1 Document Scope

A series of new LOYTEC products allows the user to configure the network variable interface of the devices according to the project requirements. These devices include visualization devices (L-Vis 3E100) as well as the LOYTEC gateway family (L-Gate, L-Proxy, L-OPC). LNS based tools provide the possibility to change the device interface also for devices which are already installed in an LNS data base. However, there are a number of rules to follow when using changing interfaces together with LNS. This document explains the basic terms such as static/dynamic network variables, XIF File and Program ID and how these terms fit together. Understanding the "big picture" helps the user during the project integration to work efficient, without running into problems.

The document assumes that the reader is familiar with the basic terms of EIA-709 networking (e.g. nodes, bindings, network variables) and has already worked with LNS based network management tools.

2 Terms

2.1 Program ID

The program ID is an 8 byte wide ID and identifies the application in the device. The structure of the program ID is defined by LonMark. An exhaustive definition of this ID can be found on [1].

The 16 hex digits are organized as 6 fields:

FM:MM:MM:CC:CC:UU:TT:NN

„F“defines the format of the ID, where „8“ stands for LonMark certified devices and „9“ for devices which comply with the LonMark guidelines but are not (yet) certified. All other values indicate that the program ID does not comply with the described format.

“M” identifies the manufacturer of the device. A list of manufacturer IDs can be found on [1].

“C” indicates the (basic) device class of the device which describes the primary function of the device. The device classes are also listed on [1].

“U” shows the usage ID of the device. This field consists of two flags (changeable interface flag, functional-profile specific flag) as well as a 6 digit usage number. Please refer to [1] for further details.

“TT” identifies the channel type of the device (e.g. FT-10, IP-852L) It enables the management tool to check if the device is compatible with the type of channel on which the device is placed. This can help avoiding user errors during the installation process.

“NN” ist the model number of the application. It defines the software or firmware revision of the product. Whenever a new software version or, for devices like L-Vis, L-Gate, L-OPC and L-Proxy, a new project version with a changed static network variable interface is loaded into the device, the model number must be changed in order to signal the management tool that the interface has changed.

Together with the node ID, the program ID is sent out in the service pin message of a device. In that way, the management tool is able to detect the device and firmware/software type of the device on the network.

Whenever the external interface of a device is changed – usually this is the case when a static network variable is added or removed – the program ID has to be changed. To perform this operation, the configuration software for a device (like the L-Vis configuration software) changes the model number part of the program ID. Inside a single network, the program ID must be unique for one application. The same program ID must not be assigned to multiple different applications.

2.2 XIF File

The XIF File (External Interface File) describes the external interface of a device and the application loaded into the device. It does not contain information about the functionality of the application, but completely describes the network interface of the device. The XIF file is generated by the programming tool for the devices and is provided by the device manufacturer. Using the XIF file, the user can integrate the device in the network management tool without having physical access to the device.

A XIF File contains the following parts:

Header: The header provides information about the XIF file version as well as some timestamp and copyright information.

```
File: Interface for L-Vis (FT-10), XIF Version 4.2
Copyright (c) LOYTEC electronics GmbH
All Rights Reserved. Run on Mon Jan 01 00:00:00 1990
```

Program ID: The program ID forms the connection between the XIF File and the device. When receiving the service pin message of a device, the management can determine which XIF file matches the device.

```
90:00:A9:01:1E:84:04:97
```

Node resource information: Beside other parameters, this includes the network variable count, the address table count, some information about the network transceiver parameters as well as information about the device capabilities.

```
2 15 1 12 0 15 15 15 15 15 15 0 0 0 0 3 0 16 524 1 1 11 524 0 1 2 1 0 0 0 0
0 2 524 0 0 0 0 0
0 7 128 13 28 1412 32768 15 5 3 150 4
1 7 1 5 4 4 1 15 200 0
78125 0 0 0 0 0 0 0 0 0 0 0
90 0 240 0 0 0 40 40 0 0 0 0 0 0 0
*
```

Node self-documentation string: From this string the management tool can read which LonMark objects are implemented in the node.

```
"&3.3@0Node,3300Timer,1040Temperature,3040Relay,3200Switch,20010[8]L-Vis
;L-Vis
```

Network Variable information: The XIF File contains a record for each (static!) network variable in the node. The record contains information like the network variable name, direction (input/output), SNVT_type, default values, etc.)

```
VAR nviRequest 0 0 0 0
0 1 63 0 0 1 0 1 0 0 0 0 0
"@0|1
92 * 2
2 0 0 0 0
```

```
1 0 0 1 0
```

Configuration file information: This contains information about the device configuration properties and the default configuration values.

```
FILE CPT 0 2
"1.1;
"1,0,0\x80,22,7;
"1,0,0\x84,167,1;
"1,0,0\x84,168,1;
"1,0,0\x84,165,1;
"1,0,0\x84,166,1;
"1,1,0\x81,97,1;
"1,1,0\x81,98,2;
"1,1,0\x81,99,7;
"1,1,0\x81,100,7;
"1,1,0\x81,101,1;
"1,1,0\x84,167,1;
"1,1,0\xA4,168,1;
"1,2,0\x80,52,2;
"1,2,0\x80,49,2;
"1,2,0\x80,64,2;
"1,2,0\x80,70,2;
"1,4,0\x80,52,2;
"1,4,0\x80,49,2;
"1,4,0\x80,92,1;
"1,4,0\x80,93,1;
"1,4,0\x80,170,2;
```

A complete description of the XIF File can be found in [2]. Whenever the program is changed in a way that the contents of the XIF file have to be changed, also the program ID of the application has to be changed.

2.3 Device Template

The device template is an object inside the LNS data base. A typical network contains only a couple of different device types but uses multiple devices of a single type. The LNS data base stores the information about the (static) device interface not with every single device but holds this information inside of a device template object. For every device an application device object is created which holds a reference to a device template object.

The device template can be created in two ways, either the information is read from a XIF file or, if a new device is available, the information is directly read out from the device. The device template holds the complete information of the XIF file, including the program ID. The program ID again forms the link between the application in the device, the XIF file and the device template.

When a new device is commissioned in the LNS data base, the user either has to select an existing device template which has been created before, using one of the methods described above, or has to create and assign a device template by loading a XIF file or loading the information out of the device.

The commissioning process links an application device object in the LNS data base with a physical device. This link is established using the node ID (Neuron ID) of the device. Figure

1 shows how the program ID connects the XIF file, the device template, the application device object and the physical device. The node ID builds the connection between the physical device and the application device object in the LNS data base.

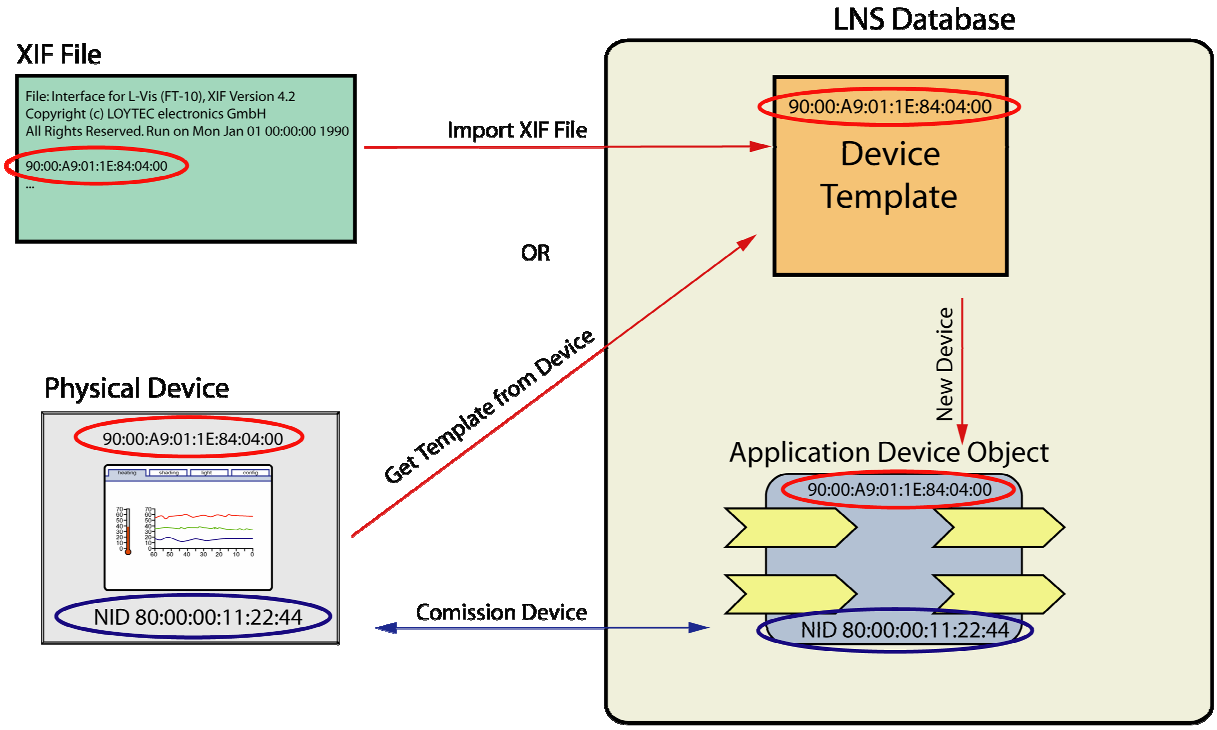

Figure 1: Meaning of Program ID and Node ID

Even when an existing device template was selected to create the application device object and the program ID of the device matches the program ID of the device template, LNS reads some interface information from the device to check if the interfaces are consistent. In case that the interfaces do not match (e.g. different network variable count, changed network variable names), LNS throws an error message ("Program interface does not match", or "There is an incompatible or outdated program version in the device"). Since the program ID uniquely identifies an application, there must be only one device template with a single program ID in one LNS Database. **It is not possible to have multiple device templates with the same program ID!**

2.4 Static Network Variables

Static network variables are a fixed part of the application. All static network variables are also described in the XIF file and are a fixed part of the device template object. All devices with the same program ID must contain the same static network variables.

2.5 Dynamic Network Variables

Dynamic network variables are created on the device during run time using network management commands. The only information about dynamic network variables stored in the XIF file and the device template is the **maximum possible count** of dynamic NVs in the

node. Neither the actual count nor the actual dynamic NVs is part of the device template and XIF file. This information is stored in the application device object. The actual dynamic network variables can be different for every application device, even if the application devices refer to the same device template. Whenever a new device which supports dynamic network variables is added to the LNS data base, LNS sends a command to clear all existing dynamic network variables rather than adding the existing dynamic network variables to the application device object. Dynamic network variables can be created on a node through the LNS data base. Usually this is done using the LNS based network management tool or by a LNS plug-in which calls LNS Api functions. What happens then is that the dynamic network network variables are added to the application device object in the LNS data base. Further, LNS sends the network management messages to the device to create the dynamic network variable in the device.

As a consequence, it is not possible to create dynamic network variables from within the device application or using configuration software which is not connected to the LNS data base. All these variables would be deleted when the device is commissioned by LNS.

2.6 External Network Variables

For an external network variable there is no (static or dynamic) network variable representation on the device. The configuration software is used to configure the address of a network variable in a remote device and the local device uses network management commands to poll or update the network variable data in the remote device. LNS is not involved in that configuration process. Adding or deleting external network variables does not change the external interface of a device, so changes can be made even if there is no LNS connection of the configuration software. A disadvantage of external network variables is that whenever the address configuration of the remote device is changed, also the information in the local device has to be adjusted using with the configuration software.

2.7 Configuration Software Connection

The L-Vis, L-Gate, L-OPC and L-Proxy devices are configured with configuration software packages. The software can always be started from the Windows Start Menu and a new configuration can be generated and saved to a file. To load a new configuration into the device, the software has to connect to the device. This can be done either by starting the software as an LNS plug-in, starting the software standalone and connecting to an LNS data base or by establishing an IP connection to the device. Figure 2 shows the two connection methods of the software.

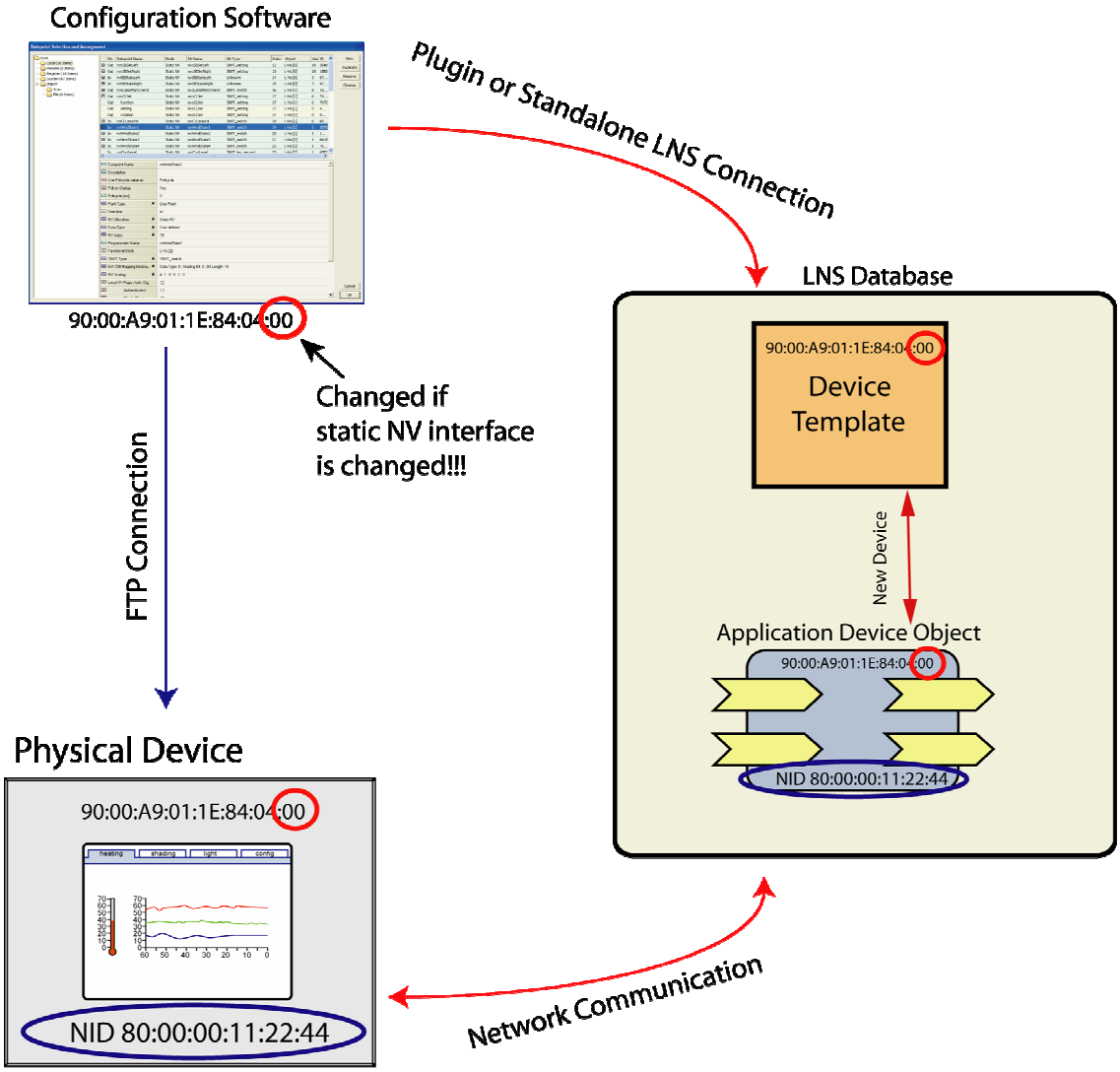

Figure 2: Configuration Software Connection

2.7.1 LNS Plugin Mode (Plugin Connection)

In the LNS Plug-in mode, the software can connect to the device using the LNS API. The software can be started by selecting a device in the network manager project and starting the software as a LNS plug-in on that device. The software "knows" which device in the LNS data base is configured and therefore can also perform LNS operations on that device. Such operations include the creation of dynamic network variables, creation of network variable bindings as well as an upgrade of device templates, which makes it possible to change the static network variable interface of a node even when the node is already installed. Usually the project file is loaded into the device using the LonMark file transfer protocol.

2.7.2 Standalone Mode with LNS Connection

Another possibility is to start the software standalone and select an LNS data base as well as the device inside of the LNS data base which ought to be configured (only L-Gate, L-Proxy

and L-OPC). Also in that case the tool can apply changes to the LNS data base as described above.

2.7.3 IP Mode (FTP Connection)

In IP mode, the software connects to the device under configuration using an IP connection. In that case, the IP address of the device is entered in the configuration software and the configuration tool communicates with the device using the FTP and/or Telnet protocol. The advantage of this connection method is that the communication over the IP protocols is much faster than transferring files over a LonWorks channel. E.g. on the L-Vis devices, the project data not only can be transferred much faster, but the project data is also transmitted in uncompressed format which significantly reduces the boot time of the L-Vis device. However, since there is no connection to the LNS Database, operations like creating network variable bindings can not be performed. In addition, configuration changes on already commissioned devices must not change the static NV interface of the device to avoid inconsistencies between the device on the network and the LNS data base.

3 Loading a new Configuration into a Device

Having described all subjects which are relevant to the external device interface in connection with LNS Databases, it is now possible to understand what happens when a new configuration is loaded into a device. Loading a new configuration into a device which is not commissioned in a LNS Database as well as loading a new configuration where no changes to the network variable interface have been made, is trivial.

The user has to be careful when changing the static network variable interface of a device which is already commissioned in an LNS Database. As explained in the chapters before, the change of the static network variable interface (this is, deleting, adding or renaming static network variables) has the following consequences:

1. The model number in the program ID is changed → the program ID is changed
2. if a XIF file is required, a new XIF file must be generated
3. the device template in the LNS Database must be updated

LNS provides methods to update the device template of an already existing device in the LNS Database. There are configurations where the update of the device template causes a loss of already defined network variable bindings or wrong network variable bindings after the upgrade. In particular, a change of the static network variable interface in a node which also has defined dynamic network variables can not be handled correctly by LNS.

Therefore, the LOYTEC configuration software performs the following steps when the external interface of the device is changed by the new configuration :

1. Save and delete all existing bindings of the node
2. Delete all existing dynamic network variables from the device
3. Load the new configuration software into the device
4. Update the device template of the device so that it reflects the new external interface and commission the device
5. Restore the dynamic network variables on the device
6. Restore all saved binding on the device.

All these operations are automatically performed in the LNS Database when a new configuration is loaded into the device.

If the configuration software is started stand-alone and the configuration is loaded into the device using an IP connection, there is no access to the LNS Database and the upgrade can not be performed. However, because the static NV interface has changed, a new program ID is assigned to the device. Since the LNS Database has already commissioned the device using the old program ID, LNS throws an error message when trying to re-commission the device. Some management tools provide methods to repair devices with a changed program ID by

upgrading the device template. But this method does not provide the smart algorithm of saving and restoring the network variable bindings, so this can lead to wrong or missing network variable bindings. If dynamic network variables are configured on the node, the method will fail in any case.

As a consequence, only change the static network variable interface of a node which is commissioned in an LNS Database when the configuration tool is started in LNS mode (plug-in or LNS connected).

4 References

[1] LONMARK Standard Program ID:
http://www.lonmark.org/technical_resources/resource_files/spid_master_list.asp

[2] LONMARK Device Interface File Reference Guide, Revision 4.401, September 2005

[3] LONMARK Application-Layer Interoperability Guidelines, Version 3.4, September 2005

5 Legal notice