

L-VISTM

LOYTEC Visualization

User Manual

LOYTEC electronics GmbH



Contact

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

Version 4.5

Document No. 88068517

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	10
1.1	What is L-VIS?	10
1.2	Scope.....	12
2	Getting Started.....	13
2.1	Installing the L-VIS Configuration Application.....	13
2.2	Connecting to the Device	13
2.2.1	CEA-709 Devices	13
2.2.2	BACnet Devices	16
2.3	Quick Start Tutorial CEA-709.....	17
2.3.1	Add the device to the network	17
2.3.2	Create Dynamic NVs (LNS only).....	17
2.3.3	Start the L-VIS Configuration Software	18
2.3.4	Create Static NVs (if no dynamic NVs available)	19
2.3.5	Create Menu Structure and Pages	19
2.3.6	Add Controls to the Page	20
2.3.7	Connect the Data-Points	20
2.3.8	Write the Project to the L-VIS device.....	21
2.3.9	Test the Project	22
2.4	Quick Start Tutorial BACnet.....	22
2.4.1	Add the device to the network	22
2.4.2	Start the L-VIS Configuration Software	23
2.4.3	Create Data Points	23
2.4.4	Create Menu Structure and Pages	24
2.4.5	Add Controls to the Page	25
2.4.6	Connect the Data-Points	25
2.4.7	Write the Project to the L-VIS device.....	26
2.4.8	Test the Project	26
3	Mechanical Installation.....	28
3.1	Dimensions and Mounting.....	28
3.2	Theft Protection.....	28
4	Electrical Installation	29
4.1	Connection diagram.....	29
4.2	Electrical Characteristics LVIS-3E100/ME200.....	29
4.3	Electrical Characteristics LVIS-3E11x/ME21x.....	29
4.4	Touch Panel Cleaning Instructions.....	30
4.5	Terminals and Jumpers	30

4.6	LEDs and Buttons.....	31
5	Interface Configuration	33
5.1	Selecting the Interface	33
5.2	Configuring the IP-852 interface.....	34
6	Operating L-VIS.....	35
6.1	Touch Screen.....	35
6.1.1	Operation.....	35
6.1.2	Calibration	36
6.2	Data Input	37
6.2.1	Simple Schedule Control.....	38
6.2.2	Extended Schedule Control	41
6.3	Setup Menu	46
6.3.1	Custom Setup Menu	48
7	Configuration Software	49
7.1	Main Window.....	49
7.2	Context Menus	49
7.2.1	Add Objects.....	50
7.2.2	Enable/Disable Access Control	50
7.2.3	Cut / Copy / Paste.....	50
7.2.4	XML Export / Import	51
7.2.5	Manage Data Points.....	51
7.2.6	Expand / Fold Tree.....	53
7.2.7	Lock / Unlock Position.....	53
7.2.8	Hide / Show, Hide Contents, Show Contents	53
7.2.9	Add Items from Data Points	54
7.2.10	Add Pages from Data Points.....	54
7.2.11	Update Instances	54
7.2.12	Restore Defaults	54
7.3	Main Menus.....	54
7.4	Tool Bar	54
7.5	Workflow	55
7.6	Common Property Pages	56
7.6.1	General	56
7.6.2	Common Properties	56
7.6.3	Color.....	59
8	Object Description.....	60
8.1	Introduction	60
8.2	Menu	61
8.2.1	Menu Properties	61

8.3	Menu Item.....	61
8.3.1	Menu Item Properties.....	62
8.4	Page	63
8.4.1	Page Properties	63
8.5	Controls.....	64
8.5.1	Text Control.....	64
8.5.2	Bitmap Control	65
8.5.3	Push Button.....	66
8.5.4	Numeric Control	66
8.5.5	Bar Control	68
8.5.6	Trend Control.....	71
8.5.7	Date Control.....	76
8.5.8	Data Log Control	77
8.5.9	Schedule Control.....	78
8.5.10	Alarm-List Control	83
8.5.11	Vector Drawing Control	85
8.6	Data Point	87
8.6.1	Data Point Properties	89
8.7	Mapping Table	91
8.7.1	Mapping Table Properties.....	92
8.8	Color Mapping	93
8.8.1	Color Mapping Properties.....	93
8.9	Action Objects	94
8.9.1	Action Properties	94
8.9.2	Page Links	98
8.10	Collections.....	100
8.10.1	Collection Properties.....	100
8.10.2	Collection Handling	100
8.11	Alarm Generators	101
8.11.1	Alarm Generator Properties	102
8.11.2	Application Hints	102
8.12	Data Point Connectors.....	104
8.13	Mathematic Objects	104
8.13.1	Math Object Properties	105
8.13.2	Application Hints	105
8.13.3	Function List.....	106
8.14	Trigger Objects.....	107
8.14.1	Trigger Object Properties.....	108
8.15	Templates.....	110
8.15.1	Template Properties	110

8.15.2	Creating a Template	110
8.15.3	Instantiating a Template	111
8.15.4	Nested Templates	112
8.15.5	Local Changes	113
8.15.6	Base Path Placeholder Tags.....	114
9	Data Point Management	115
9.1	Concept.....	115
9.2	User Interface.....	116
9.2.1	Folder List	117
9.2.2	Data Object List	118
9.2.3	Configuration	118
9.2.4	Buttons	119
9.3	Object Creation.....	119
9.3.1	User Register.....	119
9.3.2	Local Network Variable (CEA-709)	120
9.3.3	Local Server-Object (BACnet).....	120
9.3.4	Remote Network Object.....	121
9.4	Local Scheduler and Calendar	124
9.4.1	Point Creation.....	124
9.4.2	Calendar Point Configuration.....	124
9.4.3	Scheduler Point Configuration	124
9.4.4	Schedule Configuration Data	125
9.4.5	System Resources.....	126
9.4.6	Resource allocation	127
9.4.7	Using the Scheduler.....	128
9.5	Local Alarm Server	128
9.5.1	Intrinsic Reporting.....	128
9.5.2	Algorithmic Reporting	128
9.5.3	System Resources.....	129
9.5.4	Using Alarm Servers	129
9.6	Local Trend	129
9.6.1	Trend Data Object and Trend Controls	129
9.6.2	Trend Mode.....	130
9.7	Remote Scheduler and Calendar	131
9.8	Alarm Clients	132
9.9	E-Mail Templates	133
9.9.1	Common E-Mail Properties.....	133
9.9.2	E-Mail Trigger.....	134
9.9.3	Attachments.....	134
10	Standard Procedures (CEA-709)	135

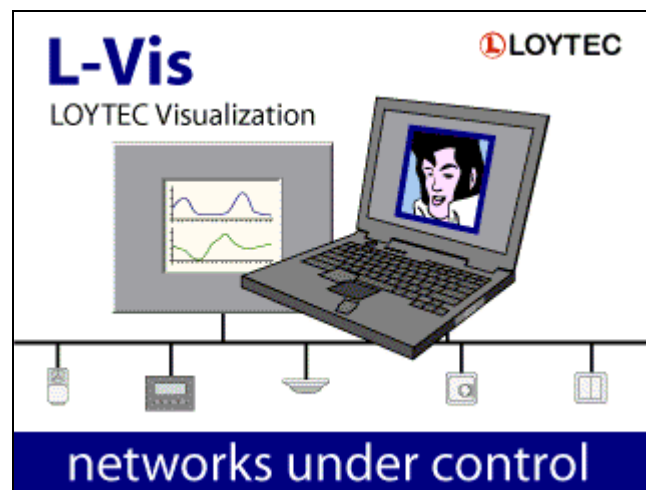
10.1	Device Integration	135
10.1.1	New Device.....	135
10.1.2	Pre-Programmed Device.....	137
10.2	Device Replacement	138
10.2.1	Dynamic NV Projects	138
10.2.2	Static NV Projects.....	138
10.2.3	Mixed Projects (static and dynamic).....	139
10.3	Configuration Change	140
10.3.1	Same NV Configuration	140
10.3.2	Dynamic NV Change.....	140
10.3.3	Static NV Change	141
10.4	Configuration Recovery.....	142
10.4.1	Change Device Configuration.....	142
10.4.2	Change Network Configuration	143
10.4.3	Recover from a failed Update	143
11	Special Functions (CEA-709).....	146
11.1	Binding Management.....	146
11.2	XIF Export.....	146
11.2.1	Offline.....	147
11.2.2	Online	147
11.3	CEA-709.1-A Operation	147
11.4	Dynamic Network Variables	147
11.4.1	LonMaker® Network Management Tool	147
11.4.2	NL220.....	150
11.4.3	Alex.....	151
11.5	Built-in LONMARK® Objects	154
11.5.1	Timer Object.....	154
11.5.2	Switch Object.....	154
11.5.3	Relay / LCD Backlight control	155
11.5.4	HVAC Temperature Sensor	156
12	Special Functions (BACnet).....	157
12.1	EDE Export of BACnet Objects.....	157
12.2	BACnet Network Settings.....	157
13	Project Settings	158
13.1	Basic Settings	158
13.1.1	Page Protection	158
13.1.2	Timeout Values.....	158
13.1.3	Sound	159
13.1.4	OEM Bitmaps	159

13.1.5 System Strings.....	159
13.2 Extended Settings	159
13.2.1 System Flags	159
13.2.2 System Settings	160
13.2.3 System Colors	160
13.2.4 Time Zone and DST	160
13.3 Display	161
13.3.1 LCD Backlight	161
13.3.2 Display Properties	161
13.4 Connectivity	161
13.4.1 Remote Display Access.....	161
13.4.2 Remote Data Access (FTP)	162
13.4.3 Default OPC Access.....	162
13.4.4 Auto-Convert to LWEB	162
13.5 Mail Account.....	163
13.6 Info.....	163
14 Advanced Topics.....	165
14.1 Access Control.....	165
14.2 Undo / Redo.....	166
14.3 Layout Tools.....	166
14.3.1 Alignment Functions	166
14.3.2 Make Same Size	166
14.3.3 Space Evenly.....	166
14.4 Find Objects	167
14.5 Find Similar Objects.....	168
14.6 Project Statistics	168
14.7 Automatic Project Builder	169
14.7.1 Options	169
14.7.2 Page Design.....	170
14.7.3 Page Template	171
14.8 Preview and Edit Mode.....	171
14.9 Remote Network Interface.....	172
14.10 Local Data Point Capacity	172
14.11 Device Model and Firmware.....	172
14.12 Avoiding Update Loops.....	173
14.13 User loaded Fonts	174
14.13.1 Font Manager	174
14.13.2 Font File Location	175
14.13.3 Font Recoding	176
14.14 XML Support.....	176

14.15 Using Images.....	177
14.16 Using Sounds	179
14.17 Project Statistics	179
14.18 Format strings	179
14.18.1 Number format strings	180
14.18.2 Date format strings.....	181
14.18.3 Regular Expressions.....	184
14.19 Security Hardening Guide.....	185
14.19.1 Device Installation	185
14.19.2 Ports	186
14.19.3 Services.....	186
14.19.4 Logging and Auditing.....	187
14.20 Resource Limits.....	187
15 Appendix A: Examples.....	189
15.1 Example 1: Working with Structured Network Variables	189
15.1.1 Text control.....	189
15.1.2 Bitmap Control	190
15.1.3 Numeric Control	191
15.2 Example 2: Using Data Point conversion and Hiding of Pages	192
15.3 Example 3: Freeze warning, Alarms and Registers	193
15.4 Example 4: Using text mapping and Date Control.....	195
16 Appendix B: Solutions.....	198
16.1 Implementing Menus	198
16.1.1 Tab Menus	198
16.1.2 Icon Menus	200
16.1.3 Taskbar Menu	201
16.1.4 Bitmap Menu	201
16.2 Light Control	202
16.3 Animated Bitmaps.....	203
16.4 Sunblind Demo	203
16.5 Split Demo.....	204
16.6 Counter Demo.....	205
17 Revision History.....	207

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 one or more LINX automation servers via OPC XML-DA. Available data points on the servers are easily imported for use in the LVIS project from the corresponding LINX project files.
- Support for the UCS-2 character set of the Unicode standard (ISO-10646), to allow for the design of projects in any language, including Chinese, Japanese, and Korean (CJK).
- Display of data as numeric value, user-defined text or graphic elements, horizontal and vertical bars or trend logs. A special element is available to display date and time in form of digital or analog clocks.

- Input of data via a numeric keypad, direct drag of a bar or selection from a dropdown list of texts or graphic elements.
- 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.
- Input voltage: 24V AC or DC

The LVIS-3E100 and LVIS-3E11x models (CEA-709) also provide:

- Up to 1000 network variables and alias variables.
- 524 address table entries.
- Up to 100 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 LVIS-ME200 and LVIS-ME21x models (BACnet) also provide:

- Up to 512 server objects, accepting COV subscriptions. Available object types are analog, binary and multi-state, each as input, output, or value object.
- BACnet/IP or MSTP connectivity (selectable via jumper or 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.

1.2 Scope

This manual covers the L-VIS device family. Currently available are the following models:

Model	Network Type	Display	Colors
LVIS-3E100	CEA-709	5.7" 320x240	256 color VGA
LVIS-100-RE	CEA-709 *	5.7" 320x240	256 color VGA
LVIS-3E112	CEA-709	12" 800x600	65k color ARGB
LVIS-3E115	CEA-709	15" 1024x786	65k color ARGB
LVIS-ME200	BACnet	5.7" 320x240	256 color VGA
LVIS-ME212	BACnet	12" 800x600	65k color ARGB
LVIS-ME215	BACnet	15" 1024x786	65k color ARGB

* The LVIS-100-RE model is the same as the LVIS-3E100, but with a fixed static network interface. Dynamic and external NVs are not supported. A ready-to-go project designed for room automation is pre-installed on the device (the –RE suffix indicates that this device is such a *room edition* device).

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 the LWEB-800 and LWEB-802.

2 Getting Started

2.1 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 the 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.

2.2 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 for BACnet models, it is important to know the advantages and limitations of each method for CEA-709 devices.

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>
--------------	--

2.2.1 CEA-709 Devices

CEA-709 devices may be configured in several different ways:

- Through LNS[®] (software run as LNS[®] plug-in).
- Via a TCP/IP connection (using FTP/Telnet or SSH).
- Via a CEA-709 connection (FT-10 or IP-852).

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.

2.2.1.1 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 2.2.1.2).
- 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.

2.2.1.2 TCP / IP (Ethernet) Connections

This method connects to the device through a standard TCP/IP channel. Because data is transferred using the FTP and telnet protocols, the PC running the configuration software

must be able to establish an FTP and telnet connection to the device. The LVIS-3E11x and LVIS-ME21x devices also support encrypted connections via SSH. In this case, only one TCP port (port 22) is used for the entire communication between the PC and the device.

TCP/IP connections are possible even when the device operates in FT-10 mode, since the IP port is always active. 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.

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 may not be loaded via this method if the device is part of an LNS database, 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 2.2.1.1).
- 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.

2.2.1.3 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.

2.2.2 BACnet Devices

To configure BACnet devices, the software is always run in standalone mode and a connection to the device is made over TCP/IP. The PC must be able to establish an FTP and Telnet connection or a single SSH connection to the device. If NAT routers or firewalls are located between the PC and the device, the SSH protocol is preferred over the FTP/Telnet connection. Check your firewall rules if necessary. Note that SSH connections are not possible for the LVIS-3E100 and LVIS-ME200 devices.

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, use SSH for WAN connections.*

2.3 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).

2.3.1 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.

2.3.1.1 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 6.3).

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.*

2.3.1.2 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.

2.3.2 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 11.4.

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.

2.3.3 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 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>
--------------	---

2.3.4 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 9.3.2 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*.

2.3.5 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 8.2.

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 8.3.

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:

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.

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 8.4.

2.3.6 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.

2.3.7 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 8.14, 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 8.7, 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 14.14 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.

2.3.8 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 10 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.

2.3.9 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.

2.4 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.

2.4.1 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 5.1 (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.

2.4.2 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>
--------------	---

2.4.3 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 9.3.3 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 9.3.4 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.

2.4.4 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 8.2.

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 8.3.

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:

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.

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 8.4.

2.4.5 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.

2.4.6 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 8.14. 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 8.7, 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: *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 14.14 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.

2.4.7 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.

2.4.8 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: *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 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.

3 Mechanical Installation

3.1 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.

3.2 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.

4 Electrical Installation

4.1 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.

4.2 Electrical Characteristics LVIS-3E100/ME200

The electrical characteristics of the LVIS-3E100 and LVIS-ME200 devices are shown in Table 1.

Characteristics	Value
Operating voltage	24 VDC or 24 VAC \pm 10%
Power consumption	8 W (backlight on), 3W (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	CCFL backlight, 10 000 hours / auto off
Installation	Wall mount

Table 1: Electrical characteristics 3E100/ME200

4.3 Electrical Characteristics LVIS-3E11x/ME21x

The electrical characteristics of the LVIS-3E11x and LVIS-ME21x devices are shown in Table 2.

Characteristics	Value
Operating voltage	24 VDC \pm 10%, or with 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	CCFL backlight, 50 000 hours / auto off
Installation	Wall mount

Table 2: Electrical characteristics 3E11x/ME21x

4.4 Touch Panel Cleaning Instructions

Clean and soft clothes with neutral detergent or with ethanol may be used for cleaning. **Do not use any chemical solvent, acidic or alkali solution.**

4.5 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 1. 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 12 inch and 15 inch devices have two Ethernet ports which are internally connected through a switch.
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 3: Terminals and Jumpers

4.6 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 4: LEDs and Buttons

5 Interface Configuration

5.1 Selecting the Interface

Both the CEA-709 units and the BACnet units have two network interfaces and two different types of communication going on, which should be clearly distinguished:

- Communications with the configuration software, to read and write configuration data, upgrade the firmware, and retrieve diagnostic data from the device (configuration).
- Communication with other devices on the control network, to send out new data the user entered through the device and to receive new data which is then displayed (operation).

One of the two interfaces is the IP interface (Ethernet 10/100Base-T) which is always available for configuration and may also be used for operation, although different protocols will be used for the two tasks as shown in the table below.

The second interface is a specialized control network interface, like FT-10 for the CEA-709 devices and MSTP for the BACnet devices. Only the CEA-709 models are currently able to use their control network interface for configuration as well. The BACnet models need the IP interface for configuration.

The interface to use 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/NOT SET) or the interface mode to select on the panel.

Model	Interface	Task	Protocol	IP Jumper
LVIS-3E100	10/100Base-T	Configuration	FTP	Any Pos.
		Operation	CEA-852	SET
	FT-10	Configuration	CEA-709	NOT SET
		Operation	CEA-709	NOT SET
LVIS-ME200	10/100Base-T	Configuration	FTP	Any Pos.
		Operation	BIP	SET
	MSTP	Operation	MSTP	NOT SET
LVIS-3E11x	10/100Base-T	Configuration	FTP/SSH	Any Mode
		Operation	CEA-852	Mode IP852
	FT-10	Configuration	CEA-709	Mode FT-10
		Operation	CEA-709	Mode FT-10
LVIS-ME21x	10/100Base-T	Configuration	FTP/SSH	Any Mode
		Operation	BIP	Mode BACnet/IP
	MSTP	Operation	MSTP	Mode MS/TP

Table 5: Available interfaces

5.2 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 6.3):

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.

6 Operating L-VIS

6.1 Touch Screen

6.1.1 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 device.

L-VIS devices use a resistive touch screen. This type of screen can only detect one touch position at a time, that is, you cannot press two buttons at once. The screen delivers raw data which must be converted to LCD pixel positions before it can be used; therefore a touch screen of this type 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.

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: *Touch gestures may be disabled in the project configuration, in case the project uses a button-only navigation through the pages.*

6.1.2 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: *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 margin for error in all possible directions. For normal control sizes, the small errors introduced by differing viewing angles are not noticeable.*

6.2 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: *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.*

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 some punctuation). The entered string can be assigned directly to a connected string data point.
- **Touch and move:** For bar graph controls, the bar may be directly moved to the desired position by touching the bar and moving it to the desired height. While doing this, the control can be instructed to send periodic updates to the connected output data points, so the user gets immediate feedback (useful for light control).
- **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 controls and bitmap 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 6.2.1.
- **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 later on.

The control decides when to leave input mode. This is also called loosing 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 and stores it in non-volatile memory, so that it is retained across a power outage or reboot.

Controls usually return the focus when you touch an area outside of the control or when you leave the device idle for a specified amount of time. Some controls (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.

6.2.1 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 8.5.9.

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>
--------------	--

6.2.1.1 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.

6.2.1.2 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.

6.2.1.3 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.

6.2.1.4 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.

6.2.1.5 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.

6.2.1.6 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.

6.2.1.7 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:	<i>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.</i>
--------------	--

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

6.2.1.8 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:	<i>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.</i>
--------------	---

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

6.2.1.9 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:	<i>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.</i>
--------------	---

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.

6.2.1.10 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.

6.2.1.11 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.

6.2.1.12 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:

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).

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.

6.2.2 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.

6.2.2.1 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).

6.2.2.2 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.

6.2.2.3 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>
--------------	---

6.2.2.4 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.

6.2.2.5 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>
--------------	---

6.2.2.6 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.

6.2.2.7 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.

6.2.2.8 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:

*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.*

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.

6.3 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.

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. 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.
- **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.

The CEA-709 models 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: *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.*

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: *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.*

The BACnet models 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.

6.3.1 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: *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.*

7 Configuration Software

7.1 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 2.3.3 of the tutorial.

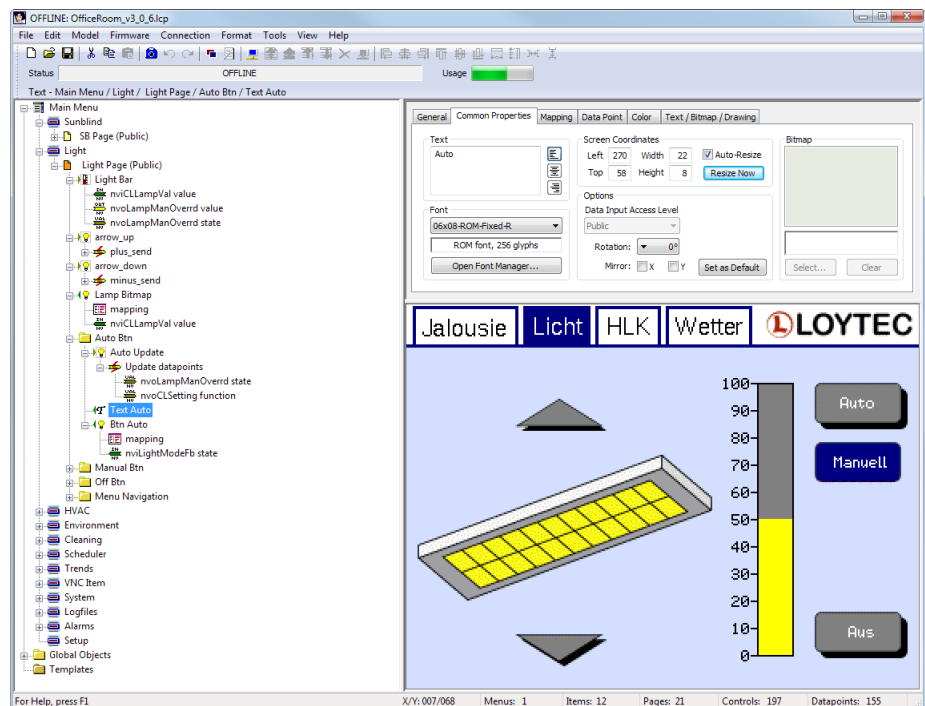


Figure 2: Main window of the L-VIS configuration software

7.2 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.

7.2.1 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.

7.2.2 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.

7.2.3 Cut / Copy / Paste

Every context menu contains commands to cut, copy, and paste the selected object. 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. The commands *Cut* and *Copy* will place a copy of the selected object with all its 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 another project.

7.2.3.1 Data References

If the copied object tree contains references to data objects, these references (data point objects) are copied without modification. For example, the copy of an entire page will reference the same data as the original page. If the new copy should reference other data objects, use the command *Manage Data Points* from the context menu of the copied object and reassign the data point objects to other data sources, as explained in section 7.2.5, 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.

NOTE:	<i>Not all types of data sources can be created automatically. This is especially true for complex data sources, such as alarm servers or local scheduler data points. These references must be resolved manually after the copy was inserted.</i>
--------------	--

7.2.3.2 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.

7.2.3.3 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.

7.2.3.4 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.

7.2.4 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 objects included in the XML output, so that importing the XML file into a different project may lead to unresolved references to fonts or data objects 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 14.14 for details about this function.

7.2.5 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 object they reference and which type and direction they have. This dialog is useful to organize data references, do batch renaming of data sources or reassignment of data points to different data sources, as is often required after 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 the data objects 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.

7.2.5.1 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 source Test1 from the original source 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 14.18.3 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.

7.2.5.2 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>
--------------	---

7.2.5.3 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 source, which must be of compatible type (same type and direction as the original source).

7.2.5.4 Reassign Data Point

This operation allows reassigning a batch of entries to a new data source, based on the currently assigned data source. 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>
--------------	---

7.2.5.5 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).

7.2.5.6 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 sources based on existing remote sources, instead of existing local sources.

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

7.2.6 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.

7.2.7 Lock / Unlock Position

This entry is available on collection objects. It can be used to lock the screen position of all objects inside the collection, so that these objects are not moved unintentionally, while working on other objects on the page.

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 once selected. The lock only prevents objects from being selected in the LCD view. To quickly move a locked collection or individual controls, select them in the tree view and move them to their new position in the LCD view, without the need to first unlock the entire collection.

7.2.8 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.

7.2.9 Add Items from Data Points

Available for menu objects, starts the automatic page create wizard described in section 14.7. 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.

7.2.10 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 14.7.

7.2.11 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.

7.2.12 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.

7.3 Main Menus

Many of the items in the main windows menus do not require additional explanation, as they are commonly found in modern 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.

7.4 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, the individual buttons will be mentioned in the sections where their functionality is described.

7.5 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 item's 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 an additional 78kB of 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 save a large amount of memory.
- 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.

7.6 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:

7.6.1 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.

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>
--------------	---

7.6.2 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 C style format string. 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 14.13.1.

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>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 on the Numeric Control property page to construct a suitable format string without knowledge of the details.</p>
Date Control	<p>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 the property page of the Date/Time Control, 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>

Table 6: 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 elements in the mapping table. The checkbox may be turned off for some controls to prevent automatic resizing and set the desired size manually.

NOTE:	<i>Manual size for text controls is most often used when the selection area should be larger than the longest text. In this case, the text is often also center-aligned, using the buttons to the right of the text input field. Another case is text received from the network. In this case, the configuration software does not know the size of the longest text at configuration time, therefore the user needs to reserve a suitable size for the text manually.</i>
--------------	--

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 an 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 14.15.

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, since this format 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.

Table 7: 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 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: *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.*

7.6.3 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 a new 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 unset and will be displayed as a black cross.*

8 Object Description

8.1 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 and the objects are designed as generic as possible, to ensure a steep learning curve and allow 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 automatically will show only the kind of objects which can be attached to the existing object, so there is no room for error.

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

8.2 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:	<i>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.</i>
--------------	---

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).

8.2.1 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 8: 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:	<i>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.</i>
--------------	--

8.3 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.
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 9: Function modes of a menu item

Always add a return item as the last item of your sub-menu, or the user will not be able to navigate back to the parent menu.

In the root menu, a return item is used in a different 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.*

8.3.1 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 10: Color assignment for menu items

8.4 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). Technically, the function of a page therefore is to group a number of controls together to show or hide them all at the same time.

8.4.1 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 7.6.2 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.

8.5 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 and 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.

8.5.1 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, 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 11: 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.

8.5.2 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 and fix its size according to the loaded bitmaps. If there is no bitmap to display, the size can be set manually and the bitmap controls area will be empty, with just a frame drawn around it in the selected frame color.

NOTE: *Bitmap controls without any bitmap and with the frame color set to the transparent background color are often used 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 empty transparent bitmap controls, rectangular regions may be defined on the page, which the user may select. The control itself usually has an action connected to it, to jump to a new page or update a data point.*

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 12: Color assignment for bitmap controls

NOTE: *For best results, use PNG format for your images and embed transparency information as alpha channel in the image. 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.*

8.5.3 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'.*

8.5.4 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.

The **Numeric Control** property page provides access to the specialized properties of numeric controls, like the allowed value range and a wizard to generate a suitable format

string for your control, if you don't want to enter it yourself on the common properties page.

In the area *Input Mode*, the properties concerning value input are defined:

- **Value Range:** The values entered in these two fields (min –max) define the limits for input of new values. The user can not 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.
- **Input Resolution:** This value is only used by the older LVIS-3ECTB model, which has a jog dial instead of a touch screen. The setting defines the amount by which the value is increased or decreased with each turn of the jog dial.
- **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:

- **Convert Data Point Value to Integer:** This check box 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 also controlled 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 (*).

The *Format String Wizard* on the right side of the page allows creating a correct format string easily 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 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.

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 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 14.18.1 of this manual. To output a percent sign, enter %% in the format string.</i>
--------------	--

8.5.5 Bar Control

Bar controls are 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).

The basic application of a bar control 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.
- **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 frame to resize the bar, similar to text and number controls.*

On the **Common Properties** page, a font may be selected to be used for the scale. 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.
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	Bar which represents the value. This color may be overridden by color specifications from a connected mapping table.
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 13: Color assignment for bar controls

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 is to 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.

The Flags control which components are drawn and how they are drawn. The following flags are available:

- **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.
- **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.
- **Invisible Bar:** This option disables drawing of the moving bar (the fill height). If the control is an input control, it is still possible to touch and drag to enter values, but the current value will not be visualized as a moving bar.

NOTE:

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.

8.5.5.1 Special 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 14.3) 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.

8.5.6 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 inputting 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 in default settings shows 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.

There is also an option to display absolute time on the X axis.

The control is able to display three curves per connected data point: the minimum value, maximum value, and average value. To do this, the control records 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 right end of each curve.

NOTE: *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 $(0 \cdot 9s + 20 \cdot 1s) / 10s = 2$. In contrast, the time independent arithmetic average value would be $10 (0+20)/2$.*

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.

Regarding color, the trend control is one of the most 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.
Background	Color of the background outside the graph. If the transparent check box is set, all pixels using this color will be transparent.

Table 14: Color assignment for trend controls

To set the color of the individual curves, select a data 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.

Table 15: 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 4 sections, which are described below.

On the top left, the **Value Axis** can be configured. Set the desired *Range* for this axis and enable or disable the *Tick Marks* and/or the *Scale Labels*. Optionally, a second range (*Range 2*) can be configured and enabled via the option *Secondary Scale*. If enabled, the secondary scale is drawn on the right end of the graph and the final scale range is adjusted such that a common horizontal grid between the primary and the secondary axis can be drawn. 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.

On the right side, the **Time Axis** is configured. Select a suitable *Time Span*, which is 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. To keep older data as well, which is not visible on the first page because it is older than the given time span, set the desired amount of *History Buffer* that should be used by this control. This value is given in pages, that is, if you set the time span to one hour and configure one history page, the control will store the values of the last two hours. Older data will be overwritten by new data.

NOTE:

For trend controls which are recording trend data locally (whenever a simple data point is connected to the control), the memory requirements can get 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 point and does not record any data by itself. The configured size of the history buffer is irrelevant in this case.

In the same section, you may set a minimum *Shift Interval* for the control. 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 you want the control to average data over a longer time frame than this, 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). The *Flags* to enable or disable the tick marks and the scale are the same as for the value axis.

At the bottom left of the property page, the desired *interpolation* method can be set for each of the connected data points. Select the data point first, and then set the desired interpolation. Currently available are two options:

- **No interpolation:** 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 interpolation:** 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.

The checkbox *Absolute Time* causes the scale labels on the time axis to display the date and time of the recorded data instead of a relative time. The label will normally display the time of day, except at midnight, in which case the new date will be displayed. The formatting of the time and date labels can be controlled through the input fields *Time Format* and *Date Format*. The same format strings as used for date controls are usable in these fields. The format is explained in section 14.18.2 of this manual.

In the middle of the property page are some additional **Flags** to be set. Depending on the selected object, there are different flags available.

With the trend control selected, the following options can be set:

- **Draw Grid:** This option enables a grid which is automatically laid out to fit the current tick mark spacing and allow easier readout of the displayed data. Set the final control size after enabling the grid and choose a size which yields a good looking result for the grid.
- **Draw Buttons:** This option enables the display of standard buttons to mark the areas of the control which are used to flip through the pages and zoom in and out. The buttons are not required to operate the control, so they may be hidden without losing functionality. It is also possible to place custom bitmaps on top of the trend control to mark the active areas. On true color devices, these bitmaps may also be semi-transparent.
- **Enable Cursor:** This option causes the control to become selectable. When selected, a cursor appears which allows navigating through the recorded data (see the next section).
- **Preserve Data:** When enabled, the Control is instructed to store the 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).

- **Line Width:** Sets the line with for all curves drawn in this trend control.

With a data point selected, the following 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.

NOTE: *Use the preserve data option with care and avoid unnecessary saving of unimportant data, since these operations may take up a significant amount of system resources and adversely affect performance and life time of the device in case they are over-used in a project.*

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.*

8.5.6.1 External Trend Data

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.

Trend services like these are represented by a special type of data point (trend data 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 local trend which contains more than one data recording, the trend will show the trended data points as child points. Selecting and attaching one of the child points to the trend control will show the corresponding curve.

NOTE: *The desired settings for the time axis and the number of history pages available for navigation must be configured even in case of external trend data.*

8.5.6.2 Trend Data Access

For every trend control which stores the 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::@<address>/`

Replace `<user>` with the user name for data access as defined in the project settings and `<address>` with the IP address or DNS name of the device, for example:

`ftp::data@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.

8.5.6.3 Operating Trend Controls

Trend controls do not allow data input, but they can be marked selectable. If the control is selectable, touching it on the screen will display a vertical line at the touch point and open a data window, showing the exact data which was recorded for the selected point of time. The control will show the absolute date and time for the recorded data, as well as the value 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.

The vertical line may be moved around in the control by dragging it around or touching another location of the curve. The data in the data window will be updated accordingly.

If there is more than one page (the number of history pages was set to 1 or more), the user can flip through the pages by moving the cursor to the rightmost or leftmost end or touching the rightmost or leftmost end of the time axis. Small arrows will appear on both ends of the time axis to indicate a good place to touch if you want to flip pages.

The control will return to normal operation when another control is selected or the user touches any other location outside the selected trend control.

8.5.7 Date Control

The date control is a specialized control to display date, time and calendar weeks. 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 the first of January 1970.

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 two different display modes for a date control: a text based mode and a graphical analog clock simulation. Even though one date control can only use one of the two modes 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 the text mode can be selected on the **Common Properties** page. All other settings can be done on the **Date/Time Control** property page. There is a *wizard* to generate a suitable format string for text mode, set the *Style* of the control (text or graphic) and modify the look of the *Graphic Style*. The date format string for text mode can also be entered directly on the **Common Properties** page. The exact format to use is described in section 14.18.2 of this manual.

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 16: 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.

8.5.8 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.

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 8.14 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 17: Color assignment for data log controls

8.5.9 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.

8.5.9.1 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 6.2.1 for details about operating a schedule control in standard mode on the running L-VIS device.

8.5.9.2 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 an LWEB-801 server together with the Master Schedule Configurator software.

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:	<i>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.</i>
--------------	---

Please see section 6.2.2 for detailed information about how to operate a schedule control in extended UI mode on the running device.

8.5.9.3 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:	<i>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.</i>
--------------	---

8.5.9.4 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.

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 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).
- **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 daylight:** 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.
- **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 8.5.9.5 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 three fixed levels to

choose from (low/normal/high). The mapping is optimized to work together with the LWEB master schedule Configurator system, which also uses three 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).

- **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 18: Color assignment for schedule controls

8.5.9.5 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.

8.5.9.6 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 presets 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.

8.5.10 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.

The default information displayed for each alarm 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.

On the **Alarm Log** Control property page, the following options can be set:

- **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).
- **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.
- **Floating point value field precision:** This input field specifies the number of significant digits to print for alarm values.

NOTE:	<i>If an action object is connected to the alarm list control and the action trigger is set to 'value update', the action will be executed whenever a new entry is added to the alarm list. This can be used to switch to a specific page (action 'show page') or write a value to a data point (action 'update data points') to notify the user of the new alarm list entry.</i>
--------------	---

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.
Active Backgr.	Background color for entries which are currently in alarmed state.
Selection Frame	Frame drawn around the entire control, if the control is selected.
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 19: Color assignment for alarm list controls

8.5.11 Vector Drawing Control

The vector drawing control is similar to a bitmap control (see also section 8.5.2). 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 8.5.11.1 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 8.5.11.1) 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 with 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 20: 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 8.5.3 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.

8.5.11.1 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 drawing object, 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 selection the item *Edit Contents* from the context menu of the control.

While in drawing mode, the control occupies the entire screen. New lines or points may therefore be drawn anywhere on the page. When leaving drawing mode later, 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. If another line segment should be appended, left-click at the desired end of the first line and beginning of the next line and repeat the process. If no further line segment should be added, right-click at the desired end position to finish.

- *Add new points (polygon drawing):* To add another point to the 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 procedure as needed. Press the right mouse button to end point insertion mode.
- *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. 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.
- *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 can then be moved to the desired position by dragging with the mouse and releasing the left mouse button when done.
- *End drawing mode:* To exit drawing mode, right-click anywhere on the page, except over a line (see the remove lines function above). 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.

8.6 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, or action objects.
- A **data object** 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.
- A **data point** object which references a specific data object and thereby links a control to a data source or sink. It also defines the properties of this data link, for example when new data is to be transferred between the data object and the control and how the data should be converted when it is transferred.

The **control** is created as part of the object tree, as outlined in the tutorial. It exists independently of any register or network object. Even if all registers and network objects were to be deleted, the control would not be affected.

The **data object** 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 objects

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 objects representing the system parameters or data objects representing dynamic network variables, which may be present on a CEA-709 device.

NOTE:	<i>Data objects may be thought of as the source or sink of a data flow, where every object has a defined direction (either source or sink, e.g. data coming in or data going out). For internally created registers, two such data objects are created, one to write data to the register and one to fetch the current register value. Input and output network objects are represented by a single data object with the appropriate direction, BACnet value objects and other objects which may be read and written are represented by two data objects, one for read access and one for write access.</i>
--------------	---

Structured network objects, as available on CEA-709 devices, are represented by individual data objects for each of the structure elements in addition to the object representing the entire network variable. However, most of the complex data objects, 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:** Data output objects of this type may be directly connected to an alarm generator to control the entire switch. See also section 8.11 for detailed information about alarm generators.
- **SNVT_alarm2:** As above, data 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 objects will be displayed directly on the text control, without a lookup in a mapping table. If the data point is an output point, 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 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 objects to controls is done via a separate object, instead of directly connecting a physical data object to the control has a number of reasons. The most important ones are:

1. Data objects represent network objects which exist on the device. They must be independent of the existence of any control or other user interface object. If a data object were to be connected to a control and the control is deleted, the data object would be deleted as well. Using a reference object to link the control to an existing data object solves this problem. Deleting the control will also delete the reference object (the data point), but not the data object itself.
2. Many projects need to link multiple controls to the same data object, for example to display a temperature which is received by a certain object from the network on a bar control and on a numeric control at the same time. This would not be possible if the data object would be directly connected to a control, since it could only be at one place in the object tree at a time. A separate link or reference object is required, which points to the data object from which the control should receive its value updates.
3. Sometimes it is required to apply simple linear transformations to the data objects raw value before it can be displayed. For example, an object 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 object 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 on the fly, while passing values between the control and the data object. This makes it easy to display for example a temperature from an input data object both in degree Celsius and in degree Fahrenheit side by side, using two data points referencing the same data object, but using different value translations.

It is important to understand this concept to understand what can be configured on the **Data Point** property page. The key is that there can and usually will be multiple data points referencing the same data object.

8.6.1 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:	<i>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 $DP = -1 * NV + 1$.</i>
--------------	---

The *Details* section shows details of the referenced data object, 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 object, by which the object 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 will not request the underlying data object to be updated, even when the value on the control side changes. Useful if this data point represents one element of a structured data object, and updates to this element should not cause the complete object 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 8.5.3 for more information).
- **Immediate:** All values are immediately transmitted to the underlying data object. 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 underlying data object is a value output, the default value or the last stored value will be sent out, if it is a

value input, the device will try to request the current value from the sender. Both actions can only be successful if the underlying network technology supports them. In cases where the requested operation cannot be executed, it is silently ignored. 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 can not send a request for a value update.

- **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 to be the opposite of the referenced data source, 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.

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 ultimately refers to the value of the underlying data object. Since the referenced data object can only have one value at a time, the entered default value will apply to all other data points which reference the same data object as well. The entered value will be run through the value translation to determine the resulting value for the underlying data object and this value will be distributed back to all connected data points to maintain consistency. Only if the data point 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 non-constant data points.*

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 points which reference the same underlying data object, if the data point is marked as constant value.

NOTE: *Constant value data points 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 referencing the same data object). Now you can set the data points to constant value and enter different values for each of the three points (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 object, this option will also affect all other data points which reference the same data object.

NOTE: *The persistent option is useful for output data points which need to be sent out after system boot (System Startup flag is set) but are controlled by actions or math objects instead of direct user input. Such data points must preserve their last calculated output value across the reboot, but there is no control which would normally cause the output value to be stored when the user finishes data input. Checking the persistent flag will solve this problem.*

The *NV Timing* section is used to control the timing of update messages on the network, resulting from this data point. A minimum and maximum send time may be specified for output data points, a receive timeout may be specified for input data points. A value of 0 always means 'unlimited'. The individual parameters work as follows:

- **MinSend:** At least the specified amount of time has to pass between two successive update messages on the network. If the device delivers more updates than this, intermediate values will not be sent out, until the minimum send time has passed, at which time the current (most recent) value is sent out.
- **MaxSend:** If this amount of time has passed without any updates, usually because the value did not change in the meantime, then the current value will be sent out. This is commonly described as a heartbeat function, used to allow the receiver to detect if the sender is still alive and just not sending new information because there is none, or if the sender has died and therefore no new data is received.
- **PollTime:** Can work as a simple poll interval, or as a receive timeout. Usually used as receive timeout for bound NVs on CEA-709 devices and otherwise used as poll cycle time. If used as a receive timeout, it specifies the amount of time after which the device sends out a poll request, if no value update was received. As a poll cycle, it specifies the amount of time after which a periodic poll request should be sent out, independent of any updates which might have been received in the mean time.

8.7 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.

8.7.1 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: *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.*

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: *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.*

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.

8.8 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 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.

8.8.1 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 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.

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.*

8.9 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. Currently these objects are controls and data point connectors.

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 and how to avoid the bad ones will be given later on.

8.9.1 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 connected output data points. The value to send out can be preset by connecting 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 9.9 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.

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* (NOT to the action) or when the user enters a new value using the control.

- **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 13.1.2 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, without further conditions. This is one of the most widely used settings.
- **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 exactly once (like the condition execute once) but only if the value condition applies.
- **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.
- **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.
- **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.

8.9.2 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:

- **..** 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 x^{th} 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 x^{th} 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 8.15.6 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.

8.10 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 8.15 of this manual.

8.10.1 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*).

8.10.2 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:	<i>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.</i>
--------------	--

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.

8.11 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 section 9.5 of this manual.

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:	<i>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.</i>
--------------	--

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.

8.11.1 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:	<i>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.</i>
--------------	--

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.

8.11.2 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.

8.11.2.1 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.

8.11.2.2 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: *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.*

8.12 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. This can be useful for example to connect an internal register to a network output. If the register contains the result of a complex calculation or is controlled by various objects across the whole project, it is easier to have one single point of connection to the outside world, instead of connecting the network output in parallel to the register write data point at every location where the register is written.

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 9 for more information about this topic.

NOTE: *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 14.12 for more details on this topic.*

8.13 Mathematic Objects

Mathematic objects are used to do calculations on connected input data points and assign the result to the connected output data points. The 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.

8.13.1 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:	<i>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.</i>
--------------	--

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:	<i>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.</i>
--------------	--

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, either double click the variable in the list at the left 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:	<i>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.</i>
--------------	---

8.13.2 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))
```

If you have to use it as an argument to another function, it may only be 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 the equivalent 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).*

8.13.3 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

Function	Return Value
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^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 <= 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 >= 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 radians) whose cosine is v1
atan(v1)	Returns the arc tangent of v1; that is the value (in radians) whose tangent is v1
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 no positive integers. For nonnegative 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)	computes the absolute value of the argument v1

8.14 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.

8.14.1 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 8.5.8 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, 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:

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.

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: 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.

8.15 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.

8.15.1 Template Properties

Both templates and their instances share the same property page called **Template / Instance**. The page contains only a few input fields to set:

- **Display Offset:** The *Display Offset* is only used for template instances and defines the relative offset of all visible objects based on their position in the template. This is used to adjust the final position of the templates contents on the screen, which is especially required when the same template is instantiated multiple times on the same page. If possible, the template should be designed with the desired final position in mind, so that in most cases the offset can be left at zero and the contents of the template are shown at their original position.
- **Base paths for relative data point references:** On the right side of the property sheet is a section which shows up to 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.

8.15.2 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 display offset is not changed.

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 to make the template work. 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.

8.15.3 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.

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). Also available is the *display offset* for this instance, which can either be entered here or simply set by moving the bounding box in the LCD preview window.
- **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.

8.15.4 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:	<i>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.</i>
--------------	--

8.15.5 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.

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 let the user know if a template instance has local changes or not, a pen symbol is added to the icon in the tree view whenever at least one local change is stored with the instance.

NOTE:

*It is also possible to take back individual changes. Either use the **Restore Defaults** command in the context menu of the object to reset all local changes related to this object, or set the property in question 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.*

8.15.6 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.

9 Data Point Management

9.1 Concept

As outlined in the description of the data point object in section 8.5.11, the network variables, BACnet server objects, client mappings, user defined registers, and system parameters are represented by objects which exist on the device independent of any controls or other objects in the object tree. These data objects are then referenced from the data point objects in the tree by their unique ID.

While some of these objects are created automatically, most of the data objects need to be created by the user. There are ways to manually create individual data objects, import object definitions from CSV files or scan the network and create new data objects based on the information collected by the network scan. Once created, a data object may be used to exchange data with other devices on the network, store and load internal registers, or access internal system parameters.

Every data object on L-VIS has a distinct direction of data flow, either read (input) or write (output). 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:

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 object will receive new data from the network and deliver this data to the application running on the L-VIS device, whereas an output data object will transmit new data from the L-VIS out to other devices.

All data objects fall into one of the following categories:

- **System Parameter:** This is a data object which cannot be created by the user. It is provided by the device to modify certain internal system parameters, like the time zone, IP address, BACnet 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 parameter object is either a read-only or a write-only value, depending on the direction of the data object.
- **Register:** A 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 by two data objects. 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 such objects are either write-only (output NV, BACnet input objects) or read-only (input NV, BACnet output objects). They are therefore represented by one data object with the appropriate direction. One exception to this rule are BACnet value objects, which may be read and written and which are therefore represented by two data objects, just like registers.
- **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 object on the L-VIS device to a data point on a different 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. There is currently only one reference per external NV. 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 does not change its configuration.

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.*

9.2 User Interface

Since the data objects 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 objects, when a data point is to be added to the object tree. It is therefore possible to create and select a data object 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:

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.

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:

9.2.1 Folder List

At the left is a list of folders which is used to sort the available data objects 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 objects in the import folder are not stored on the L-VIS device when the project is downloaded. They represent data objects which are available on remote devices and are shown here as templates to create suitable data objects for use on the L-VIS device (see section 9.3.4 for details).
- **Local NVs / Server Objects:** Objects 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 object.
- **External NVs / Client Mappings:** Objects 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 objects which were created from points in the import folder, by using the option *Use on Device* from the context menu of the imported point.
- **Register:** 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. As explained before, a new register will automatically yield two new data objects (read and write).
- **System:** This is the folder in which the available system data objects 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 for the object tree.
- **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 object, that is, the data objects 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:** This folder is used for local trend service data points. Trends which are defined on this level (as opposed to trends recorded by trend controls) are independent of the visualization and may be accessible by other devices on the network (currently only possible on BACnet networks).
- **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.

Using the context menu on a folder, sub-folders may be created to organize the available objects. If new objects are created automatically, they are usually placed in the base folder and can then be moved by the user to any of his sub-folders.

9.2.2 Data Object List

At the top right, a list of all data objects 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. A double-click will select the data object and then close the window (same as the OK button), which is useful when adding new data point objects to the object tree in the main window.

The columns of the list will vary depending on the type of objects shown. Always visible are the data object name and the unique ID by which this object 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 objects which are currently unused (the Use column shows 0) may be removed from the list.

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. The list may be sorted by a click on the desired list column.

When an entry is selected, detailed property information is shown below the list, in the property view.

9.2.3 Configuration

Data point configuration is done in the area below the data object list. There are a number of different tabs available.

9.2.3.1 Properties

The property view is used to display detailed information about a selected data object. Some of the properties may be modified in this area, while others (marked with a lock symbol) are read-only.

If more than one data object is selected in the object list, the property view shows only the properties which are common to all selected objects.

9.2.3.2 Manage Datapoints

This view is used to execute bulk operations on the list of selected data points. There are buttons to rename the selected data points based on a search/replace pattern or regular expression, create copies of the selected points and assign suitable names to the duplicated points, and a button to create local data points from remote data points.

9.2.3.3 Manage Relations

This page provides an overview of all existing property relations. It can be used to assign data points to these relations to access the data for visualization. Property relations are used by some of the more complex data points to provide access to extra information, originating from internal data processing. For example, an alarm server data point provides access to the current number of active, acknowledged, and unacknowledged alarms, and a data point with an intrinsic alarm condition provides information about the current alarm state.

NOTE:

Relations are only links to real data points. As such, they cannot be used on their own like normal data points. A real data point needs to be assigned to the relation for the data to be trended or used on a control or as an input for an action or a formula.

9.2.3.4 Local Connections

This view is used to manage local connections between different data points on the device. New connections can be defined, organized in folders, and data points can be added to connections easily by drag&drop from the data point list.

9.2.3.5 Global Connections

This view is used to create global connections. These connections can be used to exchange data between different LOYTEC devices on the IP network.

9.2.4 Buttons

At the right side of the window are a few buttons to create new data objects, 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 object folders and removes any objects which are currently not referenced by data points 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.
- **OPC Server:** This button opens the OPC Server 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.
- **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. The main window will show this selection when the data point manager window is closed. From this first occurrence, you may use the *Find Next* function (or F3) of the main window to find similar references throughout the project.

9.3 Object Creation

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.

9.3.1 User Register

A new user register is very easy to create. Select the folder named Register and click the *New* button. A new dialog appears, asking for two names, a *Datapoint Name* and a *Register Name*. The first one is used as a base name for the data objects 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.

9.3.2 Local Network Variable (CEA-709)

For CEA-709 models, local network objects refer to local network variables on the L-VIS device. If there are dynamic NVs present on the device, suitable data objects will be created automatically. All manually created objects will therefore create additional static network variables on the device.

NOTE: *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 2.2.1 for more information on this topic.*

To create a new static NV, select the folder named *Local NVs* and use the *New* button or the context menu to add a new data object. 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 object. If this is left blank, the data objects 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.

9.3.3 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, much like a register, two data objects will be created for each value server object.

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 object, whereas a BACnet output object is represented by a data input object.

To create a new server object and the required data object(s) to represent it, select the folder named Server Objects and use the *New* button or 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 object 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 object(s) to represent it. As explained before, you will get two data objects if you create a server object of type ‘value’, in order to be able to read and write your object.

9.3.4 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 client mappings may be created.

9.3.4.1 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 object 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 9.3.4.3 below).

9.3.4.2 Network Scan

Scanning an existing network is another way to generate template 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 via FTP:** 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 FTP:** 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 object management window. The scan folder should now have a sub-folder for each of the scanned devices, containing objects representing the data points available on these devices. Now the required mappings may be created from these points (see next section).

9.3.4.3 Create Mapping

Browse through the available points in the import folder and select the data points for which mappings should be created. Multi selection within the same folder is possible. Then open the context menu and select *Use on Device*. For all selected points, a suitable remote data object will be created.

NOTE:

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 data objects which are available on the remote device (for a network variable input on a remote device, you will see an input data object in the import folder). Once the point is used to create a remote network object on the L-VIS device, the direction will change, since a network variable output will be required to send data to the remote devices network variable input.

To simplify the selection and configuration of remote network objects 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 variable objects 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 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 point matching the template is used on the device.

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 points. To do this, select the command *Use on device and apply all templates* on the top level scan folder.

NOTE:

A remote network object refers to another device in the network, which had a certain configuration at the time the scan data was collected. If the configuration of the remote device changes later on, the remote network object 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 objects instead when using a CEA-709 device and use the network binding 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.

9.4 Local Scheduler and Calendar

To use scheduler and calendar functionality on the device, the appropriate data objects need to be created first, followed by a few additional configuration steps which are detailed below.

9.4.1 Point Creation

As the first step, the required data points must be created. Create the calendar point in the calendar folder and the required scheduler points in the scheduler folder, as described earlier in this chapter.

9.4.2 Calendar Point Configuration

Once the calendar point is created, attach new calendar patterns as required to describe the individual day classes, like holiday, maintenance day, and so on. Do this by using the command *Create Calendar Pattern* from the context menu of the calendar point. A new dialog appears where you can enter a name for the calendar pattern and edit the days which should belong to this day class.

When done, close the dialog using the *Create Pattern* button. Repeat this step to create all required patterns. All patterns configured here will later be available on the local schedulers for use.

9.4.3 Scheduler Point Configuration

To configure the local scheduler points, select the command *Configure Schedule* from the context menu of the scheduler point. The same dialog which appears when a new scheduler is created is shown and allows one to configure the scheduler. Of course, this step can also be done directly when the point is created.

First, attach the data points which should be controlled by the scheduler. Switch to the tab *Scheduled Datapoints* and use the *Attach Datapoints* button to select and attach one or more data points to the scheduler.

NOTE:

There may be limits in how many and which data points may be attached to a scheduler. For example, on a BACnet scheduler, only data points with compatible data types may be attached to the same scheduler object. On a CEA-709 scheduler, only data points representing an entire NV may be connected, but not individual elements of a structured NV.

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.

Enter a Description text in the second column of each line. This text will be shown when the user changes a value set on the device later on.

Add new value presets by entering a name and pressing the *Create* button next to the input field. 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.

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.

When done with the point and value setup, switch back to the *Configuration* tab. You should now see the list of defined value presets in a list at the top right of the window. Drag and drop presets from this list into the time table area to define the desired output values throughout the day. Days may be copied to other days using the *Copy to* button.

At the bottom left, there is a list of all available days, which consists of the seven week days which can be used to define default schedules for each day of the week, plus any additional special days defined by the local calendar. For the special days, a priority must be assigned, so that the system knows which schedule to follow on a day which is part of more than one calendar pattern.

NOTE:	<i>The priority settings are according to the underlying network technology. For CEA-709, the highest priority is 0, the lowest is 126 (127 is reserved for the default weekly schedules). For BACnet, the highest priority is 1, the lowest priority is 16.</i>
--------------	--

When the setup is done, save the changes and exit the configuration dialog.

9.4.4 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.*

9.4.5 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 the LWEB Master Schedule Configurator and/or 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.*

9.4.6 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.

9.4.7 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 8.5.9 for more information about schedule control configuration and sections 6.2.1 and 6.2.2 for a guide on how to operate a schedule control on the device.

9.5 Local Alarm Server

To use a local alarm server, create an alarm server point in the folder called *Alarm*. On CEA-709 devices, there may only be one alarm server for the entire device. BACnet devices support an arbitrary number of alarm servers.

There are two different types of alarming available. The current software supports one type for BACnet and the other type for CEA-709 devices:

9.5.1 Intrinsic Reporting

BACnet devices support an alarm mechanism called intrinsic reporting. In this concept, high and low limits are specified for individual server objects and alarms can be generated by the server object and reported to an assigned alarm server when the value of the server object exceeds these limits for a specified time period.

Other devices in the network can register for alarm updates at the alarm server object (which is a notification class object) and display the current list of alarms as well as acknowledge individual alarms.

To set up a local alarm server and assign it to server objects, the following steps are required:

- Create the needed alarm server points (more than one may be created to organize alarms into groups). From the properties of the alarm server point, set the *Ack To-Fault* and/or *Ack To-Offnormal* checkboxes if the alarms from this server need to be acknowledged.
- Add alarm conditions to the server objects you want to monitor. This can be done via the context menu of the server object, using the command *Create Alarm Condition*.
- Assign one of the existing alarm servers to this condition, which is done in the same dialog where the alarm condition is specified.

9.5.2 Algorithmic Reporting

CEA-709 devices do currently not support the mechanism of intrinsic reporting. Instead, the application needs to provide an algorithm which determines if an alarm state is reached.

On L-VIS devices, the alarm generator objects as described in section 8.11 are used for this purpose.

To set up alarming using algorithmic reporting, the following steps are needed:

- Create the alarm server point.
- Add alarm generator objects to your project as required.
- Set up the alarm generator objects. Connect the input data points which should be monitored and use the alarm server point as the output point of the alarm generator.

9.5.3 System Resources

On BACnet devices, each alarm server point automatically instantiates a notification class object to handle alarm subscriptions and notifications. The object name and instance number for the object can be configured in the same way as any other server object.

On CEA-709 devices, the device firmware must provide a software module to handle alarming. To activate this module, open the *Network Settings* Dialog which is accessible from the *File* menu and check the option **Enable Alarm Server**. This will add new NVs and thus change the static interface of the device.

9.5.4 Using Alarm Servers

Each alarm server keeps a list of currently pending alarms, distributes them to any clients which may be connected to the server and also receives acknowledge requests from either the application or connected alarm clients.

To access the current list of pending alarms on an L-VIS device, there is a specialized control available to which alarm server and also alarm client data points may be connected. Please refer to section 8.5.10 for more detail about alarm list controls.

9.6 Local Trend

To set up a local trend of a data source, create a new trend point in the folder called *Trend*.

On BACnet L-Vis devices, such a trend recording is accessible to other devices in the network, but the trend can only operate on BACnet server objects and will accept only one object per trend, which is a limitation imposed by the BACnet specification for trend data exchange between devices.

On CEA709 L-Vis devices, a trend may record data from multiple objects at the same time, including registers and other non-CEA709 data objects, but the trend data is currently not accessible for other devices due to the lack of a standardized way to exchange trend data on CEA709 networks.

9.6.1 Trend Data Object and Trend Controls

Since trends may be recorded either directly by a trend control or independently by a trend data object, 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:

9.6.1.1 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.

9.6.1.2 Recorded by a Trend Object

- Trends recorded by an independent trend data object 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.

9.6.2 Trend Mode

When creating a new trend data object, there are a number of different modes of operation available to choose from. Since they can have a big influence of the resulting trend data, they are explained in this section in more detail.

9.6.2.1 Interval

This is the most commonly used trend mode. A new value is recorded at a certain fixed interval, for example once every 10 minutes. Given a certain maximum size of the trend log buffer, the time frame covered by the trend is known. For example, taking one reading every hour with a maximum buffer size of 240 entries will provide a history of the last 10 days.

9.6.2.2 Change of Value

In this mode, a new recording is made whenever the current value changes by at least the amount specified as the COV delta. This may reduce the number of records stored, when the value does not change significantly. However, it is no longer known for how long the allocated log buffer will last, since the number of records varies. Also note that for small COV delta settings or data points which are constantly change their value, the number of recorded entries may be higher than desirable. This mode is best suited for binary data points, where every change (On/Off) must be recorded.

9.6.2.3 Trigger

This mode uses an external trigger event to record a new reading. The trigger is controlled by a separate data point, which may for example be scheduled by a local scheduler unit to store a new record once every day at a certain time. This is very useful in recording meter readings or similar data, where readings should be recorded at a specific time of day.

9.7 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: *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.*

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.

9.8 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: *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.*

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.

9.9 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 connected 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 8.8).

In addition to the mail template setup, the mail account setup as described in section 13.5 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.

9.9.1 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: *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.*

9.9.2 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 8.8 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.

9.9.3 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: *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.*

NOTE: *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.*

10 Standard Procedures (CEA-709)

10.1 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.

10.1.1 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.

10.1.1.1 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.

10.1.1.2 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.

10.1.1.3 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 a manual device integration if there are static NVs to be created, even if you don’t actively use the dynamic NV points.

NOTE:	<i>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.</i>
--------------	---

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.

10.1.2 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.

10.2 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>
--------------	---

10.2.1 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.

10.2.2 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 PCs 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.*

10.2.3 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.

10.3 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>
--------------	---

10.3.1 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.

10.3.2 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.

10.3.3 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 unconfigured 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:

*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.*

10.4 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.

10.4.1 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).

10.4.2 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.

10.4.3 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.

10.4.3.1 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.

10.4.3.2 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:

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.

11 Special Functions (CEA-709)

11.1 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: *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.*

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: *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.*

11.2 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:

11.2.1 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>
--------------	--

11.2.2 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.

11.3 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.

11.4 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.

11.4.1 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 3.

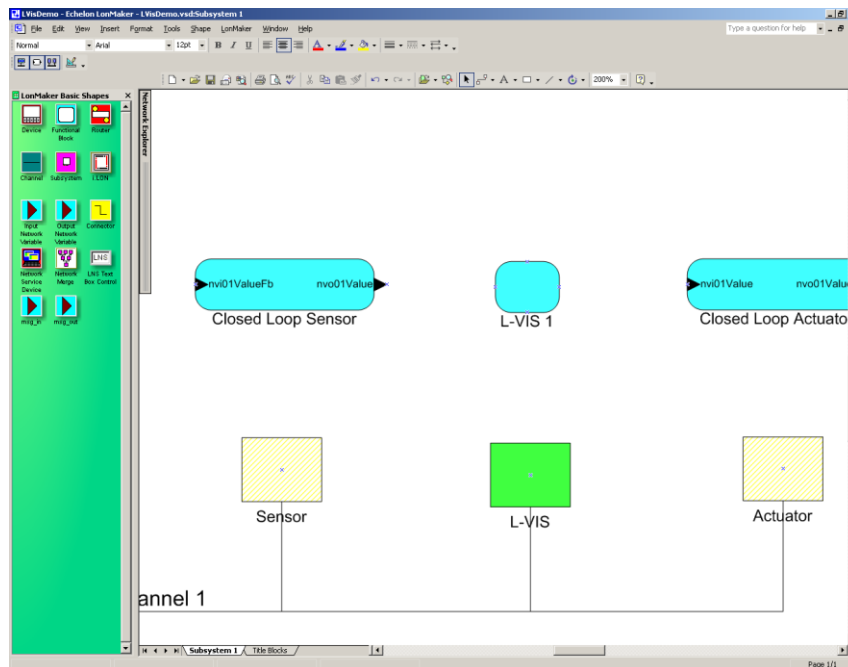


Figure 3: Create a function block

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 4 is shown.

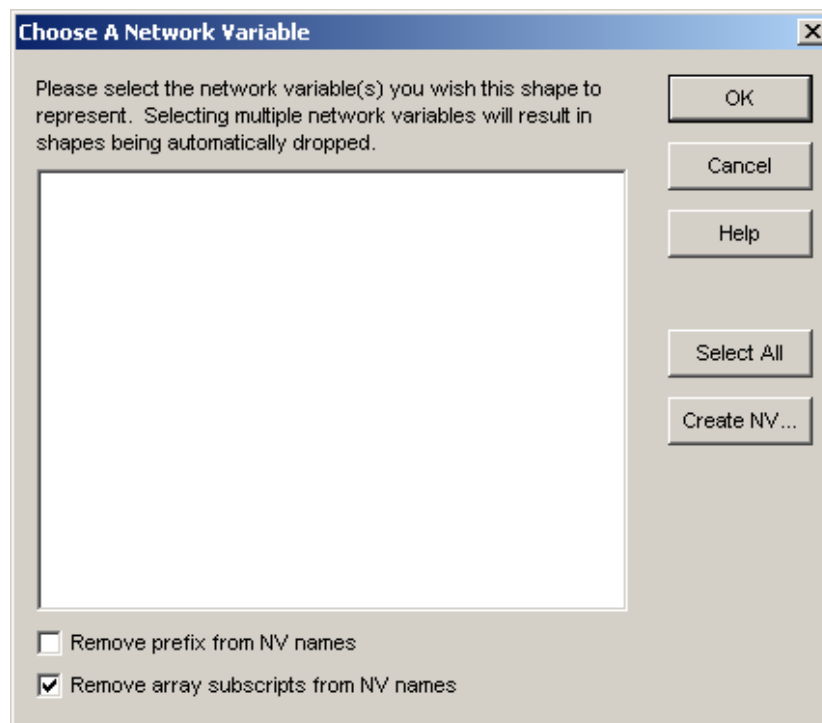


Figure 4: Choose network variable dialog

Select 'Create NV' and in the displayed dialog (Figure 5), enter the name for the new network variable and click on 'Browse'.

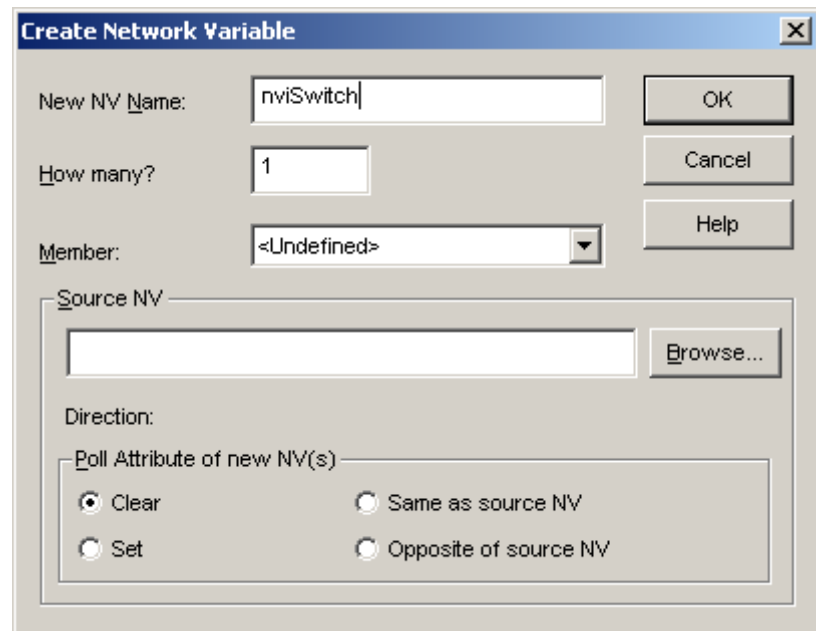


Figure 5: Create network variable dialog

In the dialog (Figure 6), 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.

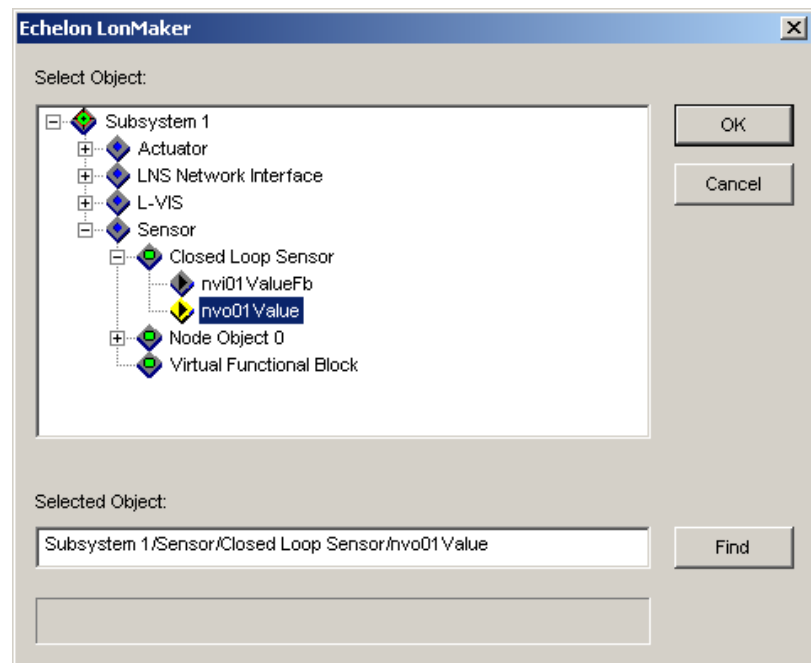


Figure 6: Select complementary variable dialog

Now the source NV name in the Create Network Variable dialog is filled in (Figure 7).

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.

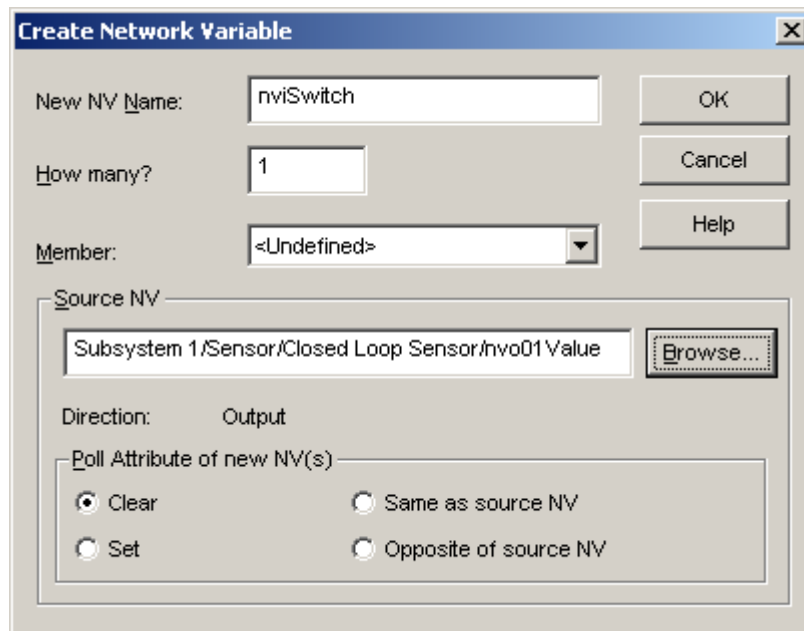


Figure 7: Create network variable dialog

This completes the creation of the new dynamic network variable (Figure 8). 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.

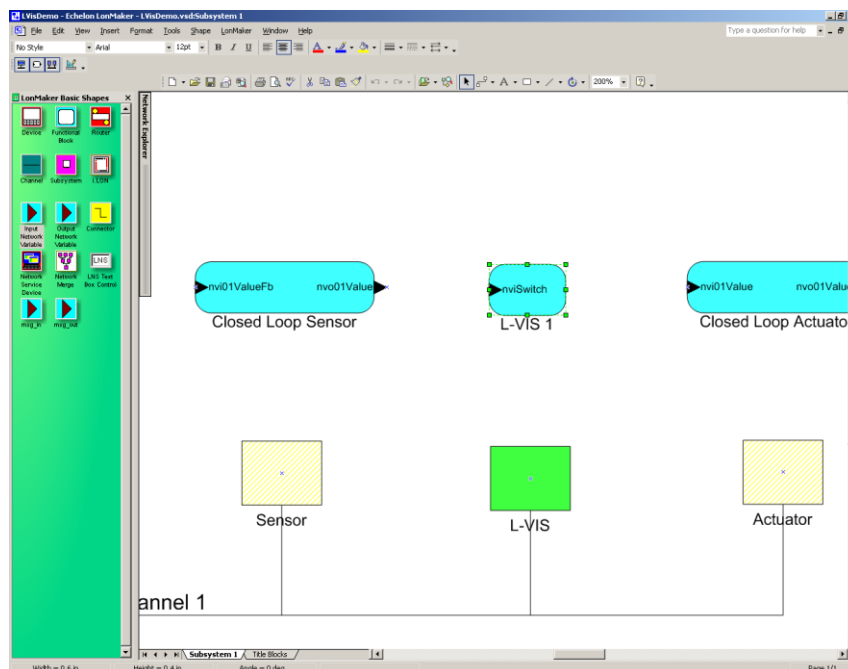


Figure 8: New dynamic network variable

11.4.2 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

9), 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.*

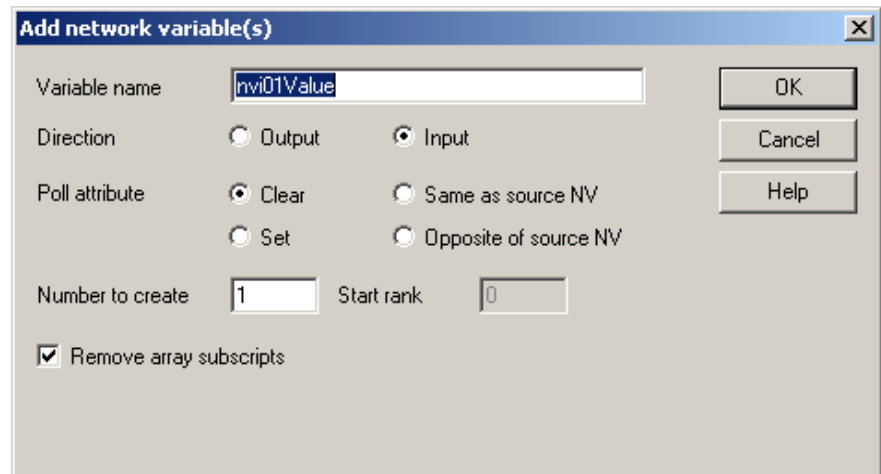


Figure 9: Add network variable dialog

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 10).

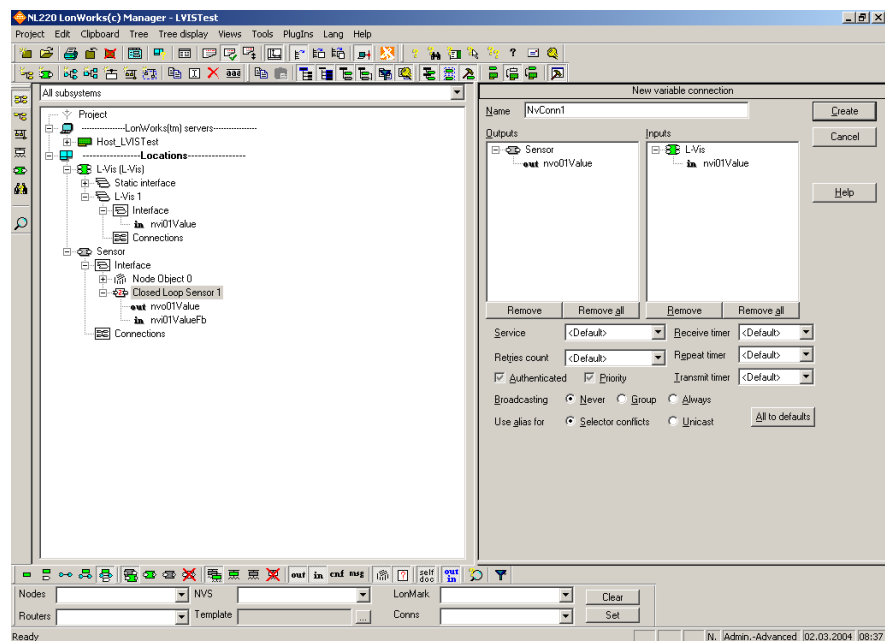


Figure 10: Network variable connection window

11.4.3 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 11).

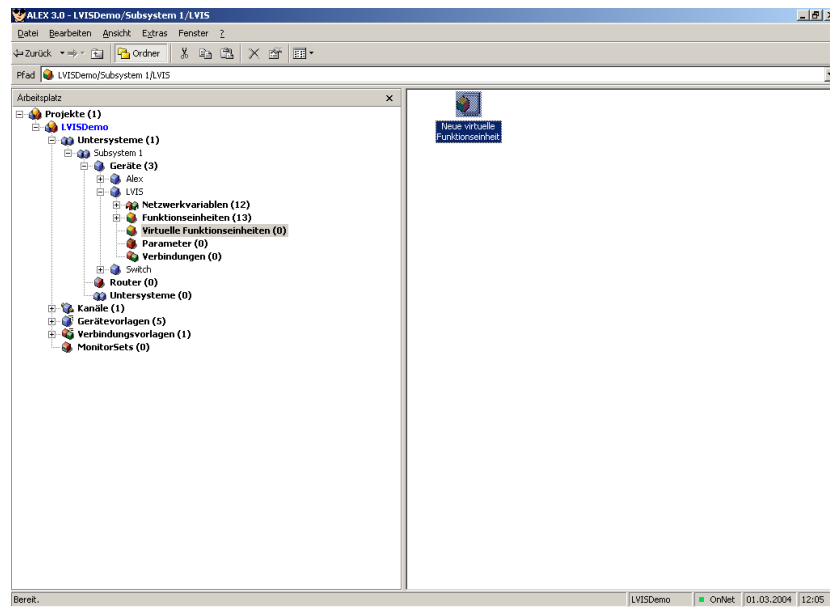


Figure 11: Alex main window

Enter a name in the dialog (Figure 12) and press *OK*.

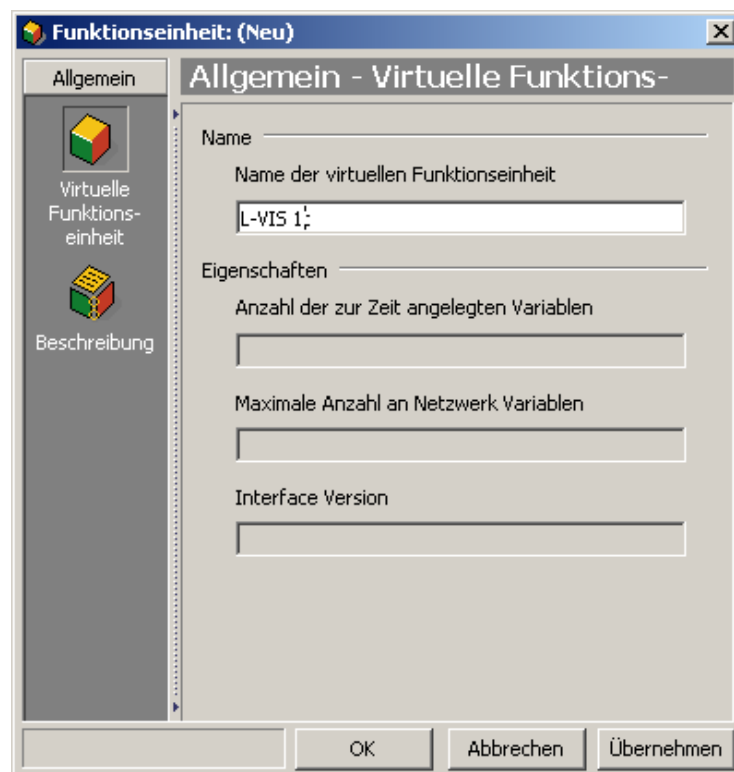


Figure 12: Virtuelle Funktionseinheit dialog

Open the Netzwerkvariablen collection in the tree view and select *Neue Netzwerkvariable* (Figure 13).

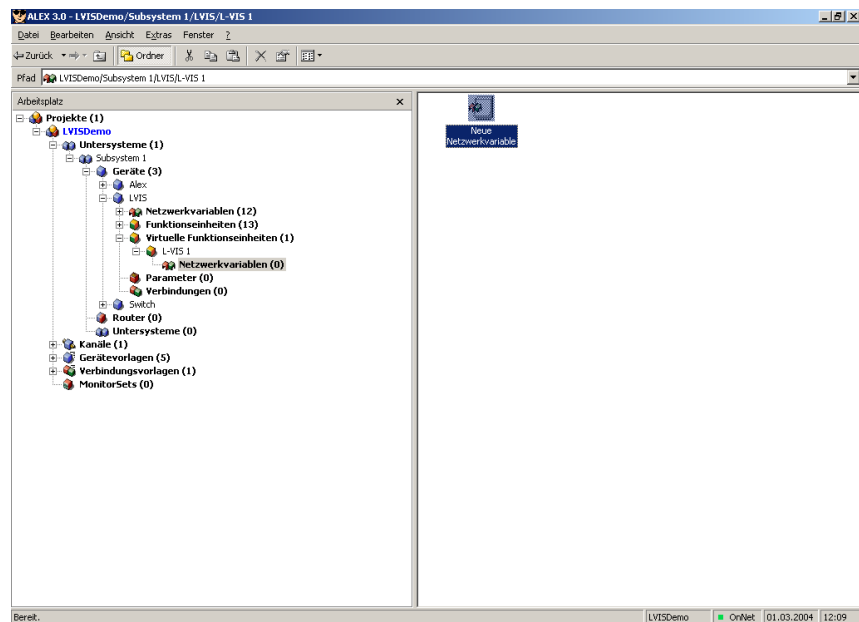


Figure 13: Create new network variable

In the Netzwerk Variable dialog (Figure 14), select the device template (Gerätevorlage) and the network variable to which the new dynamic network variable will be connected.

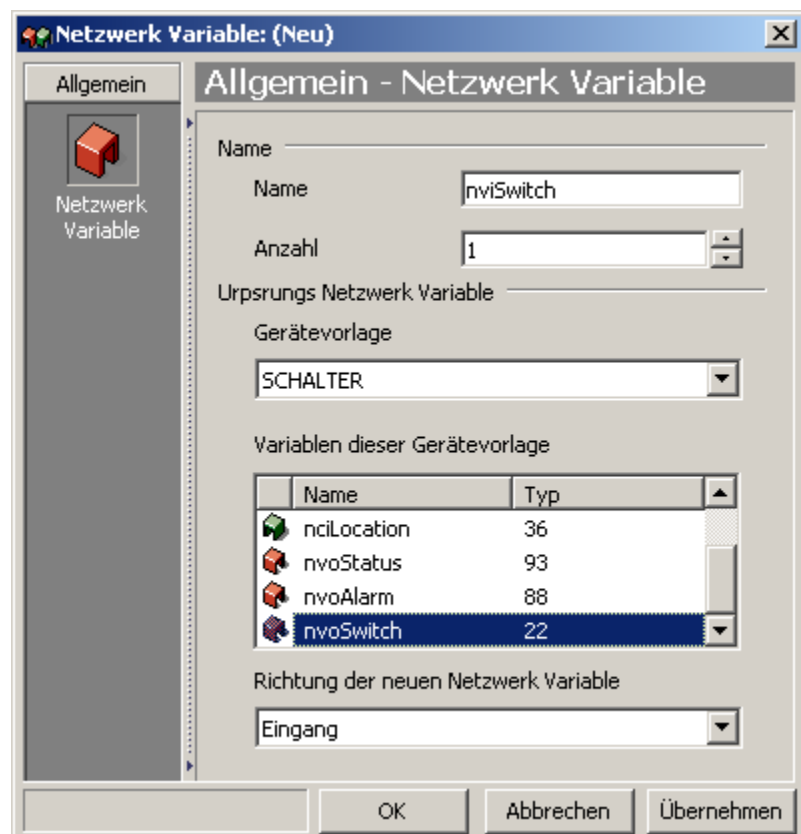


Figure 14: New network variable dialog

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 15).

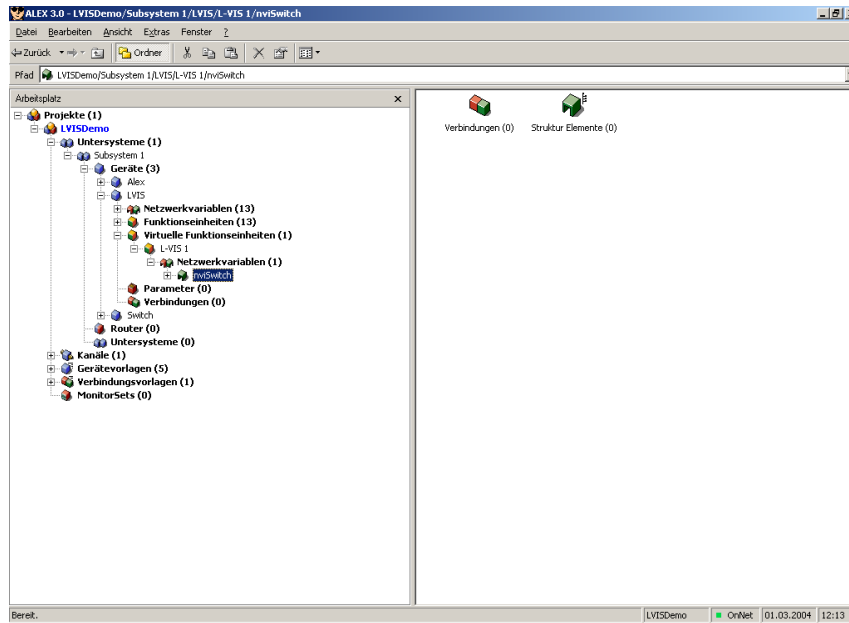


Figure 15: Finished creation of new network variable

11.5 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.

11.5.1 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.

11.5.2 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.

11.5.3 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.

11.5.4 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.

12 Special Functions (BACnet)

12.1 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.*

12.2 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. Currently, the only setting available is the desired character code to be used for string type properties of local server objects (object name and description, as well as state texts).

Since all strings in an L-VIS project are Unicode strings (ISO-10646), the default setting is to use UCS-2 encoded strings for BACnet properties. However, especially in the US market, 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 can be restricted to either ISO-8859-1 (Latin-1), which is a subset of ISO-10646 and sufficient for most western languages, or even further to ANSI X3.4 (US-ASCII), for tools which are severely limited in their ability to process string data types. If the character set is reduced by this option, the user must ensure that no foreign characters are used for server object names and descriptions. Unprintable characters will be replaced by a question mark.

13 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.

13.1 Basic Settings

On this property page, the basic setup is performed. The areas of the property page are explained in more detail below.

13.1.1 Page Protection

In this area, all options related to access control can be configured. Access control is described in detail in section 14.1. 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.

13.1.2 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 menu after:** The amount of idle time after which an open navigation menu 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.

13.1.3 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.

13.1.4 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 8.4.1 for more information about this feature.

13.1.5 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.

13.2 Extended Settings

On this property page, some less frequently used options are available.

13.2.1 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 6.3 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.

13.2.2 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.
- **Keypad Button Size:** Specifies the desired size of the keypad buttons on the display (width x height) in millimeters.

13.2.3 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.

13.2.4 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.

13.3 Display

This page defines the display properties of the device.

13.3.1 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.

13.3.2 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.

13.4 Connectivity

This page collects settings related to the communication between the device and a PC.

13.4.1 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: *The display of L-VIS currently uses a VGA color map (256 colors). 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. 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 reduced. A client which supports color maps is RealVNC.*

13.4.2 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.

13.4.3 Default OPC Access

This area defines parameters used for polling of currently active OPC data points. An active data point is one which is either displayed by a control located on the current page, or a data point which is connected to a trigger, action, trend, or formula and thus must be monitored all the time. There are two modes of operation available:

- **Polling:** Data is requested at a regular interval, which can be specified in the input field called **Pollcycle**.
- **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 if there is no value change to report during the hold time. It defines how much longer 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.

13.4.4 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.

13.5 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. Other types of authentication, such as SSL/TLS or POP-before-SMTP are currently not available and are usually only required by free mail service providers, which have no other way of authenticating their users. Standard ISP mail service usually 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.

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:	<i>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.</i>
--------------	--

13.6 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.

14 Advanced Topics

This chapter discusses some of the more advanced topics.

14.1 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.
- On a normal project 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.

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.

14.2 Undo / Redo

The current version of the configuration software provides a simple form of undo and redo. To make use of this feature, it is required to take a snapshot of the project from time to time by pressing the *Snapshot* button in the tool bar, using the *Snapshot* command from the *Edit* menu, or the keyboard shortcut (Ctrl+A). Typically, a snapshot will be taken before major changes are done to the project.

Once there is at least one snapshot in the internal snapshot buffer, the undo and redo functions become available. Using the *Undo* function, the project can be restored to the previous state. Using *Redo*, the project can be restored to the next state (the state it was in before undo was executed). With multiple snapshots of the project in the buffer, it is possible to move back and forth between the individual versions, without having to save each of these versions to a temporary file on the PC. The software will keep the last 32 snapshots, removing older snapshots when new snapshots are taken.

NOTE: *The snapshot buffer will be cleared when the project or the configuration software is closed. If a certain state of the project should be saved for later reference, it must still be written to a file on the PC.*

14.3 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: *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.*

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.

14.3.1 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.

14.3.2 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.

14.3.3 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.

14.4 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 14.18.3 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.

14.5 Find Similar Objects

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.

14.6 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.

14.7 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 7.2.9 and 7.2.10. 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).

14.7.1 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.
- **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.

14.7.2 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.

14.7.3 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	<i>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.</i>
-------------	---

14.8 Preview and Edit Mode

Some designs want to hide the selection frame from the user when a control is selected on the device. To archive this, the selection frame color is set to the background color, which in turn is set invisible in order to hide the selection frame.

This may cause problems when editing the project in the configuration software, because the LCD preview normally shows the behavior of the device, therefore such a selection frame is invisible in the editor as well.

To force the editor to ignore the device behavior and always draw a visible selection frame when a control is selected in the editor, the button *Edit or Preview Mode*, which is a toggle button, can be pressed in the tool bar, or the option *Edit Mode* may be checked in the *Edit* menu of the main window.

14.9 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: *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 same device as a remote network interface at the same time.*

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.

14.10 Local Data Point Capacity

BACnet models are currently fixed to a maximum number of 512 local server objects. 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.

CEA-709 models allow setting 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.

14.11 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. This enables the user to create or edit a project, which is intended to run on an older firmware version, even when the firmware version to use cannot be queried from the device. The real time LCD preview is automatically adjusted to match the behavior of the selected model and firmware version. This way, the preview is always accurate down to the pixel level (what you see in the preview is exactly what the device will display, pixel by pixel).

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 or 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: *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 version it knows. 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.*

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 set the CLEAR jumper at the back of the device and reboot the device 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: *While older versions of the firmware also allowed uploading the current firmware from the device, this is not possible anymore in current firmware versions, due to technical limitations. It is therefore suggested to keep a copy of the downloaded firmware somewhere on a PC, in case the device needs to be set back to this version at a later time.*

14.12 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.

14.13 User loaded Fonts

Every device has a set of three basic fonts which cover the ISO8859-1 (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 PCF fonts to cover Western (ISO8859), CJK (Chinese, Japanese, and Korean) and also Unicode (UCS-2) character sets. These should cover your immediate needs for a large number of languages.

NOTE:	<i>To open a font from the Windows system font directory, the font or the entire directory must be copied to some other location first, since the file open dialog recognizes the operating systems font directory and switches to a special mode, in which normal selection of files from a list is not possible. Since the application cannot disable this behavior, the files must be copied to some other place, from which they can be opened as usual. Alternatively, the font can be added to the list of loaded fonts via drag & drop of the font into the font manager window.</i>
--------------	---

The provided PCF fonts are fixed size fonts, meaning that each of them is designed and optimized to a specific resolution. Most fonts are therefore available in a number of variants, starting at very small sizes, up to about 24 pixel resolution. For larger sizes or other designs you may load vector fonts as well, in which case you will have to specify a font size in points (1/72 inch) and the configuration software will render the font in the desired size for you. If the loaded font file is a font catalog (for example *.ttc), you may also select the desired face which should be rendered.

NOTE:	<i>Vector fonts on the 12" and 15" devices are rendered with anti aliasing technique. This may lead to faded line segments when the line width falls below the LCD resolution at small font sizes or. 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>
--------------	--

14.13.1 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, and the source file from which the font was originally imported or where it was found the last time it was searched.

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

- **Add:** Opens a file dialog to select a font file. The new font will be added to the list of loaded fonts if not already loaded. In case of a vector font, an additional dialog will appear, asking for the desired size of the font. The same font can be loaded multiple times in different sizes.

- **Reload:** This will try to reload all selected fonts from their currently known source 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.
- **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 no matter in which environment the project is opened. 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 font sources to trade off project size with project portability.

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.

14.13.2 Font File Location

When a project contains a user defined font, it is necessary to have access to the original font file when working with the project. This is guaranteed as long as only the pre-installed fonts are used, since they 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 at 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 full path from where the font was built the last time.
- The directory containing the project file.
- 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.

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. For example, create a folder to store the project file together with all custom font files in one place. Now the entire folder may be copied to any other PC and the required font files will always be found automatically.

14.13.3 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 the 16-bit international Unicode standard UCS-16. PCF and BFD fonts may use other encodings, such as ISO8859, JISX0212, Big5, IBM, or Microsoft codepages (for example MS-CP1250). If such a font is loaded, it is automatically recoded to UCS-16 before it is used, since the L-VIS device does not recognize encodings other than UCS-16. 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 fonts.

NOTE:	<i>Fonts in other formats than BDF or PCF should contain an ISO8859-1 or ISO10646 (Unicode) codepage or Adobe Type-1 glyph names in order to facilitate recoding to UCS-16. 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.</i>
--------------	---

14.14 XML Support

Similar to the XML import/export function for individual objects, as explained in section 7.2.4, 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.

14.15 Using 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.

LVIS-3E11x and LVIS-ME21x models support true color images, which are physically displayed on the LCD in 65k colors (RGB-565 format). 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.

If a true color image is loaded, like for example a JPG, 32-bit BMP or similar, 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.

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). If the transparent background option is turned on, this data will be processed and pixels will be drawn according to their RGB color values from the main image data and the opacity values from the alpha channel. If the transparent background option is turned off, the alpha channel data is ignored and all pixels are drawn fully opaque (this is usually not desired).
- 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.

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.

14.16 Using Sounds

The 12 inch and 15 inch L-Vis devices as well as recent versions of the LWEB-800 OPC client software 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.

14.17 Project Statistics

While the total estimated memory requirements for a project are always shown in the top right of the main window, it may be difficult to locate the problematic parts of the project where a lot of resources are used, when it comes to project optimization.

To help the project developer find the most critical parts of the project, where optimization may be possible or needed, a project statistics dialog is available via the *Project Statistics...* command of the *File* menu.

This dialog shows two lists. The first one is a list of all the different types of objects. For each type, the number of instances is shown together with the total resource usage. Also shown is the largest item with its size and location in the project.

The second list shows objects sorted by size, where each objects size is the total size of it and all child objects. Therefore, this list will most likely start with the root menu and continue with the largest menu items and largest pages and so on. This list makes it easy to find the menu item, page, or control in the project which uses the most resources.

14.18 Format strings

This chapter describes the various types of format strings, as well as regular expressions, which are used by the configuration software.

14.18.1 Number format strings

The format string used in text controls and 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. The L-VIS device uses '.' as radix character, and does not have a grouping character.

14.18.1.1 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.

14.18.1.2 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.

14.18.1.3 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.

14.18.1.4 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.

14.18.1.5 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.

14.18.2 Date format strings

The format string used by L-VIS date controls is equivalent to the format string used by the ANSI-C function `strftime`. The locale setting on the L-VIS device for the purpose of

formatting dates is fixed to U.S. setting, therefore the names of weekdays and months will be in English language only.

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.

14.18.3 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.

14.18.3.1 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:

- | | |
|--------------------|---|
| <code>?</code> | The preceding item is optional and matched at most once. |
| <code>*</code> | The preceding item will be matched zero or more times. |
| <code>+</code> | The preceding item will be matched one or more times. |
| <code>{n}</code> | The preceding item is matched exactly <code>n</code> times. |
| <code>{n, }</code> | The preceding item is matched <code>n</code> or more times. |

{n,m} The preceding item is matched at least **n** times, but not more than **m** 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 **{ }**.

14.18.3.2 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.

14.19 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.

14.19.1 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 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.

14.19.2 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.

Required Ports:

- 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 is an LVIS-3E1xx and the IP852 interface is enabled. The port can be changed.

Optional ports not necessary for the primary product function:

- 21 tcp: This port is opened by the FTP server. The port can be changed and disabled.
- 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.
- 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.
- 1629 udp/tcp: This is the configuration server port of CEA-852. Exactly one device in the system 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 LVIS-3E1xx models running on the FT-10 interface (IP852 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.

14.19.3 Services

Possibly required services:

- LVIS-3E1xx 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.
- LVIS-ME2xx with BACnet/IP interface enabled: BACnet/IP. This service is in accordance with the standard ANSI/ASHRAE 135-2010.

Optional services not necessary for the primary product function:

- 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, available on LVIS-3E11x/ME21x devices only. It can be used for a secure connection to the device by the configuration software for configuration and firmware upgrade. On devices that do not have SSH, Telnet/FTP must be used instead.
- FTP and Telnet: The FTP and Telnet servers are used for connection to the device by the configuration software for configuration, firmware upgrade, and access to the log file. On devices without SSH server, these services must be enabled during device configuration.
- 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 LVIS-3E1xx devices running on the FT-10 interface. It is also used by the remote LPA feature. If these features are not needed the service can be disabled.

14.19.4 Logging and Auditing

The device contains a log file which can be read out via FTP, the configuration software, or the Web server. 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 on LVIS-3E11x/ME21x devices 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.

14.20 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
Total number of data points	~5,000	~5,000	~10,000	~10,000
OPC Tags	~2,000	~2,000	~10,000	~10,000
User Registers	~2,000	~2,000	~5,000	~5,000
NVs (static, dynamic)	1,000	n/a	1,000	n/a
External NVs	~1,000	n/a	~2,000	n/a
Alias NVs (ECS and CEA709.1-A mode)	1,000	n/a	1,000	n/a
Address table entries (15 for CEA709.1-A)	524	n/a	524	n/a
LONMARK Calendar objects/patterns	1/100	n/a	1/100	n/a
LONMARK Scheduler objects	100	n/a	100	n/a
LONMARK Alarm Servers	1	n/a	1	n/a
BACnet objects (analog, binary, multi-state)	n/a	512	n/a	512
BACnet client mappings	n/a	~2,000	n/a	~2,000
BACnet scheduler objects	n/a	100	n/a	100
BACnet calendar objects	n/a	25	n/a	25
BACnet notification classes (alarm server)	n/a	32	n/a	32
Trend Logs	100	100	512	512
Total trended data points	256	256	512	512
Total aggregated size	6MB	6MB	60MB	60MB
E-mail templates	~100	~100	~100	~100
Math objects	~500	~500	~2,000	~2,000
Alarm Logs	~20	~20	~100	~100
Connections (local)	~1,000	~1,000	~2,000	~2,000
Connections (global)	250	250	250	250
L-WEB Clients	20	20	20	20

15 Appendix A: Examples

This section shows how to use the different controls in a sample project. The project can be found in the *Projects* directory of the L-VIS configuration software install directory (Users_Manual_Color.lcp).

15.1 Example 1: Working with Structured Network Variables

This example describes how to work with structured network variables like SNVT_switch. It explains how structured input and output network variables can be used with different controls.

This example assumes that a dynamic input variable of type SNVT_switch (nviSwitch) and a dynamic output network variable of type SNVT_switch (nvoSwitch) is already present on the node. We will use the output variable to update the value on the network and will use the input variable as a feedback value from the network. The example explains how to use this setup with different controls.

15.1.1 Text control

With this example, we will be able to show the state of a SNVT_switch (e.g. a lamp) and control the state from the L-VIS device, using a text control. Text controls are typically used to display a complex text or to allow input where the user can select between several different input strings (e.g. HIGH-MID-LOW).

To use the SNVT_switch variables with a text control, perform the following steps. The control only changes the state value of the output network variable and fixes the value element of the network variable to 100%.

- Add a text control to the page
- In the common properties tab, add a label to the format string in the **Text** field, (e.g. Switch: %s).
- Add data points to the control. To do this, select *Add Data Point* from the context menu of the control. Add an input data point for nviSwitch.state and output data points for nvoSwitch.state and nvoSwitch.value. A default mapping between the state input values and the displayed text is automatically added to the control.
- For the input data point (nviSwitch.state), set the **System Startup** flag on the **Data Point** configuration tab.
- For the nvoSwitch.value network variable, go to the **Data Point** configuration tab, configure the **Default Value** to 100 and set the **Constant Value** flag. Set the **NV Update Flags** value to **Focus Loss**.

- For the nvoSwitch.state network variable, set the **NV Update Flags** value to **Focus Loss**.

The configuration now looks similar to Figure 16. The project can now be loaded into the L-VIS device.

To change the state of the output network variable, select the text control by operating the jog dial until a frame is displayed around the text control. Select the control with a short click.. Now the frame starts flashing which signals that the control is in input mode. Turn the jog dial to change the state. Pressing the jog dial again deselects the text control.

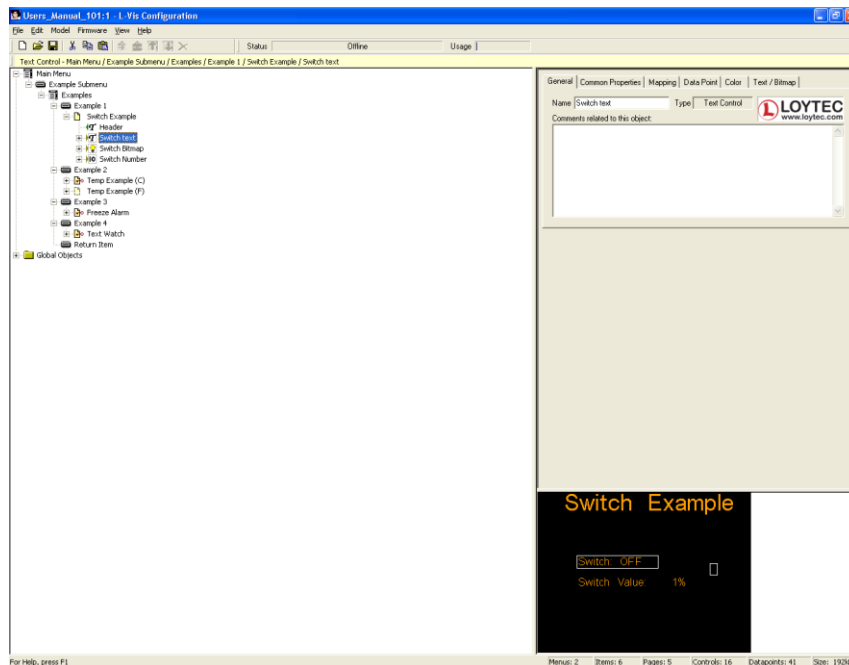


Figure 16: SNVT_switch with a text control

15.1.2 Bitmap Control

Bitmap controls are typically used for information which has only a few states (like heating on/off, light on/off).

This example shows how to use a SNVT_switch together with a bitmap control. It also fixes the value part of the SNVT_switch network variable to 100 and toggles only the state part.

- Add a bitmap control to the page
- Add data points to the control. To do this, select *Add Data Point* from the context menu of the control. Add an input data point for nviSwitch.state and output data points for nvoSwitch.state and nvoSwitch.value. A default mapping between the state input values and the displayed text is automatically added to the control.
- For the input data point (nviSwitch.state), set the **System Startup** flag on the **Data Point** configuration tab.
- For the nvoSwitch.value network variable, go to the **Data Point** configuration tab, configure the **Default Value** to 100 and set the **Constant Value** flag. Set the **NV Update Flags** value to **Focus Loss**.

- For the nvoSwitch.state network variable, set the **NV Update Flags** value to **Focus Loss**.
- Select the mapping item in the tree view and go to the **Mapping** tab. In the value list, select OFF. Click on **Select Bitmap** and choose a bitmap which should be displayed when the switch is OFF. After this, select ON in the value list and select the bitmap for the ON state.

The configuration now looks similar to Figure 17. The project can now be loaded into the L-VIS device.

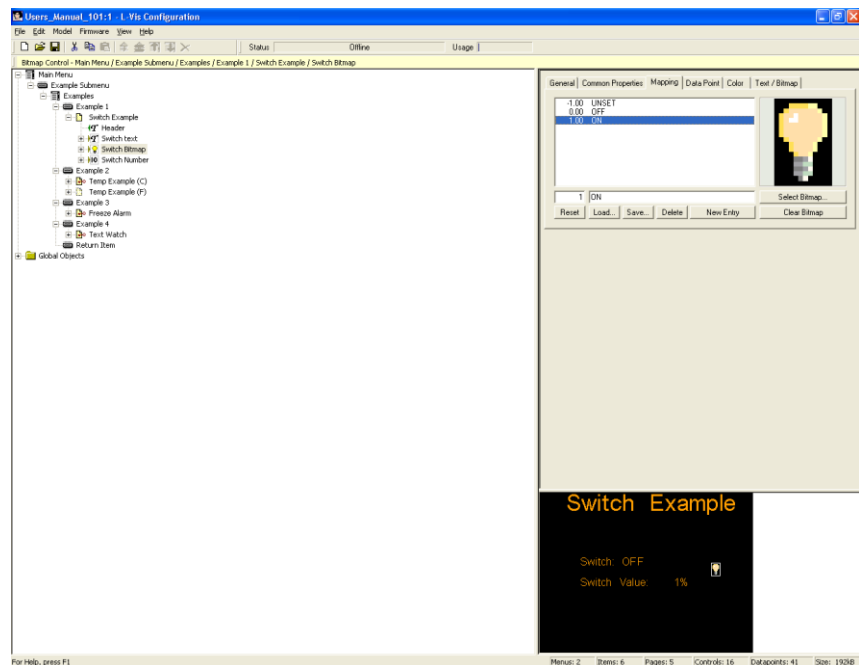


Figure 17: SNVT_switch on a bitmap control

15.1.3 Numeric Control

A numeric control is used whenever a data point needs a continuous numerical output or input, like a percentage value or a temperature value.

This example shows how to use a SNVT_switch together with a numeric control. The state element of the SNVT_switch network variable is fixed to 1, whereas the value part can be set between 0 and 100.

- Add a numeric control to the page
- Add data points to the control. To do this, select *Add Data Point* from the context menu of the control. Add an input data point for nviSwitch.value and output data points for nvoSwitch.state and nvoSwitch.value. A default mapping between the state input values and the displayed text is automatically added to the control.
- For the input data point (nviSwitch.value), set the **System Startup** flag on the **Data Point** configuration tab.
- For the nvoSwitch.state data point, go to the **Data Point** configuration tab, configure the **Default Value** to 1 and set the **Constant Value** flag. Set the **NV Update Flags** value to **Focus Loss**.

- For the nvoSwitch.value data point, set the **NV Update Flags** value to **Focus Loss**, adjust the **Value Range** to values between 0 and 100 and set the **Acceleration** to 2.

The configuration now looks similar to Figure 18. The project can now be loaded into the L-VIS device.

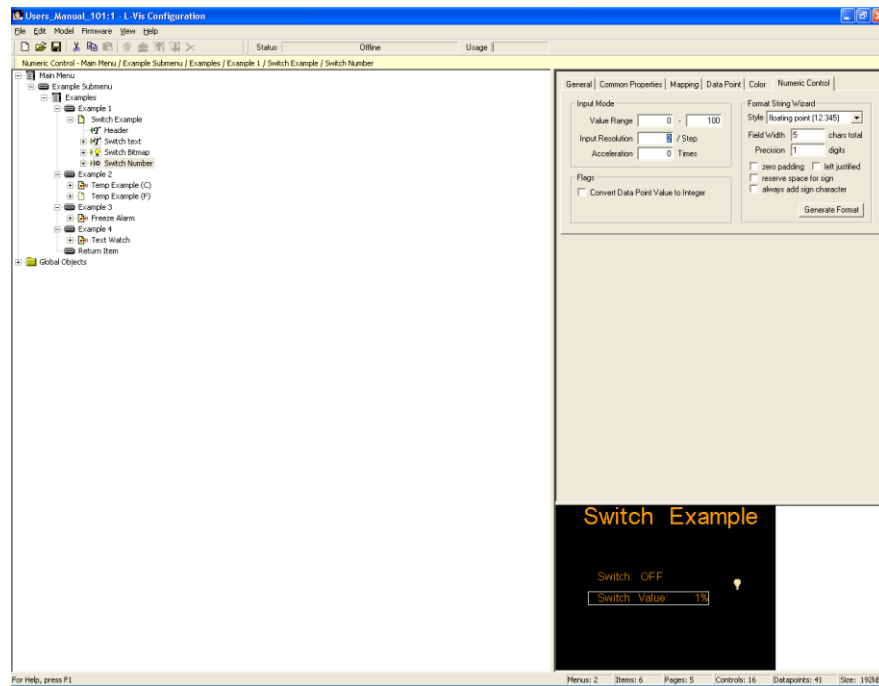


Figure 18: SNVT_switch on a numeric control

15.2 Example 2: Using Data Point conversion and Hiding of Pages

This example will use an input value of type SNVT_temp_p. It will provide the temperature value on a bar graph and a numeric control. The data will be shown in two versions, one for degree Celsius and one for Fahrenheit. The two display versions will be placed on different pages. The example will show how to use the hide flag in order to load the project into the L-VIS device with the localized configuration. The example walk-through description assumes that a dynamic network variable nviTemp of type SNVT_temp_p is already created on the L-VIS device.

- Create a menu item and add a page to the menu item. Name the page, e.g. to Temp Example (C). Add a text control to the page and enter a header text for the page (e.g. Temperature).
- On the page, create a bar control (select *Add Bar* from the context menu of the preview frame). Name the bar graph (e.g. Temp Bar) in the **General** tab.
- On the **Bar Graph** tab, enter the value range for the bar graph (e.g. -20 to +40) and set the flag for **Thermometer Style**.
- Select *Add Data Point* from the context menu of the bar graph. From the data point template list, select the nviTemp input data point.
- Make sure that the **Value Translation Setting** on the **Data Point** tab is set to OFF.

- Select the page in the tree view. In the context menu of the page, choose *Copy*. Go to the menu item (e.g. Example 2), open the context menu and choose *Paste*. A copy of the page is inserted in the menu.
- Change the name and text of the page to Temp Example (F).
- Select the input data point which is connected to the bar graph in the newly created page. Set the Value Translation Setting to °C-->°F.
- Depending on whether the Temperature should be shown in Fahrenheit or Celsius, set the **Invisible Page** flag on the **Menu/Page** tab of the corresponding page.

Load the project into the L-VIS device. Only the page, which does not have the Invisible Page flag set, shows up in the L-VIS device (see Figure 19).

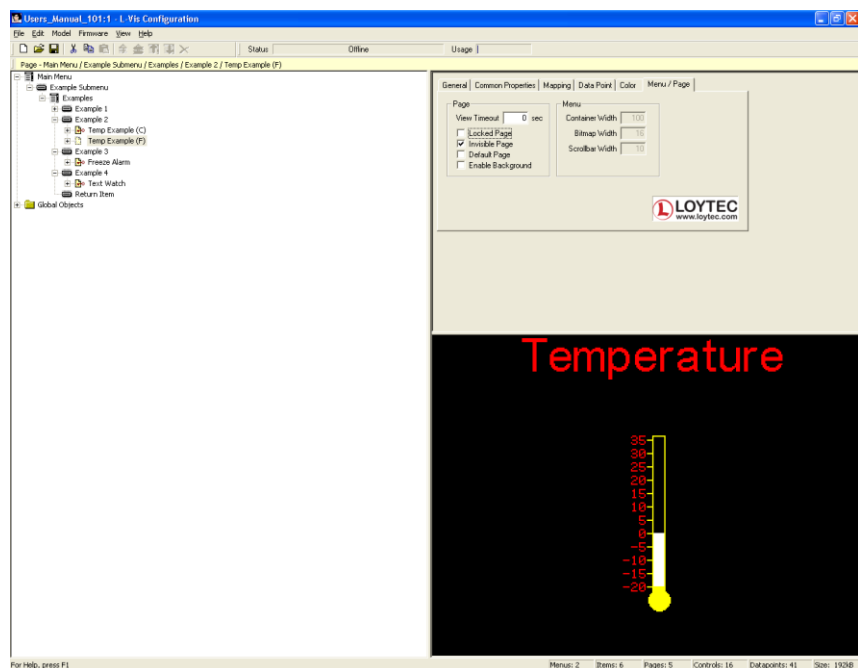


Figure 19: Hiding pages

15.3 Example 3: Freeze warning, Alarms and Registers

This example shows how to create a translation table which converts a numeric value to a text value and to generate an alarm output on special occasions. It also shows how to use registers to exchange data between internal objects. The example takes an nvoTemp input data point and checks if it is below 0°C (freeze warning) or not. It controls an output text and an alarm output register of type SNVT_alarm_2.

Generate a new menu item (Example 3) and a page for this menu item and name the menu item and the page. Generate a header for the page (Freeze Warning).

Add a new text control to the page. Set a label for the text control (e.g. Temperature State: %) in the text field of the **Common Properties** tab.

Add the nviTemp data point to the text control. Select the **Mapping** tab on the data point. Select New, enter 0 for the value and FREEZE for the text. Add two more mapping values (0,1, ATTENTION and 3, NORMAL) to the mapping table. This will cause that for input

values up to 0,1°C, the FREEZE text will be shown. Values between 0,1°C and 2,99°C will show ATTENTION, where as values starting with 3°C will show NORMAL in the text control. Figure 20 shows a screenshot of the mapping table.

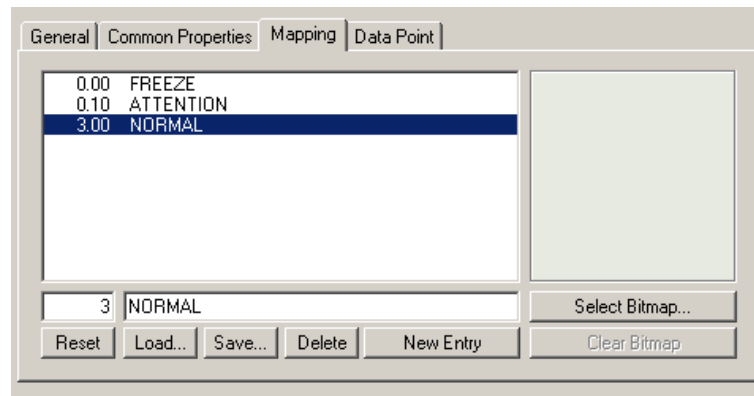


Figure 20: Mapping table

In the tree view, open the context menu of **Global Objects** and add an alarm generator. Label the alarm generator with 'Freeze Alarm'.

On the context menu of the alarm generator, select *Add Data Point*. Since we have no output network variable of type SNVT_alarm_2, we will generate a register, which stores the value of the alarm generator. To create a new register, click on the *New* button, and edit the field on the bottom of the dialog window. For the **Name**, enter 'rwFreezeAlarm', for **Type** select 'Reg.Write'. Change the **Name** in the Register Properties field to 'FreezeAlarm' and set the **Type** to alarm_2(164). For Element keep the value '*'. Select the new element in the list and click on OK.

Select *Add Data Point* again on the context menu of the alarm generator and add the nviTemp data point to the alarm generator.

In the **Alarm Generator** tab (Figure 21) of the alarm generator, set the alarm conditions for the alarm. We will trigger an alarm when the temperature falls below 3°C. For the **Data Point Value Range**, set the minimum value to 3. For the **Alarm Condition**, select *Below Min.* In the **Action** field select *Set and Clear Alarm*. For **Alarm Set Type** select AL_ALM_CONDITION. For **Alarm Clear Type** select AL_NO_CONDITION. In the **Description** field, enter 'Freeze Alarm'. The different alarm types and their meaning can be found in the LONMARK® description of the SNVT_alarm_2 network variable type.

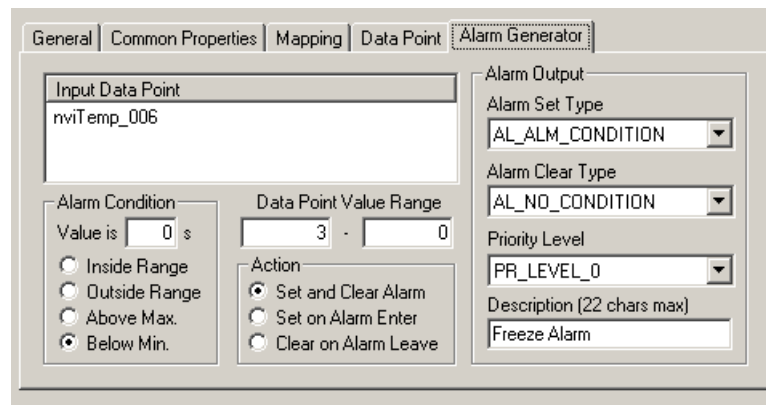


Figure 21: Alarm Generator configuration tab

To display the alarm output we will draw a bitmap which shows an ice symbol in alarm state. To do this, add a bitmap control to the page. Enter a label for the bitmap in the **Text** box of the **Common Properties** tab.

Select *Add Data Point* from the context menu of the bitmap control. Add a Reg.Read data point as described above and use the same name (FreezeAlarm) for the **Name** in the **Register Properties** field. Set the **Type** to alarm_2(164) and select alarm_type for the **Element**.

On the **Mapping** tab of the data point, select the AL_ALM_CONDITION. Click on Select Bitmap and choose a bitmap which is displayed when the alarm condition is met.

Load the project into the L-VIS device.

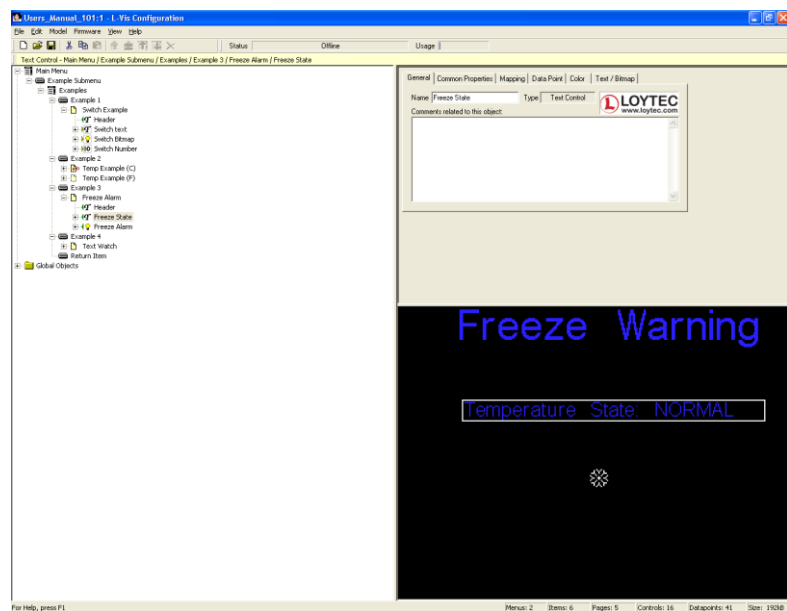


Figure 22: Freeze alarm with text and bitmap controls

15.4 Example 4: Using text mapping and Date Control

This example describes how to integrate a date control into a project. The date control will display an analog clock. It will get the time value from the internal real time clock. In addition, three text fields will be created which display the actual time in a textual format. The time value for this text will be received from a turnaround binding of the real time keeper object.

Create a new menu item and assigned a page to this menu item. On the page, add a text control for the header text, name it (e.g. 'Header' and enter a text in the **Text** field of the **Common Properties** tab (e.g. 'Text Watch')

In the context menu of the new page, select **Add Date**.

In the context menu of the date control, select **Add Data Point**. In the dialog window, select the System Time parameter and click on OK.

On the Date/Time Control tab, select *Graphic* for the **Style** and choose a Graphic Style.

For the text display, we need a turnaround bound dynamic network variable which allows to access hours, minutes and seconds separately. Create a dynamic network variable of

type SNVT_time_stamp and bind the nvoDateTime variable of the L-VIS device to this input variable (see Figure 23).

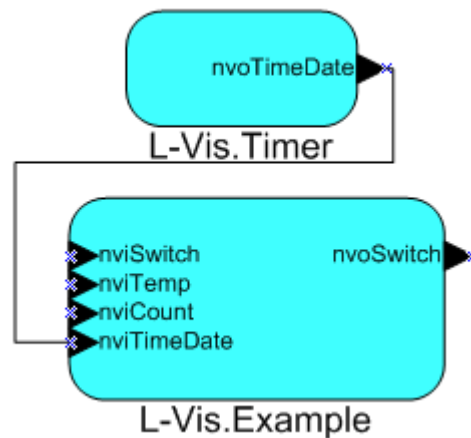


Figure 23: Turnaround binding for the time network variables

Add a text control to the page and name it 'Hour'. Use the context menu of the text control to add a data point. Select the new time stamp input network variable and select **hour** in the **Element**.

Create two more text controls, one for the minutes and one for the seconds. Assign the minutes and seconds Elements from the time stamp input data point to the controls.

On the mapping tab of one of the text controls, create new entries for the numbers from 0 to 59 and add the text for the numbers (e.g. one for 1, two for 2,...). Save this mapping table to a file (button **Save**).

Go to the mapping tabs of the other text controls and load the stored mapping table.

Load the project to the L-VIS device. To show the correct text the update interval of the timer object has to be set to 1 second.

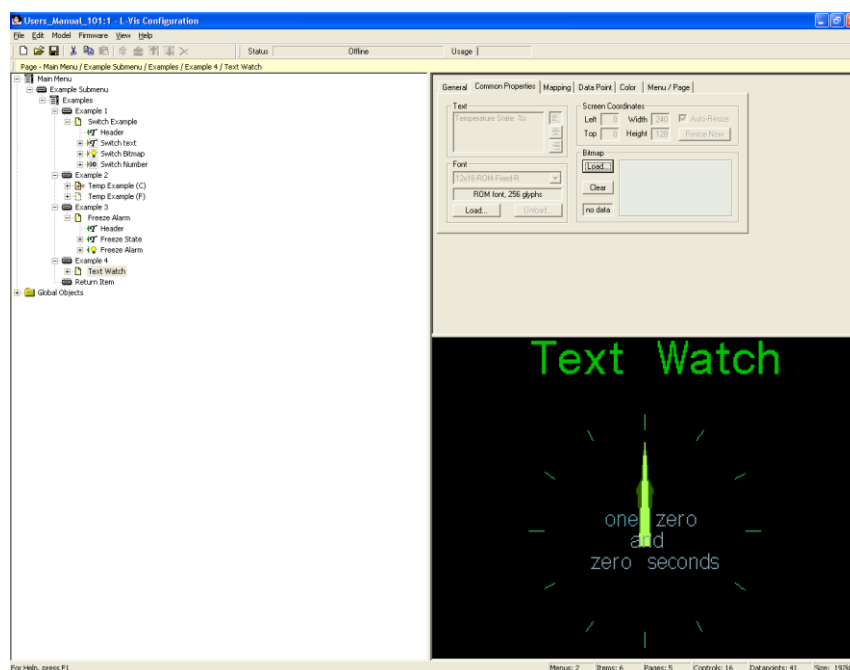


Figure 24: Text Watch project

16 Appendix B: Solutions

This section describes solutions for some standard applications. The solution projects can be found in the *Projects* directory of the L-VIS configuration software install directory.

16.1 Implementing Menus

By default, the user can navigate in a project using the main menu, submenus and push and drag operations. With the help of bitmap controls, text controls and action objects it is also possible to implement alternative ways to navigate inside an L-VIS project. The project *Menu_example1.lcp* shows different solutions.

16.1.1 Tab Menu

To switch between multiple pages, a tab menu can be implemented using bitmap controls, text controls and action objects (see Figure 25).

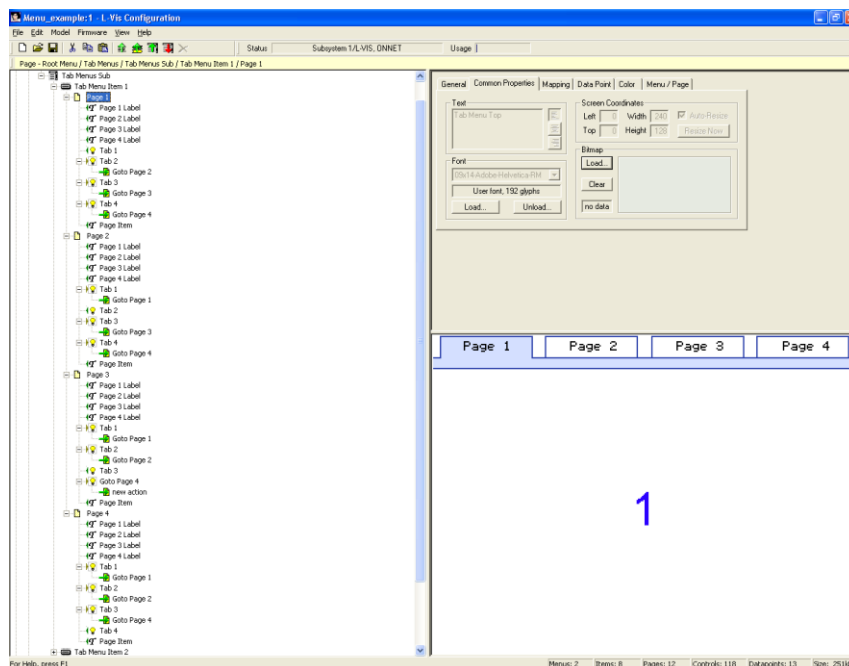


Figure 25: Page navigation using Tabs

Each page contains four bitmap controls. There are two types of bitmaps: one for the active page (Figure 26) and one for the inactive page (Figure 27).



Figure 26: Active page bitmap



Figure 27: Inactive page bitmap

The bitmaps are aligned to form a row of tabs. On each tab, a text control is placed with identifies the page and the page links (in the example it is called Page 1, Page 2,...). To each of the inactive page bitmap controls an action object is assigned. The action to execute is to change to the respective page. This setup has to be copied for each page, but the location of the active page bitmap changes according to the page on which the controls are placed.

The same mechanism can be used to build tabs on the bottom, which is also shown in the menu project (Tab Menu Bottom, Figure 28).

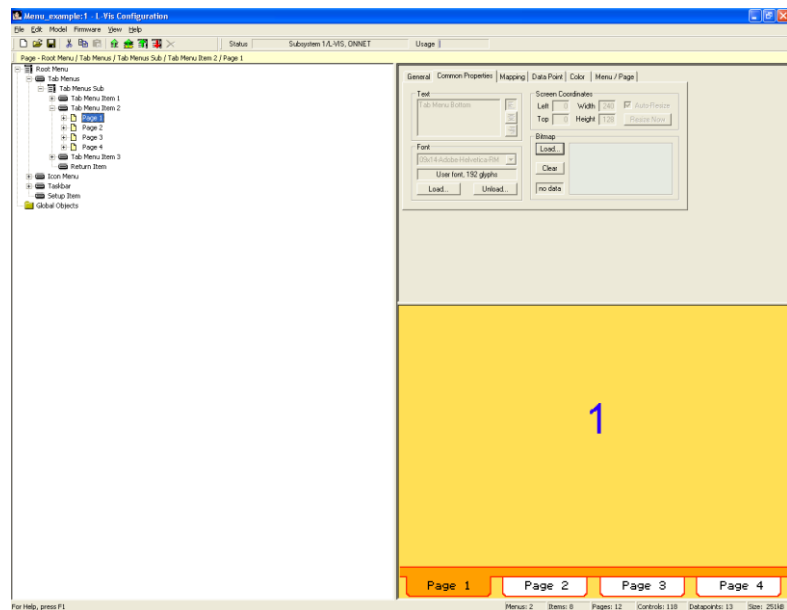


Figure 28: Bottom tab menu solution

Another tab solution in the project shows how to implement flat tabs, but using only line bitmaps. In that solution, the action objects are assigned to the text controls. The leftmost button in that case opens the main menu (Figure 29).

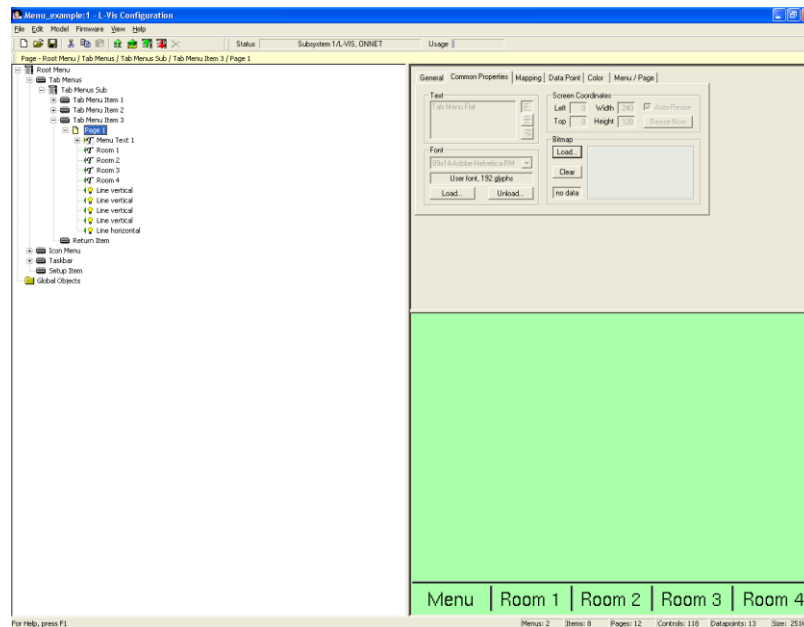


Figure 29: Flat tabs and menu button

16.1.2 Icon Menus

Another common implementation to navigate in menu trees is to use icon menus as shown in the icon menu example (see Figure 30).

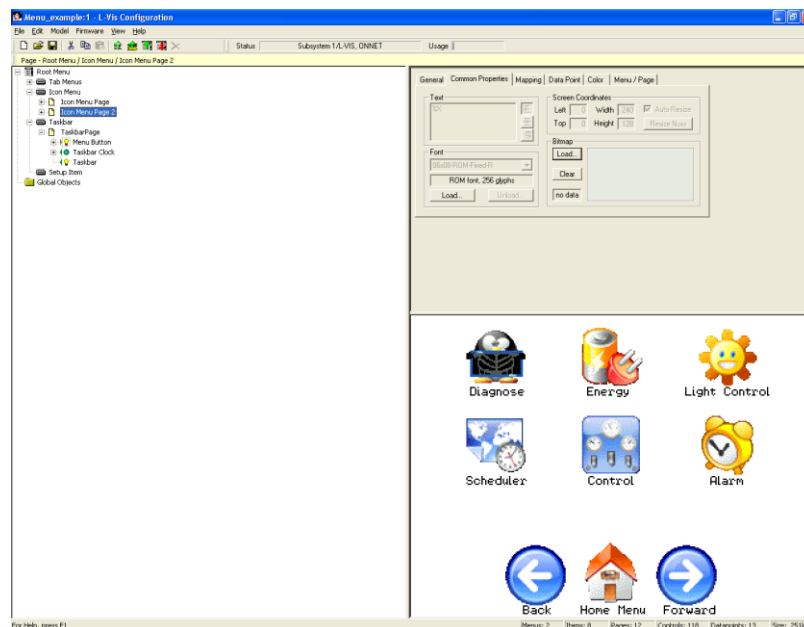


Figure 30: Icon menu

On an empty page as set of bitmap controls is placed. Each bitmap represents a page or a submenu. In the example, a text control is placed below each bitmap icon. An action object, which executes a switch to a linked page, is assigned to each bitmap control.

On the bottom of the example page, three additional bitmaps are placed, which have action assigned to switch pages back and forth and to open the system menu.

Using the icon menus, it is also possible to implement the whole navigation without the default menu. In that case, the main menu page should have the checked the default page checkbox in the Menu/Page tab.

16.1.3 Taskbar Menu

To simulate a PC environment, a system with a setup with a task bar can be created (Figure 31).

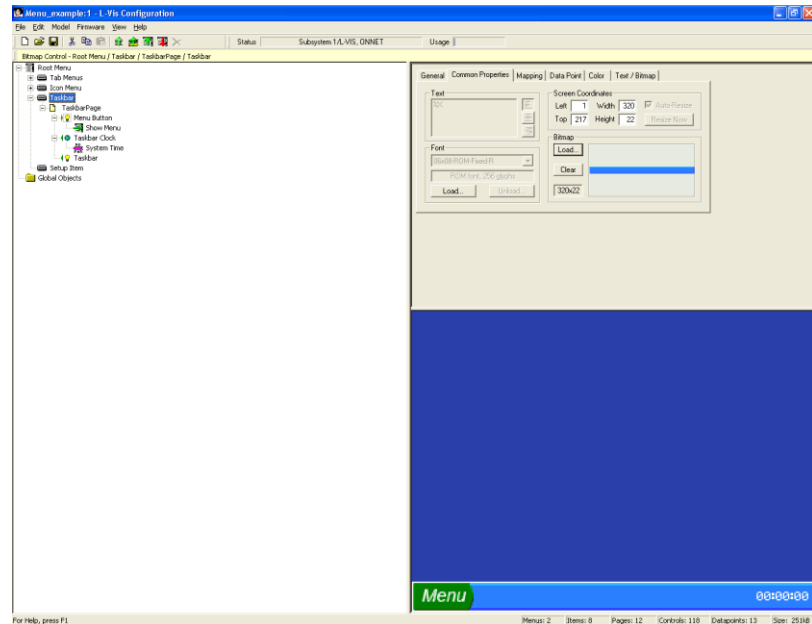


Figure 31: Taskbar menu

The taskbar is formed by a bitmap control. The bitmap for the menu button is placed on top of the taskbar bitmap. An action object is assigned to the menu button bitmap. This action object opens the main menu. In the example a date control which displays the current time is placed on the task bar.

It would also be possible to add further icons to the taskbar, which open other pages. Also, icons which are only displayed when an alarm occurs could be added.

16.1.4 Bitmap Menu

The bitmap menu shows a navigation using full screen bitmaps with touch sensitive zones. On the main page, the Floor 1 can be selected, and on the floor page the upper left room. The back button brings the user one level back.

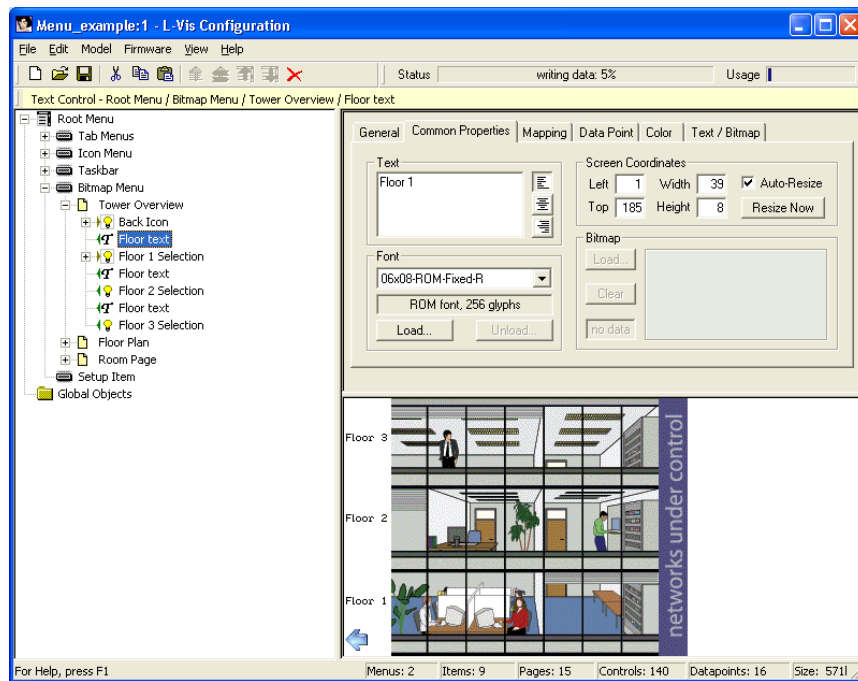


Figure 32: Bitmap menu configuration

16.2 Light Control

The project “Solution_examples_manual.lcp” contains a “Switch Demo” page.

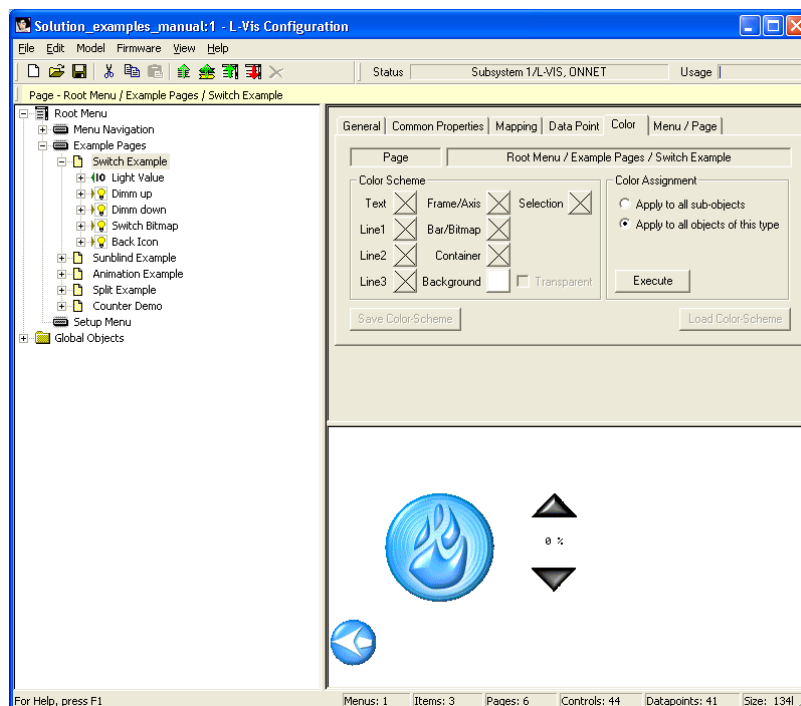


Figure 33: Switch demo page

The example uses a bitmap control in push button mode to switch the state of a SNVT_switch on or off. Two arrows allow the user to increase or decrease the value part of the SNVT_switch variable. To increase and decrease the values, action controls are

assigned to the bitmap controls. The increase and decrease operations are executed by two mathematic objects. The operation results are exchanged between the mathematic object and the bitmap controls using a register.

16.3 Animated Bitmaps

The project “Solution_examples_manual.lcp” contains an “Animation Demo” page.

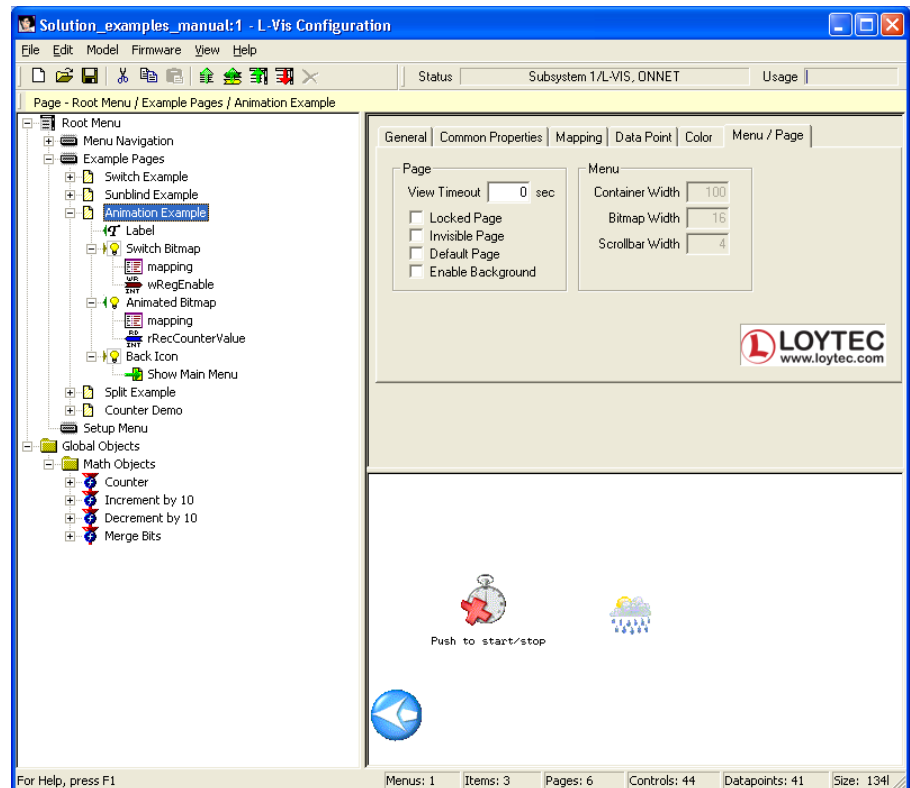


Figure 34: Animation Demo configuration

The animation demo shows how to generate animated bitmap controls.

A bitmap control in push button mode starts and stops the animation. The animated bitmap is built of a bitmap control together with a mapping table. The input for the animated bitmap is generated by a mathematic object. The mathematic object is feed by the system time and performs a modulo operation on this input. An additional enable register parameter allows enabling or disabling this modulo operation.

16.4 Sunblind Demo

The project “Solution_examples_manual.lcp” contains a “Sunblind Demo” page.

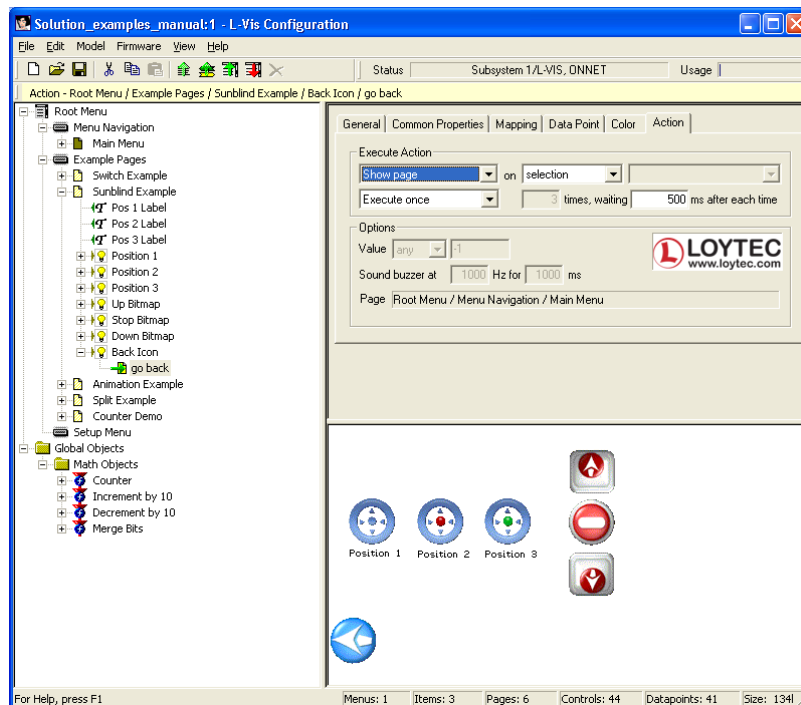


Figure 35: Sunblind demo configuration

Like in the switch example, the sunblind demo uses bitmap controls together with action objects to move a sunblind to special positions or move the sunblind up, down or stop the sunblind.

16.5 Split Demo

The project “Solution_examples_manual.lcp” contains a “Split Demo” page.

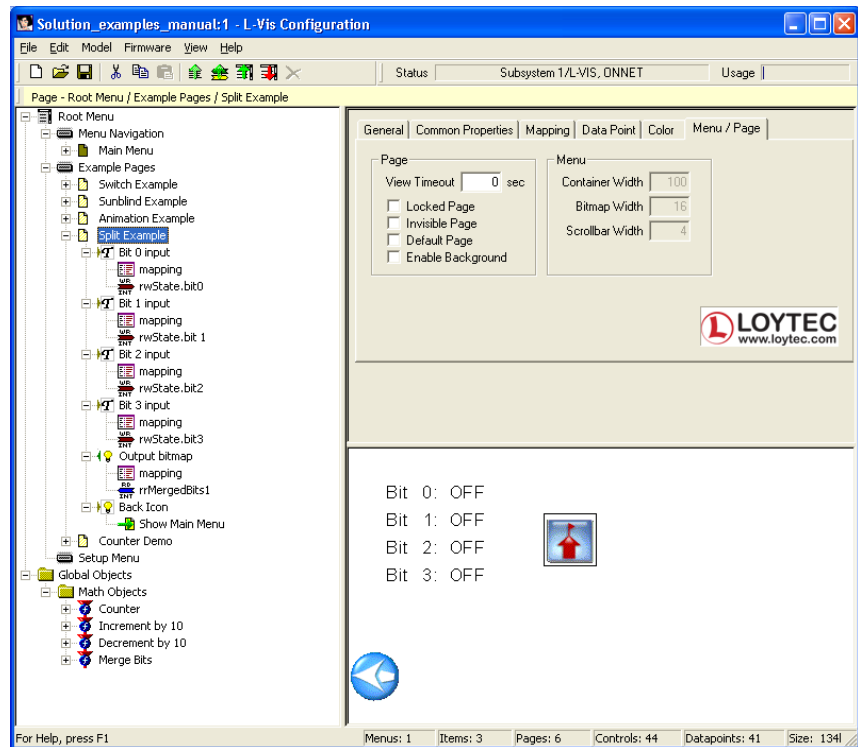


Figure 36: Split example configuration

The split demo shows how to merge the input of a SNVT_state variable into a single bitmap control. In some projects it is required to evaluate the input of a SNVT_state variable, where only one bit is set, and display the result in a single bitmap or text control. The example uses the single bits of the SNVT_state variable to feed a mathematic object. The mathematic object encodes the bit to a single numeric value and passes the result to a register. The register feeds the mapping table of a bitmap control, which displays different bitmaps according to the register contents.

16.6 Counter Demo

The project “Solution_examples_manual.lcp” contains a “Counter Demo” page.

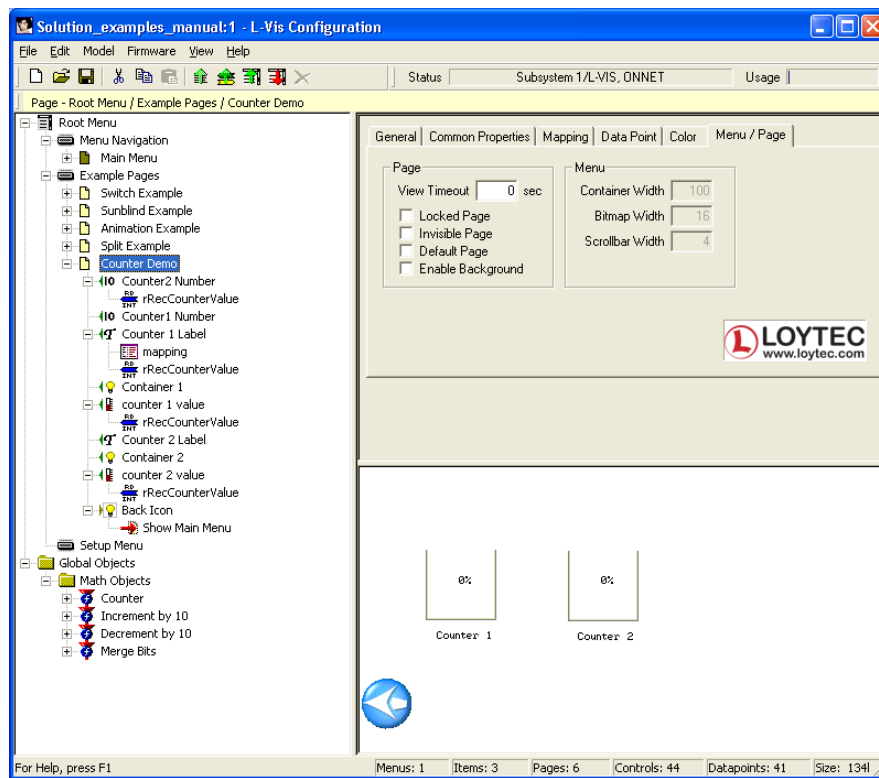


Figure 37: Counter demo configuration

The counter demo shows how to use a bar graph control to simulate the fill level of a container. An empty container is formed by a bitmap control. On top of the bitmap control, a bar graph control is placed. All relevant colors except the Bar/Bitmap color are set transparent, so that the bar graph forms the filling of the container.

17 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