 'AM' or 'PM' according to the given time value, or the corresponding strings for the current locale. Noon is treated as 'pm' and midnight as 'am'.
- %P** Like %p but in lowercase: 'am' or 'pm' or a corresponding string for the current locale
- %r** The time in a.m. or p.m. notation. In the POSIX locale this is equivalent to '%I:%M:%S %p'
- %R** The time in 24-hour notation (%H:%M). (SU) For a version including the seconds, see %T below.
- %s** The number of seconds since the Epoch, i.e., since 1970-01-01 00:00:00 UTC.
- %S** The second as a decimal number (range 00 to 61).
- %t** A tab character.
- %T** The time in 24-hour notation (%H:%M:%S).
- %u** The day of the week as a decimal, range 1 to 7, Monday being 1. See also %w.
- %U** The week number of the current year as a decimal number, range 00 to 53, starting with the first Sunday as the first day of week 01. See also %V and %W.
- %V** The ISO 8601:1988 week number of the current year as a decimal number, range 01 to 53, where week 1 is the first week that has at least 4 days in the current year, and with Monday as the first day of the week. See also %U and %W.
- %w** The day of the week as a decimal, range 0 to 6, Sunday being 0. See also %u.

- %W** The week number of the current year as a decimal number, range 00 to 53, starting with the first Monday as the first day of week 01.
- %x** The preferred date representation for the current locale without the time.
- %X** The preferred time representation for the current locale without the date.
- %y** The year as a decimal number without a century (range 00 to 99).
- %Y** The year as a decimal number including the century.
- %z** The time-zone as hour offset from GMT. Required to emit RFC822-conformant dates (using “%a, %d %b %Y %H:%M:%S %z”).
- %Z** The time zone or name or abbreviation.
- %+** The date and time in date(1) format.
- %%** A literal ‘%’ character.

Some conversion specifiers can be modified by preceding them by the E or O modifier to indicate that an alternative format should be used. If the alternative format or specification does not exist for the current locale, the behavior will be as if the unmodified conversion specification were used. (SU) The Single Unix Specification mentions %Ec, %EC, %Ex, %EX, %Ry, %EY, %Od, %Oe, %OH, %OI, %Om, %OM, %OS, %Ou, %OU, %OV, %Ow, %OW, %Oy, where the effect of the O modifier is to use alternative numeric symbols (say, roman numerals), and that of the E modifier is to use a locale-dependent alternative representation.

15.8.3 Regular Expressions

A regular expression is a pattern that describes a set of strings. Regular expressions are constructed analogously to arithmetic expressions, by using various operators to combine smaller expressions.

15.8.3.1 Syntax

The fundamental building blocks are the regular expressions that match a single character. *Most characters, including all letters and digits, are regular expressions that match themselves.* Any meta-character with special meaning may be quoted by preceding it with a backslash.

A *bracket expression* is a list of characters enclosed by [and]. It matches any single character in that list; if the first character of the list is the caret ^ then it matches any character not in the list. For example, the regular expression [0123456789] matches any single digit.

Within a bracket expression, a *range expression* consists of two characters separated by a hyphen. It matches any single character that sorts between the two characters, inclusive, using the locale’s collating sequence and character set. For example, in the English locale, [a–d] is equivalent to [abcd].

Most meta-characters lose their special meaning inside lists, but some characters must be placed at specific positions, as pointed out below:

- To include a literal] place it first in the list.
- To include a literal ^ place it anywhere but first.
- To include a literal – place it last.

The following special characters and symbols exist:

- The period `.` matches any single character.
- The symbol `\w` is a synonym for any alphanumeric character and `\W` is a synonym for any non-alphanumeric character.
- The caret `^` and the dollar sign `$` are meta-characters that respectively match the empty string at the beginning and end of a line.
- The symbols `\<` and `\>` respectively match the empty string at the beginning and end of a word.
- The symbol `\b` matches the empty string at the edge of a word, and `\B` matches the empty string provided it's not at the edge of a word.

A regular expression may be followed by one of several *repetition* operators:

- ? The preceding item is optional and matched at most once.
 - *
 - +
- The preceding item will be matched zero or more times.
- The preceding item will be matched one or more times.

Two regular expressions may be *concatenated*; the resulting regular expression matches any string formed by concatenating two substrings that respectively match the concatenated sub-expressions.

Two regular expressions may be *joined* by the infix operator `|`; the resulting regular expression matches any string matching either sub-expression.

Repetition takes precedence over concatenation, which in turn takes precedence over alternation. A whole sub-expression may be enclosed in parentheses to override these precedence rules.

The back-reference `\n`, where `n` is a single digit, matches the substring previously matched by the `nth` sub-expression enclosed in `{ }`.

15.8.3.2 Ambiguity

If a regular expression could match two different parts of the input string, it will match the one which begins earliest. If both begin in the same place but match different lengths, or match the same length in different ways, life gets messier, as follows.

In general, the possibilities in a list of branches are considered in left-to-right order, the possibilities for `*`, `+`, and `?` are considered longest-first, nested constructs are considered from the outermost in, and concatenated constructs are considered leftmost-first. The match that will be chosen is the one that uses the earliest possibility in the first choice that has to be made.

If there is more than one choice, the next will be made in the same manner (earliest possibility) subject to the decision on the first choice, and so forth.

For example, `(ab|a)b*c` could match `abc` in one of two ways. The first choice is between `ab` and `a`; since `ab` is earlier, and does lead to a successful overall match, it is chosen. Since the `b` is already spoken for, the `b*` must match its last possibility – the empty string – since it must respect the earlier choice. In the particular case where no `|` operators are present and there is only one `*`, `+`, or `?`, the net effect is that the longest possible match will be chosen. So `ab*`, presented with `xabbbby`, will match `abbbb`. Note that if `ab*` is

tried against **xabyabbbz**, it will match **ab** just after **x**, due to the begins-earliest rule. In effect, the decision on where to start the match is the first choice to be made; hence subsequent choices must respect it even if this leads them to less-preferred alternatives.

15.9 External Storage

Some L-Vis devices are equipped with a micro SD card slot and/or USB host ports to connect external storage media. Once the memory device is connected, it can be managed via the web page of the device. The section *Config->Removable Media* contains all options to mount and eject memory devices, allocate swap space, or write a backup of the internal trend data to the external memory. Note that allocating or removing swap space will require a reboot of the device to take effect.

The built-in setup menu of the device contains a page called *Storage* which provides buttons to save and restore complete device backups or project files, and to copy internal trend data to the external memory. These buttons will automatically mount the media if needed, execute the command, and eject the media again if it was mounted automatically (not already mounted manually before).

New storage media normally comes preformatted and is suitable for immediate use with L-Vis devices and PCs running the Windows operating system. If the media needs to be formatted manually, create one primary partition and format it using standard settings. L-Vis will mount the first primary partition found on the media.

NOTE:	<i>A maximum of 64 MB of swap space may be allocated on an external storage media to extend the available memory of the device and allow larger projects to be loaded. Note that the first boot after a new swap space was activated will take some time to set up the memory. Subsequent reboots of the device will not be affected. Also note that removing the media during run time while a swap space is active will cause the device to malfunction.</i>
--------------	--

15.10 Security Hardening Guide

This guide contains security-relevant information for operating the product on IT networks. For devices operated solely on a field bus network like FT-10 without any connection to a TCP/IP network, most of the following steps do not apply.

15.10.1 Device Installation

To run the device with a minimum set of enabled services and maximum security, follow the steps below during device installation and configuration:

- In the project settings, set up a PIN code for access level 15 and either check the option *Lock pages in setup menu* or *Hide setup menu*.
- Download the final project to the device and test it (access via the configuration software will be disabled later on in the process).
- Open the Web UI of the device and change the default admin password.
- Browse to the port configuration, open the *Ethernet* tab and disable FTP, Telnet, SSH, and possibly RNI (LVIS-3E1xx) or BACnet/IP (LVIS-ME2xx) services. On devices running newer firmware, you will find FTP and Telnet protocols already disabled by default.

- On the installed device, open the command page of the setup menu and execute the command *Disable HTTP Server*. The device will reboot with the web service disabled. Note that this will also disable the OPC XML-DA server.

NOTE:

On devices equipped with two Ethernet ports, it is possible to configure the ports as two separate and isolated network interfaces (LAN/WAN). In this setup, it may be acceptable to run some less secure IP based protocols on the LAN port, which is connected to an isolated building network not reachable from the internet, whereas services running on the WAN port are limited to the absolute minimum, possibly HTTPS only.

15.10.2 Ports

This Section lists all ports which may be used by the device. The port numbers listed here are default settings for their respective services. If not stated otherwise, the port numbers can be changed.

- 21 tcp: This port is opened by the FTP server. The port can be changed and disabled. FTP service is disabled in factory default settings.
- 22 tcp: This port is opened by the SSH server. The port can be changed and disabled.
- 23 tcp: This port is opened by the Telnet server. The port can be changed and disabled. Telnet service is disabled in factory default settings.
- 80 tcp: This port is opened by the Web server and the OPC XML-DA server. It can be disabled if OPC XML-DA is not required. The port can be changed.
- 1628 udp/tcp: This is the data exchange port for CEA-852 (LON over IP). It is required for global data point connections and possibly for control network communication if the device supports and uses the IP-852 interface. The port can be changed.
- 1629 udp/tcp: This is the configuration server port of CEA-852. Exactly one device in the system (the configuration server) needs this port open. Other devices register with the configuration server to form the IP-852 channel list. The port can be changed and disabled. This port is disabled by default (as long as the IP-852 configuration server is not enabled).
- 1630 udp/tcp: Only models with active FT-10 interface (IP-852 disabled). This port is used by the CEA-709 RNI and for the remote LPA. The port can be changed and disabled.
- 5900 tcp: This port is opened by the VNC server, if enabled in the project settings. The port can be changed and disabled. This port is disabled by default.
- 47808 udp: This is the data exchange port for BACnet/IP. It is required for the primary function of the device only on BACnet models running on BACnet/IP (not MS/TP). The port can be changed and disabled.

15.10.3 Services

The following is a list of services available on the device:

- Devices with IP-852 interface enabled, as well as any other device using global connections: CEA-852 (LON over IP). This service is in accordance with the standard ANSI/CEA-852-B.
- Devices with BACnet/IP enabled: BACnet/IP. This service is in accordance with the standard ANSI/ASHRAE 135-2010.
- HTTP: Web server. It provides a Web-based configuration UI. The Web UI can be disabled after setting up the device.
- OPC XML-DA: This Web service provides access to data points over the OPC XML-DA standard. It is available only when the HTTP server is enabled.

- SSH: Secure shell server, not available on all devices. It can be used for a secure connection to the device by the configuration software for configuration and firmware upgrade.
- FTP and Telnet: FTP and Telnet services were used by old firmware versions for configuration download, firmware upgrade, and access to the log files. These services are now considered legacy and are disabled by default. They were replaced by web service connections via HTTP or HTTPS.
- VNC: The VNC server can be used for remote access to the LCD display on devices where it is enabled in the project settings. The service is disabled by default.
- RNI: This service provides the remote network interface (RNI) function on devices using the FT-10 interface. It is also used by the remote LPA feature. If these features are not needed, the service can be disabled.

15.10.4 Logging and Auditing

The device contains a log file which can be uploaded by the configuration software, or viewed live on the Web-UI. This log contains information about when the device started and when crucial communication errors occur. Other events such as user log-on are logged in a separate operating system log, which is available via the Web-UI.

Events recorded in the system log:

- Time of the last power-on reset of the device.
- Time and version of the last firmware upgrade.
- Time when the device configuration has been cleared or the device was reset to factory defaults.
- Commission of the CEA-709 node.
- Static errors in the device and data point configuration.
- System overload situations as one-time log messages since last power-on.
- Crucial communication errors as they occur.

15.11 Resource Limits

The following table specifies the applicable resource limits for different LVIS device models. Note that some resources have hard limits, while others are basically limited by memory and CPU capacity of the device and/or the bandwidth of the communication channel in use. Depending on the complexity of the project regarding graphics and number of display elements, the type of network interface in use, the required rate of data updates, as well as the number of other devices communicating on the same channel, actual maximum resources which are practically useable may vary significantly. These numbers are therefore shown with a ~ to denote that they are estimates and your mileage may vary.

NOTE:	<i>Slow networks like MS/TP or FT-10 may be overloaded long before the limits specified in this table are reached. It is the responsibility of the project engineer to design the project according to the limits of the network and the communication peers involved.</i>
--------------	--

Resource	Model				
	3E100	ME200	3E11x	ME21x	3Mex
Total number of data points	~5,000	~5,000	~10,000	~10,000	~20,000
OPC Tags	~2,000	~2,000	~10,000	~10,000	~10,000
User Registers	~2,000	~2,000	~5,000	~5,000	~10,000
NVs (static, dynamic)	1,000	n/a	1,000	n/a	1,000
External NVs	~1,000	n/a	~2,000	n/a	~2,000
Alias NVs (ECS and CEA709.1-A mode)	1,000	n/a	1,000	n/a	1,000
Address table entries (15 for CEA709.1-A)	524	n/a	524	n/a	524
LONMARK Calendar objects/patterns	1/100	n/a	1/100	n/a	1/100
LONMARK Scheduler objects	200	n/a	200	n/a	200
LONMARK Alarm Servers	1	n/a	1	n/a	1
BACnet objects (analog, binary, multi-state)	n/a	512	n/a	1000	1000
BACnet client mappings	n/a	~2,000	n/a	~2,000	~2,000
BACnet scheduler objects	n/a	200	n/a	200	200
BACnet calendar objects	n/a	25	n/a	25	25
BACnet notification classes (alarm server)	n/a	32	n/a	32	32
Trend Logs	100	100	512	512	512
Total trended data points	256	256	512	512	512
Total aggregated size	6MB	6MB	60MB	60MB	100MB
E-mail templates	~100	~100	~100	~100	~100
Math objects	~500	~500	~2,000	~2,000	~2,000
Alarm Logs	~20	~20	~100	~100	~100
Connections (local)	~1,000	~1,000	~2,000	~2,000	~2,000
Connections (global)	250	250	250	250	250
L-WEB Clients	20	20	20	20	20
Remote OPC/XML/DA devices	32	32	128	128	128

Table 31: Resource Limits

15.12 Technical Support

LOYTEC offers free telephone and e-mail support for the L-Vis product series. Please contact us at the following address:

LOYTEC electronics GmbH
Blumengasse 35
A-1170 Vienna
Austria / Europe

e-mail : support@loytec.com
Web : <http://www.loytec.com>
tel : +43/1/4020805-100
fax : +43/1/4020805-99

or

LOYTEC Americas Inc.
N27 W23957 Paul Road
Suite 103
Pewaukee, WI 53072
USA

e-mail: support@loytec-americas.com
Web: <http://www.loytec-americas.com>
tel: +1 (512) 402 5319
fax: +1 (262) 408 5238

16 Script Engine

16.1 Overview

The scripting feature on LOYTEC devices that provide it is based on a JavaScript engine and follows a strict event-based model known from the node.js model. The script engine allows users to implement custom protocols using RESTful APIs, JSON, Web services for IoT integration into the LOYTEC data point model. The dpal-js API gives JavaScript modules access to the data point server. Script modules are embedded into and deployed along with the data point configuration.

The device firmware is able to run script modules in a separate process. Each script module is based on a script resource that contains the JavaScript sources and other necessary files and libraries. This concept is similar to constructing an npm package of the node package manager (see www.npmjs.com). A data point API allows the code working with data points.

The *script resource* can be considered as the “program source code”. It is packaged to the data point configuration and executed in the device. Data points in a script resource are referenced programmatically by using path and name, e.g. “/User Registers/reg1” or by browsing the data point folder tree. This results in a fixed assignment in the JavaScript code. Changing the data point references requires changing the code. This style of data point access is beneficial for global modules (e.g. a weather forecast module) or modules that can find all data point by smart browsing algorithms (e.g. MQTT implementation).

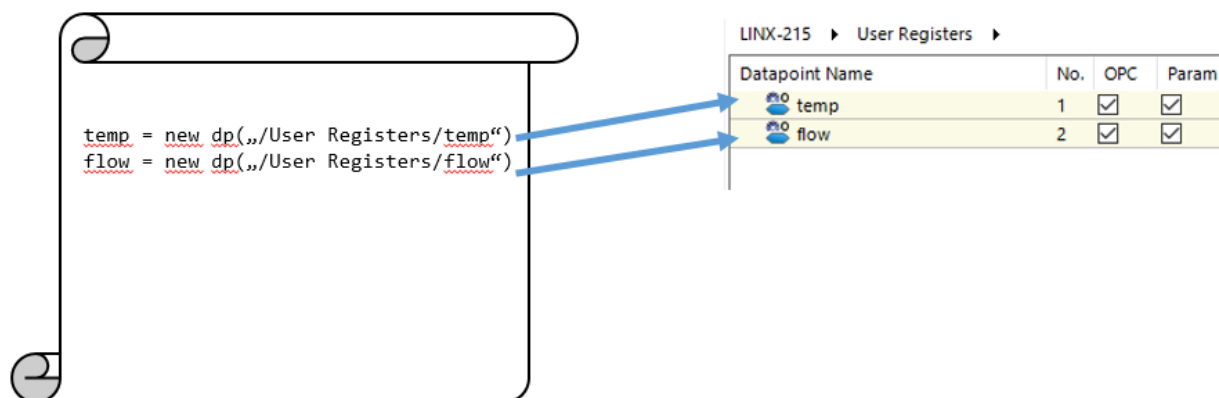


Figure 51: Explicit data point usage in a script resource.

An alternate way for referencing data points are so-called *script objects*. These are configuration objects similar to math objects that contain data point variables. A script object is based on a script resource (i.e. the JavaScript program) that implements a class interface, including an object constructor and some meta-data. The data points referred to in the script code are part of the object constructor and are accessed in the code as v[1], v[2],

etc., instead of using an absolute path. The actual linkage to data points is made by the user who creates the script object in the data point configuration. The concept is depicted in Figure 52.

For each script object, an instance of the JavaScript object implemented by the referred script resource is created. For example, a script resource “myModule.js” implements the code and exposes an object constructor. When creating two script objects Object1 and Object2 in the configuration, the constructor of myModule is called two times, creating two separate instances of the myModule object class. Each one of them with different data point references.

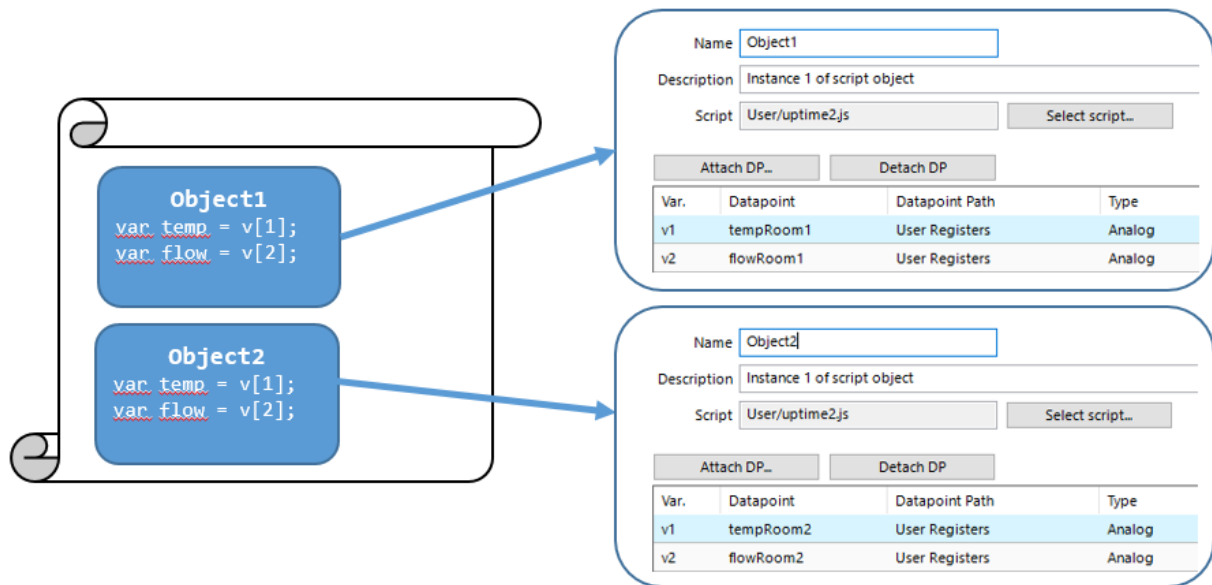


Figure 52: Concept of script objects and object instances.

Referencing data points by using script objects is beneficial for gateway applications, where certain functionality needs to be instantiated multiple times and data points are configured for each remote device. Another application of script objects are library components in L-STUDIO projects.

16.2 Script Resources

The LINX Configurator supports embedding a JavaScript program into the data point configuration. This is called a script resource, very much like an npm module. A separate dialog for managing script resources has been added, that keeps track of embedded scripts. Script resources are imported from the hard drive. There are two basic types of script resources:

- **Script file:** This is a single .js file that contains the JavaScript code in a single unit. Alternatively, a ZIP archive containing an archive of a script folder can be selected.
- **Script folder:** This represents a folder that contains a collection of files which constitutes the script resource. Similar to an npm module a main file has to be appointed. This can be implicit by naming convention (index.js or foldername.js) or explicit by looking into the package.json. The script management dialog reveals the main.js file in the expanded details as shown in Figure 53.

The script resource (folder or ZIP file) may also contain a data point configuration file. The user registers and favorites of that data point configuration are imported along with

the JavaScript program. This is a convenient way to create those data points that are referred to programmatically by the script resource.

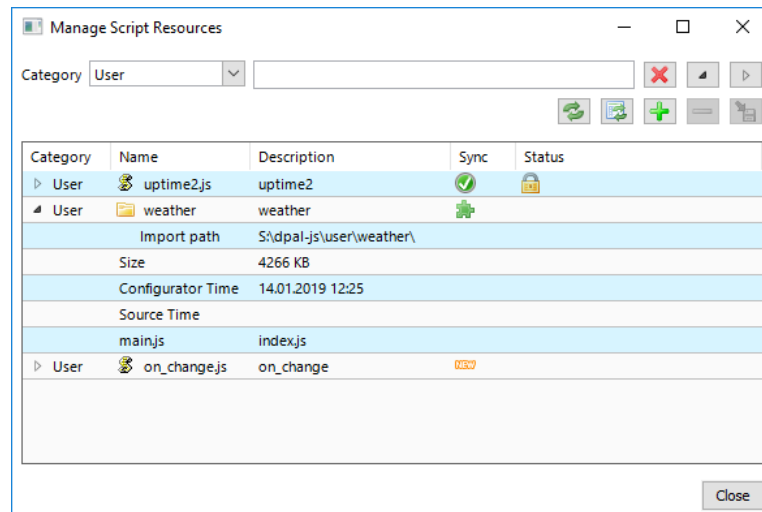


Figure 53: Manage script resources dialog

Once imported, the Configurator keeps track of changes to the script files on the source location. In the development cycle they are typically edited by a JavaScript IDE. When opening the dialog, it scans for newer files on the hard drive and marks them in the dialog as **new**. For library components the sources are typically not locally stored. They are marked by a green jigsaw 🧩.

Script resources managed by this dialog are deployed to the device. Each script resource is then instantiated on the device and executed by the script engine. Typically, a script resource will look for certain data points in the configuration to link its internal data objects to data points.

To Add Script Resources

1. Go to the menu **Tools → Manage Script Resources**.
2. In the Manage Script Resources dialog select a **Category** (e.g. User) and decide which type of script resource shall be added.
3. Click on the plus button and choose either script resource file or folder.
4. If the script resource includes a data point configuration, the user registers and favorites are merged into the existing configuration. A dialog as shown in Figure 54 lists the merged data points and identifies any conflicts. Already existing data points are not overwritten.

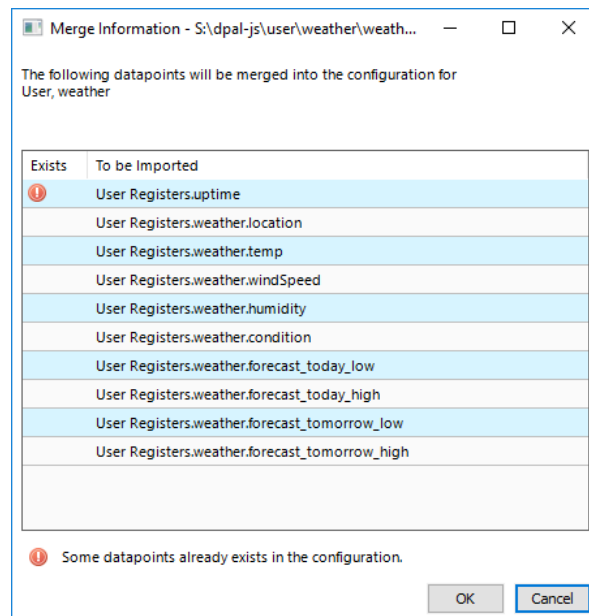











Figure 54: Merging data points defined by script resource

- When editing the script files on the hard drive, the sync status will be refreshed and displays **new** to indicate there is newer content not yet in the configuration. For library components that don't have local sources a green jigsaw  is shown.

Category	Name	Description	Sync	Status
▶ User	 uptime2.js	uptime2		
▶ User	 weather	weather		
▶ User	 on_change.js	on_change		

- To embed the latest changes, select the line and click on the **Update content** button .
- The script content will be downloaded to the device with the next data point configuration download.

16.3 Script Objects

Script objects provide an explicit way to link data points into the script resource. A script object can be instantiated like math objects in the Configurator and refers to an existing file script resource in the configuration. The variables are then linked to data points of the data point configuration. An example is shown in Figure 55.

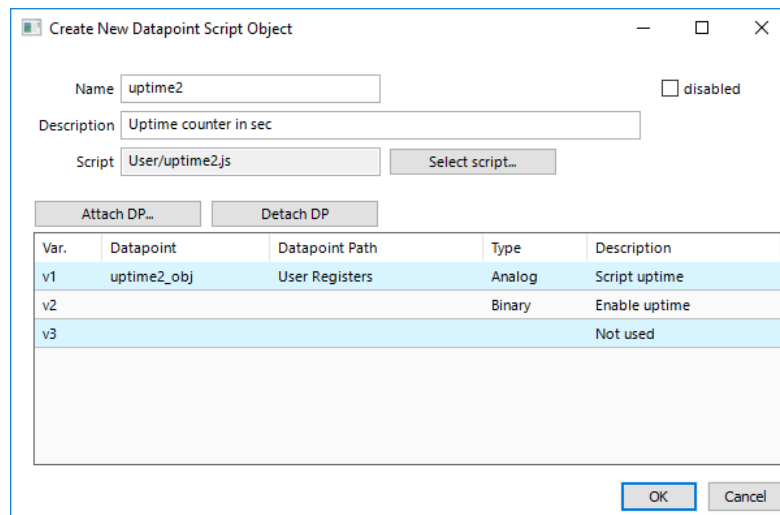
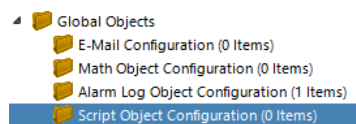


Figure 55: Script object

To Create a Script Object

1. Under the **Global Objects** folder, select the **Script Object Configuration** sub-folder.



2. Right-click and select **New Datapoint Script Object ...** from the context menu.
3. In the **Create New Datapoint Script Object** dialog enter a **Name** and optional **Description**.
4. Choose the file script resource by clicking on **Select Script....**
5. If the file script resource is valid the dialog shows the script's variable references starting v1, v2, etc.

NOTE: *The script resource must be designed to work with script objects, i.e., contain a metaData element that specifies the script variables v1, v2, etc. If this element is missing, the script resource cannot be used in a script object.*

6. Select the variable line and click the button **Attach DP...** to attach a data point to the script variable. Use **Detach DP** to remove a data point reference.

16.4 Developing Scripts

16.4.1 Data Point Integration

One of the primary functions of a script module is interfacing to the data point model on the device. For doing so, a script needs to require the dpal-js module that provides a JavaScript data point API:

```
var dp = require('loytec/dpal-js');
```

Data points can be accessed via data point objects that need to be created by path or UID:

```
var v1 = new dp('/User Registers/uptime');
var v2 = new dp(0x1150);
```

On the data point objects various methods are available to read, get notified or modify the data point value. Use the write method to write values onto a data point:

```
v1.write(20.5);
```

Or get event notifications on new values using the value event:

```
v1.on('value', (data) => { console.log('Value is' + data[0]) });
```

Operations on multiple data points can be done by the dp class methods. Reading multiple values from data points in one sweep is accomplished by calling the dp.readAll() class method:

```
dp.readAll([v1, v2, '/User Registers/someOtherDP'], (err, values) => {  
  console.log('Value is'+values[0]+' '+values[1]+' '+values[2])  
});
```

Another method to get notified about data point updates is to subscribe to all data points that are in a certain folder and its sub-folders:

```
dp.subscribeAll(['/User Registers/']).then((subscription) => {  
  subscription.cov.on('value', (data) => {  
    // iterate over all COVs; the object key represents the data point path  
    for (var key in data) {  
      var value = data[key][0];  
      // do something with the value  
    }  
  });  
});
```

A convenient way to write to multiple data points is the dp.writeAll() class method, that accepts a data point object, path or UID:

```
dp.writeAll([v1, 5], ['/User Registers/xyz', 47], [0x1150, true]);
```

Refer to the example scripts uptime.js, on_change.js and weather.js to explore the basic data point API usage. These script resources can be found in the *Scripts* folder of the Configurator program installation directory.

16.4.2 Using Libraries

Script resources may rely on third-party libraries that implement certain functionalities such as XML decoding or an implementation wrapper of a specific protocol. There are pre-installed library modules on a LOYTEC device that can be readily required in a script file:

- **axios**: Implement RESTful services
- **debug**: Enables debug output for a script
- **xml2js**: XML to JavaScript converter
- **nodejs-websocket**: utility to establish websocket communication

A script resource may also require other library modules. This can be accomplished by creating a script folder resource and add custom libraries local to this folder in its own 'node_modules' sub-folder. This is compatible with installing npm modules locally in the script resource folder.

This is an example: The folder resource 'myProtocol' requires the custom 'express' and 'debug' modules. This is done by installing the 'express' npm locally to the 'myProtocol' folder. This library module will be placed into the 'node_modules/express' sub-folder along with other dependent modules. An example is shown in Figure 56. The entire 'myProtocol' folder is then added as a folder resource to the Configurator.

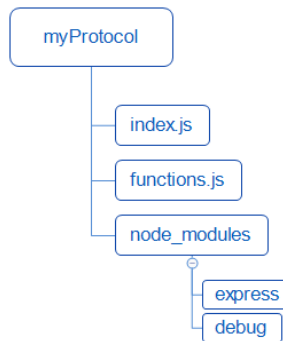


Figure 56: Example for using local libraries

16.4.3 Write a Script Object Resource

Script resources that can be used by data point script objects need to follow certain rules. They must be a single file resource and implement an object constructor of a given style. For an example look at the code snippet below in Figure 57.

```

// script module for data point script object must export a constructor
module.exports = function(dpReferences) {

    const dp = require("loytec/dpal-js");

    // create dp objects from supplied dpReferences
    var v = dp.newVector(dpReferences);

    // uptime of script in seconds
    var uptime=0;
    setInterval(function() { uptime++; v[1].write(uptime); }, 1000);
    console.log("uptime2 script object instantiated");
}

```

Figure 57: Constructor for a data point script object.

The module exports the constructor function(dpReferences). Inside the constructor a data point vector is created by calling dp.newVector(dpReferences). This vector contains the data point access objects v[1], v[2], etc. as seen in the data point script object.

The module also needs to define a metaData object as shown in Figure 58. This object describes the variables used in the script object and their required data point class (e.g. analog or binary). This meta-data is evaluated by the Configurator when instantiating a new data point script object.

```

// meta data evaluated by Configurator for display in data point script dialog
const metaData = {
    "descr" : "Uptime counter in sec",
    "v1": { "descr": "Script uptime", "otype": "analog" },
    "v2": { "descr": "Enable uptime", "otype": "binary" },
    "v3": { "descr": "Not used" }
};

```

Figure 58: Script object meta-data

Finally, the meta-data object needs to be added to the module exports at the end of the file:

```

/* export metaData */
module.exports.metaData = metaData;

```

16.4.4 Open Ports in the Firewall

A script module may need to implement a service that runs under a specific TCP or UDP port, e.g. an HTTP server on TCP port 4000. This port is normally blocked by the firewall on the LOYTEC device. In order to open a port use the system module and the following code pattern:

```
const system = require('/loytec/system');
// allow a port in the firewall
system.firewallAllow({ protocol: 'tcp', port: 4000 }).then((rule) => {
  console.log(`opened port ${rule.port} in firewall`);
}).catch((e) => {
  console.log(e);
});
```

16.4.5 Prototyping

Normally, script resources are deployed by the Configurator software. This is a streamlined solution for the end user but not very practical when prototyping a script implementation. To do a simple test of a script resource, the script files can be placed on the device as *unmanaged* scripts.

Managed script resources are located in a sub-directory named just like the scope name, e.g. “User” in ‘/var/lib/node_app/User’ on the device. This folder is managed by the Configurator and its contents will be overwritten with each deployment. A prototype script can be placed into the unmanaged folder ‘/var/lib/node_app/unmanaged’ using a file manager that can transfer over SSH (e.g. WinSCP).

If no such tool is available or SSH is blocked by a firewall, one can use the transfer functions of the script configuration Web interface (see Figure 59). The scripts need to be stopped first.

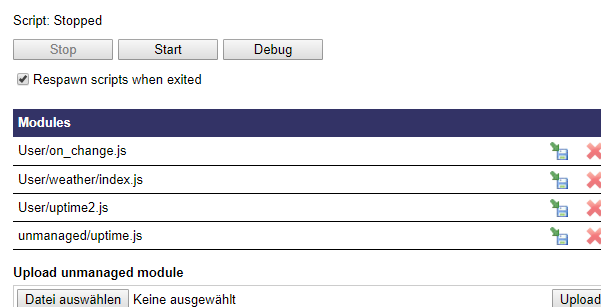




Figure 59: Transfer prototype script resources on the Web interface.

Click the **Upload** button in the **Upload unmanaged module** section to load an unmanaged script resource onto the device. If the script is a file resource, simply select the .js file. If the script is a folder resource, you need to ZIP the folder first, then upload the .zip file. The folder on the device is named after the .zip filename.

To retrieve the JavaScript code of a script resource from the Web interface, click on the save to disk button  of the desired module. To delete a script resource from the device click on the delete button .

16.4.6 Debugging

Script resources on the LOYTEC device are automatically started when the device boots up. In order to debug a script, it is necessary to stop the running scripts and start them in a debug mode. This can be accomplished on the Web interface on the device. Use the **Stop** button as indicated in Figure 60. Note that the console log persists till the next start.

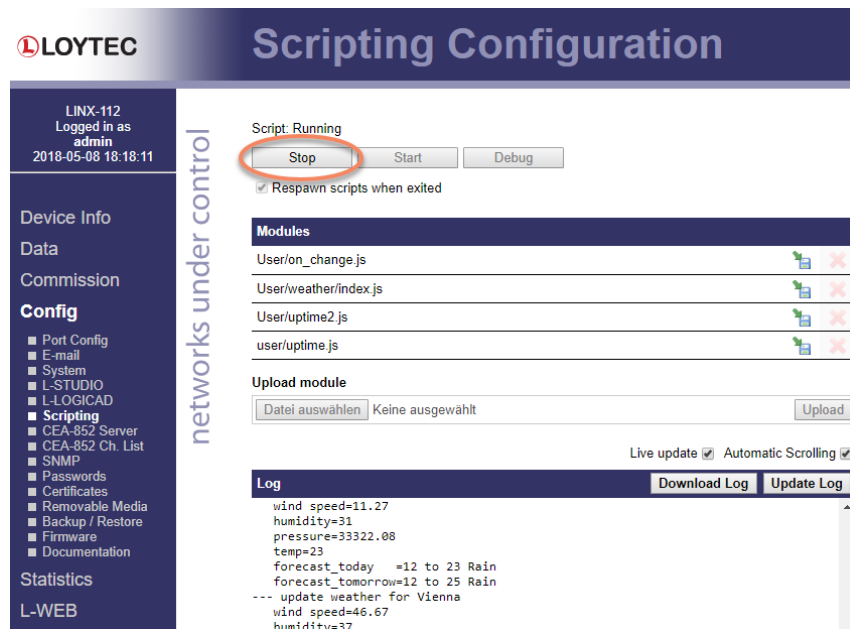


Figure 60: Stop the script and start in debug mode

To debug click the **Debug** button and wait till the script debugger URL for Google Chrome is printed on the Web UI as shown in Figure 61.

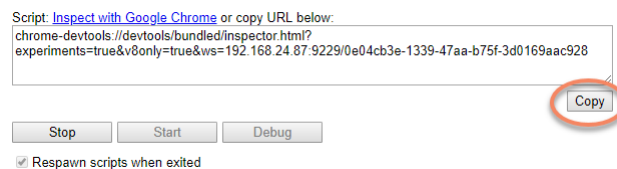



Figure 61: Debug URL for Google Chrome inspector

Copy this URL to Google Chrome and wait for Chrome to connect. Then click the GO button  to start the script. Use Chrome DevTools debugging techniques as for Web pages. When debugging it is useful to turn auto-respawn of scripts off. In this case the root cause of an uncaught exception is easier to find. For doing so, deselect the check box **Respawn scripts when exited**.

Another debugging technique is to use console.log() output and observe the output in the node.js Web UI log window. An example is shown in Figure 62.

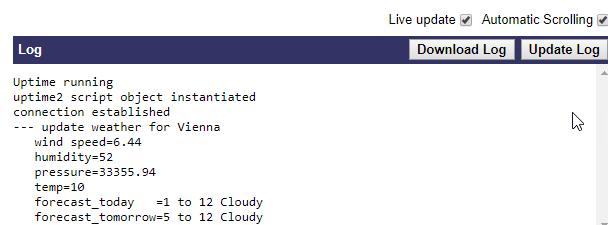


Figure 62: Example console output generated by a script

A more advanced method to output debug or log information that is separated by module is to use the debug module in the script:

```
const debug = require('debug')('dmx');
debug('Starting protocol');
```

The code to print these messages can be left in a production script resource. The output of the messages can be enabled on the script Web interface by specifying the debug keyword, e.g. 'dmx' as shown in Figure 63. The script has to be stopped and the keyword needs to be saved by clicking **Save Settings**.

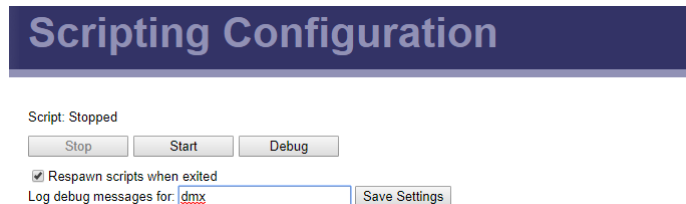


Figure 63: Configure a debug keyword to enable module messages.

16.5 Prototyping on the PC

16.5.1 Set up a Project

For fast prototyping, the script can be developed on the PC and executed on the PC. For doing so, install node.js on the PC as well as a suitable JavaScript IDE. A good example is Visual Studio Code that has a built-in method to start and debug node.js scripts within the IDE.

Since the script will eventually be running on the LOYTEC device, a number of things need to be considered. We assume the development directory on the PC is 'C:\work' and a LOYTEC device exists under the IP 192.168.2.150. For prototyping on the PC, explicit data point access via data point path or by browsing is mandatory and a matching data point configuration is loaded onto the device. Instances via script objects are not possible, because the data point configuration is not local on the PC.

To Create a Prototype Project

1. Set up the pre-installed device libraries: Copy the libraries from the LOYTEC device '/var/lib/node_modules' to 'C:\work\node_modules' using WinSCP.
2. Create a folder 'C:\work\user' to hold your developed script resources. In that user directory create a file called 'dpal-host.json' and edit it to define your device connection (assuming the default operator password):

```
{ "host": "192.168.2.150", "port": "443" }
```

3. Create a module directory, e.g. 'C:\work\user\myModule'. In that directory start your script project with an index.js file.
4. Set up the required data point configuration on the LOYTEC device.
5. Start a new Visual Studio Code project from 'C:\work\user\myModule'. Use **File → Open Folder ...** and choose the project folder.
6. Install any npm packages required by your project into the myModule directory, resulting in a 'C:\work\user\myModule\node_modules' directory.

Eventually the folder 'C:\work\user\myModule' can be zipped into an archive 'myModule.zip'. That archive is ready to be imported as a script resource in the LINX Configurator for distribution.


16.5.2 Setting up an Attach Configuration in Visual Studio Code

Visual Studio Code provides an excellent environment for developing node.js applications on the PC. In order to start a debug session in your project, an attach configuration needs to be added.

The attach option requires more work compared to an auto-attach configuration but allows configuring various debug configuration options explicitly on the command line. The simplest "attach" configuration looks like this:

```
{
  "name": "Attach to Process",
  "type": "node",
  "request": "attach",
  "port": 9229
}
```

The port 9229 is the default debug port of the `--inspect-brk` command line options. To use a different port (e.g. 12345), add it to the options like this: `--inspect-brk=12345` and change the port attribute in the launch configuration accordingly.

Then start the node application in the Visual Studio Code terminal: `node --inspect-brk index.js` as indicated in Figure 64. The node app is then launched and waits for the debugger to connect. Then attach Visual Studio Code to the process using the attach configuration by clicking on .

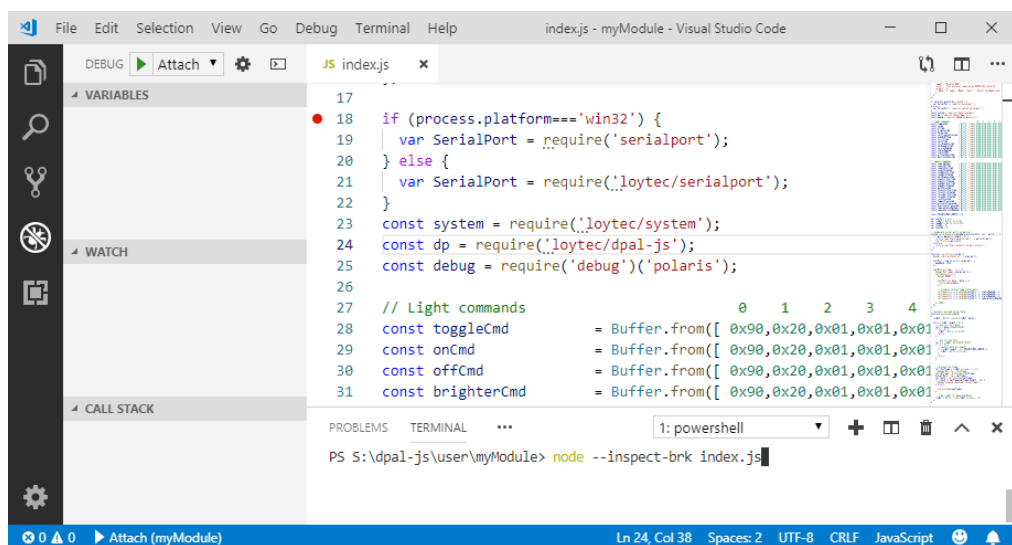


Figure 64: Launch a node app in Visual Studio Code

For a full reference on how to debug node.js apps please refer to the Visual Studio Code documentation Chapter NODE.JS/JAVASCRIPT, Section Node.js Debugging.

17 List of Figures

Figure 1: Connecting L-VIS to the wall mount frame.	42
Figure 2: Main window of the L-VIS configuration software	68
Figure 3: Example of a structured data point.	195
Figure 4: Example of property relations.	195
Figure 5: Multi-slot connection (a) and multi-slot with math block adapter (b).	198
Figure 6: Example of a recurring event in a schedule.	201
Figure 7: Example with overlapping events and different priorities.	202
Figure 8: Example historic filters for daily consumption.	205
Figure 9: Example value presets in CEA-709 schedulers.	206
Figure 10: Example value presets in BACnet schedulers.	208
Figure 11: Data point usage report.	218
Figure 12: Select multistate maps in the multistate map manager.	219
Figure 13: Manage relations tab.	220
Figure 14: Manage favorites tab.	221
Figure 15: Connection tab with a connection and data points.	224
Figure 16: Choose a custom conversion.	224
Figure 17: Edit an analog to multi-state conversion.	225
Figure 18: Create a connection with drag-and-drop.	225
Figure 19: Modify connections in the properties view.	226
Figure 20: Multi-slot connection.	227
Figure 21: Add data point to connection slot.	228
Figure 22: Multi-slot connection for math block adapter.	228
Figure 23: Edit a math block adapter.	229
Figure 24: Created new multi-slot connection from math adapter.	230
Figure 25: Create New Calendar dialog box.	234
Figure 26: Configure Calendar Pattern Dialog.	239
Figure 27: Create New Alarm Server dialog box.	244
Figure 28: Common settings for an alarm condition.	245
Figure 29: Condition for an Analog Feedback Alarm.	246
Figure 30: Alarm Condition for an Analog Value Condition.	246
Figure 31: Alarm Condition for a Binary Data Point.	246
Figure 32: Alarm Condition for a Multi-State Data Point.	246
Figure 33: Spreadsheet for multi-edit of alarm messages.	248
Figure 34: Basic Trend Object Configuration.	249
Figure 35: Dialog for adding/removing BACnet properties.	261
Figure 36: Configuration Tab for Configuration Properties in NL220.	292
Figure 37: Set Configuration Properties in LonMaker TE.	292
Figure 38: Create a function block.	294
Figure 39: Choose network variable dialog.	294
Figure 40: Create network variable dialog.	295
Figure 41: Select complementary variable dialog.	295
Figure 42: Create network variable dialog.	296
Figure 43: New dynamic network variable.	296
Figure 44: Add network variable dialog.	297
Figure 45: Network variable connection window.	297
Figure 46: Alex main window.	298

Figure 47: Virtuelle Funktionseinheit dialog	298
Figure 48: Create new network variable	299
Figure 49: New network variable dialog	299
Figure 50: Finished creation of new network variable	300
Figure 51: Explicit data point usage in a script resource	326
Figure 52: Concept of script objects and object instances.	327
Figure 53: Manage script resources dialog	328
Figure 54: Merging data points defined by script resource	329
Figure 55: Script object	330
Figure 56: Example for using local libraries	332
Figure 57: Constructor for a data point script object	332
Figure 58: Script object meta-data	332
Figure 59: Transfer prototype script resources on the Web interface.	333
Figure 60: Stop the script and start in debug mode	334
Figure 61: Debug URL for Google Chrome inspector	334
Figure 62: Example console output generated by a script	334
Figure 63: Configure a debug keyword to enable module messages.	335
Figure 64: Launch a node app in Visual Studio Code	336
Table 1: L-Vis Device Models	18
Table 2: Electrical characteristics 3E100/ME200	43
Table 3: Electrical characteristics 3E11x/ME21x	44
Table 4: Electrical characteristics 3E11x/ME21x	45
Table 5: Power consumption LVIS-3MEx	45
Table 6: Terminals and Jumpers	47
Table 7: LEDs and Buttons	48
Table 8: Available interfaces	51
Table 9: Usage of the common text field for visible objects	98
Table 10: Usage of the common bitmap for visible objects	99
Table 11: Color assignment for menus	103
Table 12: Function modes of a menu item	104
Table 13: Color assignment for menu items	104
Table 14: Color assignment for text controls	106
Table 15: Color assignment for bitmap controls	108
Table 16: Color assignment for bar controls	114
Table 17: Color assignment for trend controls	120
Table 18: Color assignment for trend curves	120
Table 19: Color assignment for date controls	128
Table 20: Color assignment for data log controls	133
Table 21: Color assignment for schedule controls	137
Table 22: Color assignment for alarm list controls	141
Table 23: Color assignment for vector drawing controls	143
Table 24: Color assignment for analog meter controls	145
Table 25: Color assignment for knob controls	147
Table 26: Color assignment for pie controls	153
Table 27: Color assignment for webcam controls	155
Table 28: Data point icons	162
Table 29: Icons used for data points in the data point list view	213
Table 30: Color coding used for data points in the data point list view	213
Table 31: Resource Limits	324

18 Revision History

Date	Version	Author	Description
2004-11-03	1.1	NR	Initial revision V1.1
2004-06-04	1.2	NR	New screenshots, added configuration chapter, fixes
2006-01-24	1.3	NR	Changes for L-VIS 3E100
2006-04-04	1.5.1	NR	Added documentation for color mapping, theft protection. Updates after review.
2006-10-13	2.0.0	CZ	Rewrote a large part of the manual to make it more useful (hopefully) based on user feedback. The example and solution sections remain unchanged for now. The new BACnet model LVIS-ME200 is now also covered by this manual.
2006-02-08	2.0.4	CZ	Added new sections about data point management and standard solutions, as well as some smaller improvements and corrections.
2008-05-05	3.1.0	CZ	Updated for Release 3.1.0, also including documentation which is relevant to the 3.0.x series of firmware, which was not included in the last version.
2008-06-26	3.2.0	CZ	Updated for Release 3.2.0. Also incorporated some clarifications and additional sections which were missing from the previous version.
2010-05-12	4.0.2	CZ	Updated for Release 4.0.2, LVIS-3E112, LVIS-3E115, LVIS-ME212, and LVIS-ME215 are added. Additionally LWEB-800 Visualization support.
2011-04-08	4.2.0	CZ	Updated for Release 4.2.0
2012-01-26	4.3.0	CZ	Updated for Release 4.3.0
2013-08-30	4.6.0	CZ	Updated for Release 4.6.0
2014-06-26	5.0.0	CZ	Updated for Release 5.0.0
2015-02-13	5.1.0	CZ	Updated for Release 5.1.0
2017-01-05	6.1.0	CZ	Updated for Release 6.1.0 Removed chapters which are now covered by the LOYTEC device manual. Removed outdated example chapters.
2017-08-29	6.2.0	CZ	Updated for Release 6.2.0 Moved chapter <i>Project Settings</i> forward to position 11. Added new section to introduce the template system.
2019-05-06	7.0.0	CZ	Updated for Release 7.0.0
2020-01-30	7.2.0	CZ	Updated for Release 7.2.0