LUIS ePayment Installation

Guide

PAYMENT

Version 5.10, Januar 2016

LUTZ Informationssysteme GmbH
Konrad-Adenauer-Platz 3
56410 Montabaur
Tel.: +49 (2602) 94922 – 0
Fax: +49 (2602) 94922 - 29
email: info@lutz-is.de
http://www.lutz-is.de

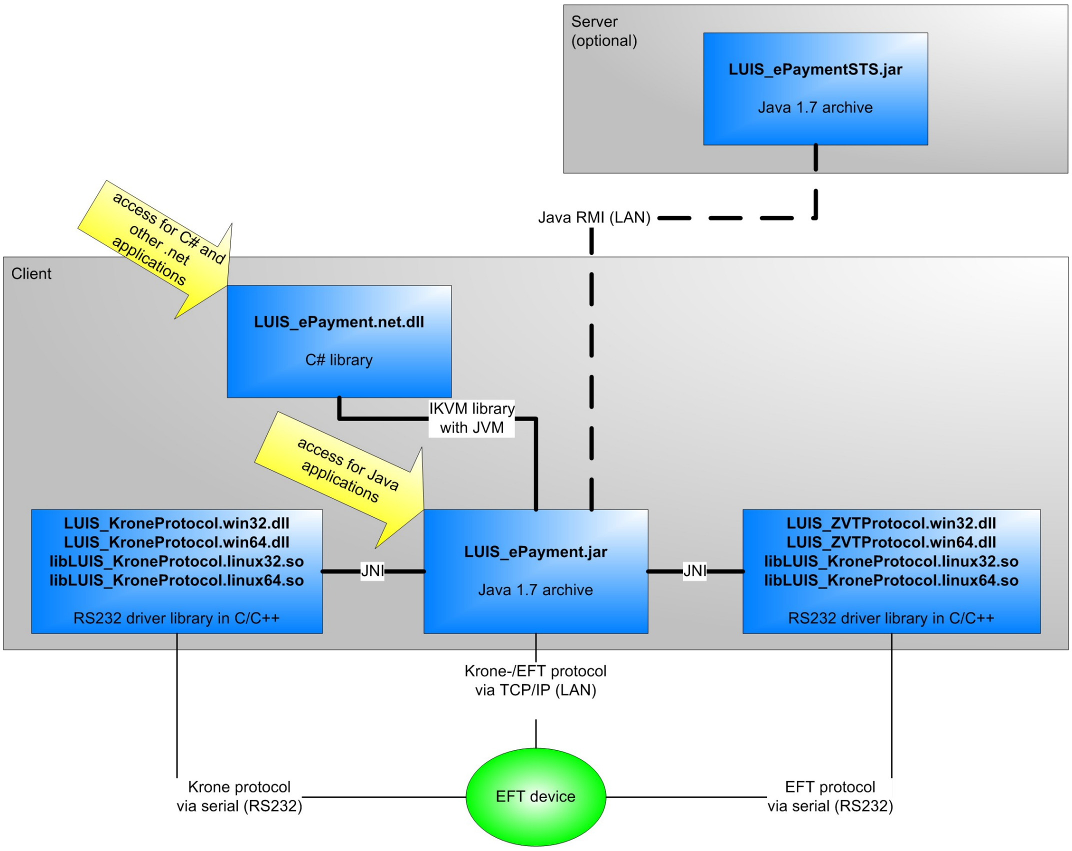
LUTZ Informationssysteme GmbH erteilt keine stillschweigenden Garantien auf handelsübliche Qualitäten und Eignung für einen bestimmten Einsatzzweck. LUTZ Informationssysteme GmbH übernimmt keine Haftung für Fehler oder Folgeschäden, die durch Ausstattung, Leistung und Gebrauch dieser Dokumentation entstehen.

Fast alle Hardware- und Software-Bezeichnungen, die in dieser Dokumentation erwähnt werden, sind gleichzeitig auch eingetragene Warenzeichen oder sollten auch ohne explizite Kennzeichnung als solche betrachtet werden.

Document History

Version	Date	Author	Changes	Comments
1.0	26.09.2007	M. Gelfort		first version
5.0	31.08.2014	M. Gelfort	Support of ZVT and KRONE protocoll.	The version 5.0 is **not backward compatible** with previous versions.
5.1	14.11.2014	M. Gelfort	SharedTerminalServer; 32- and 64-bit support	
5.2	15.12.2014	M. Gelfort		
5.3	05.02.2015	M. Gelfort	Running LUIS ePayment SharedTerminalServer as a Windows service or Linux Daemon. C/C++ libraries for Linux.	
5.6	May 2015	M. Gelfort	Modified installation for SharedTerminalServer as service/daemon. JRE 1.7 mandatory for SharedTerminalServer as service/daemon	
5.7	22.06.2015	M. Gelfort	Minor changes.	
5.9	09.09.2015	M. Gelfort	new Chapter: „Supported devices“; Actualisation of Chapter „Configuration“	
5.10	26.01.2016	M. Gelfort		

1 Architecture


Server
(optional)
LUIS_ePaymentSTS.jar
Java 1.7 archive
Java RMI (LAN)
access for C# and other .net applications
Client
LUIS_ePayment.net.dll
C# library
IKVM library with JVM
access for Java applications
LUIS_KroneProtocol.win32.dll
LUIS_KroneProtocol.win64.dll
libLUIS_KroneProtocol.linux32.so
libLUIS_KroneProtocol.linux64.so
RS232 driver library in C/C++
JNI
LUIS_ePayment.jar
Java 1.7 archive
JNI
LUIS_ZVTProtocol.win32.dll
LUIS_ZVTProtocol.win64.dll
libLUIS_KroneProtocol.linux32.so
libLUIS_KroneProtocol.linux64.so
RS232 driver library in C/C++
Krone-/EFT protocol
via TCP/IP (LAN)
Krone protocol
via serial (RS232)
EFT device
EFT protocol
via serial (RS232)

The server side components are only available, if LUIS ePayment is installed with the shared terminal option. In a simple standalone installation of LUIS ePayment, where one installation has exactly one EFT device assigned and locally connected, only the client side components are needed.

2 Supported EFT devices

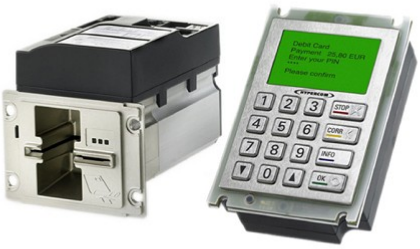


Abbildung 1: Verifone Artema Modular

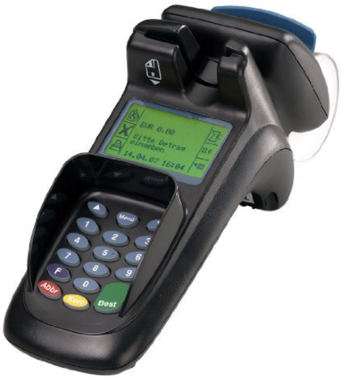


Abbildung 2: Verifone Artema Hybrid

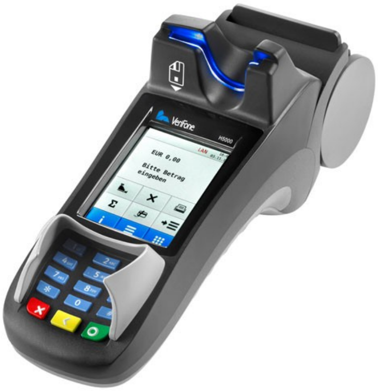


Abbildung 3: Verifone H5000

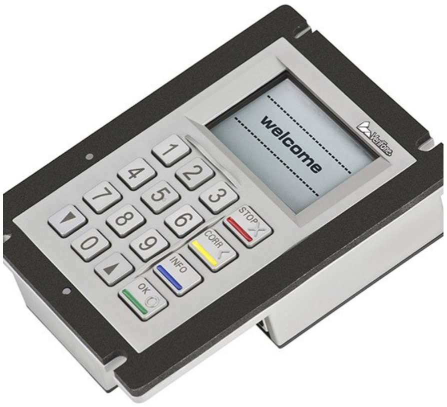


Abbildung 4: Verifone UX100

Abbildung 5: Ingenico iCT250

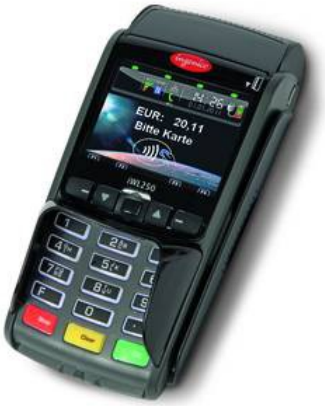


Abbildung 6: Ingenico iWL250

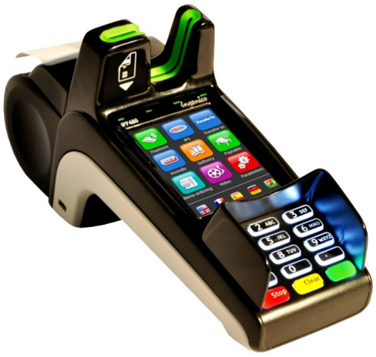


Abbildung 7: Ingenico iPP480

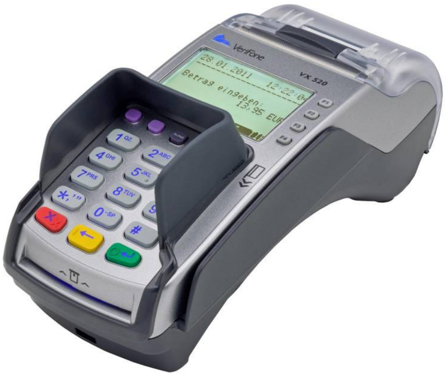


Abbildung 8: CCV VX520

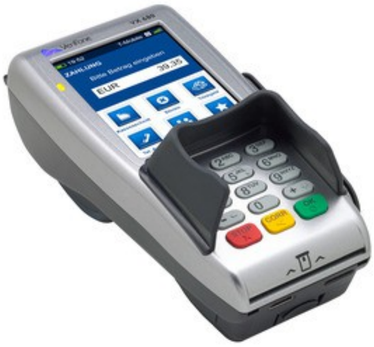


Abbildung 9: CCV VX680

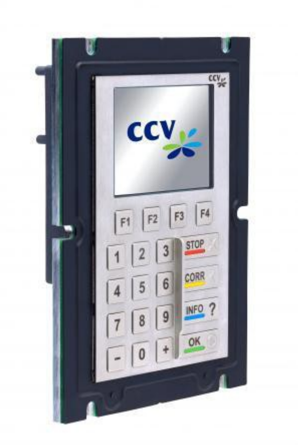


Abbildung 10:

CCV OPP C60

3 LUIS ePayment (standalone)

3.1 Files overview

3.1.1 Java files

LUIS ePayment consists of two parts. The first part contains the complete business logic and all the objects, the end user is handling with. This part is placed in the file

```
lib/LUIS_ePayment.jar
```

The second part is a thin layer, responsible for the communication via the RS232 interface with the electronic fund terminal (EFT). This second part is written in C and C++ to speed up communication and send commands/replies between Java and the EFT without major delay. Currently two different communication protocols are supported:

EFT protocol, a standard for many EFTs

KRONE protocol, a proprietary protocol from Verifone

These two different protocols are implemented for Windows 32Bit in the files

```
lib/LUIS_ZVTProtocol.win32.dll
lib/LUIS_KroneProtocol.win32.dll
```

and for Windows 64Bit in the files

```
lib/LUIS_ZVTProtocol.win64.dll
lib/LUIS_KroneProtocol.win64.dll
```

If running under Linux 32Bit, the corresponding libraries are

```
lib/libLUIS_ZVTProtocol.linux32.so
lib/libLUIS_KroneProtocol.linux32.so
```

and for Linux 64Bit

```
lib/libLUIS_ZVTProtocol.linux64.so
lib/libLUIS_KroneProtocol.linux64.so
```

The following files are needed for logging and communication via ISDN. They can be considered as utility functions for Java.

```
lib/log4j-1.2.16.jar
lib/jcapi.jar
```

Next we have some settings files, which configure the communication with the EFT and define a log level and the behaviour of runtime tracing.

```
etc/zvt.cfg
etc/LUIS_ePayment.xml
```

At last a license file is needed for a valid license of LUIS ePayment.

```
etc/LUIS_ePaymentLicense.dat
```

The name of this file can be changed since it is referenced in zvt.cfg.

3.1.2 .net files

When invoking LUIS ePayment from a .net environment, all files mentioned in chapter 3.1.1 are needed. Additionally the file

```
lib/LUIS_ePayment.net.dll
```

is mandatory. This later file has to be referenced in the .net project. Because it has a strongName, it is usable in a certified environment.

3.2 Installation steps

3.2.1 Java environment

1. A Java runtime (JRE) in Version 1.7 (or higher) has to be installed on the system. The current version can be downloaded at

   http://java.com/

   We always recommend to use the latest version of Java.

2. Create a home directory for the LUIS ePayment files. Its a good idea to use an environment vaiable like

   ```
   $LUIS_EPAY_HOME
   ```

3. Copy all files from chapter 3.1.1 to **$LUIS_EPAY_HOME**.

4. Edit the files

   ```
   etc/zvt.cfg
   etc/LUIS_ePayment.xml
   ```

   and correct the value of all files mentioned there.

5. When running in a Java environment, migrate **$LUIS_EPAY_HOME** in your Java classpath.

3.2.2 .net environment

1. Do all the steps of chapter 3.2.1.

2. The file

   `lib/LUIS_ePayment.net.dll`

   must be in the same directory with

   `lib/LUIS_ePayment.jar`

3. Install IKVM, a tool, to access Java code from .net. It can be downloaded at

   http://ikvm.net

   Instead of downloading, you can access ikvm directly from the **`lib for .net/`** folder.

4. Add a reference to

   `/lib/IKVM.OpenJDK.Core.dll`

   and

   `/lib/LUIS_ePayment.net.dll`

   in your .net project and compile it. The LUIS ePayment Implementation resides in the packages

   `de.luis.kioskComponents.ePayment.*`

4 LUIS ePayment SharedTerminalServer

4.1 Files Overview

4.1.1 Server side files

The LUIS ePayment SharedTerminalServer is written in Java and communicates with standalone LUIS ePayment clients via TCP/IP.

If **$LUIS_EPAY_STS_HOME** is the topmost installation directory, we recomend the following structure:

$LUIS_EPAY_STS_HOME/bin	contains batch script to start the java program
$LUIS_EPAY_STS_HOME/lib	all *.jar, *.dll and *.so files reside here
$LUIS_EPAY_STS_HOME/etc	this is the directory for the configuration files

Lets have a closer look at the individual files.

The main file is

```
lib/LUIS_ePaymentSTS.jar
```

where all business logic is implemented.

This *.jar additionally needs the java archives

```
lib/jcapi.jar
lib/log4j-1.2.16.jar
lib/yajsw-stable-11.11/
```

In order to easily start, stop, install and uninstall the server program, a Java Ant script is available

```
bin/start.bat
```

This Script invokes an Ant script

```
bin/build_win.xml
```

At last the zvt.cfg and license files of the standalone installation are moved on the server side.

We recommend to use a configuration file for each shared EFT device. It is good praxis to create a server side configuration directory and put all configuration files of each EFT in it, together with a license file (not compatible with the standalone version of LUIS ePayment).

A server side configuration directory (eg. **./etc**) for three EFTs with terminal IDs 11111111, 22222222, and 33333333 might look like this:

```
etc/11111111.cfg
etc/11111111.lic
etc/22222222.cfg
etc/22222222.lic
etc/33333333.cfg
etc/33333333.lic
```

The file extension *.cfg is mandatory, because the server will lookup the configuration for client request by the TID appended with ".cfg". The name of the license files can be modified, because they are referred within the *.cfg files.

The server itself needs no license file.

There is an additional file in the configuration directory named

```
etc/LUIS_ePaymentSTS.xml
```

which is a configuration file for server side logging via log4j. If you are unfamiliar with log4j, have a look at http://logging.apache.org/log4j/ .

4.1.2 Client side files

No license file is needed on the client side. The file mentioned in chapter 3.1.1 is obsolete and will be ignored.

Because the configuration files are moved from the client to the server side, we need a simplified client configuration file, which contains the address of the LUIS ePayment SharedTerminalServer. Chapter 7.2.1 decribes this in detail.

4.2 Installation steps

4.2.1 Java

1. A Java runtime (JRE) in Version 1.7 (or higher) has to be installed on the system. The current version can be downloaded at

http://java.com/

We always recommend to use the latest version of Java.

2. Install Java Ant. The current version is available here

   http://ant.apache.org/

3. Create a home directory for the LUIS ePayment files. Its a good idea to use an environment variable like

   ```
   $LUIS_EPAY_STS_HOME
   ```

4. Copy all files from chapter 4.1.1 to **$LUIS_EPAY_STS_HOME**.

5. Edit the files

   ```
   $LUIS_EPAY_STS_HOME/etc/<tid>.cfg
   $LUIS_EPAY_STS_HOME/etc/LUIS_ePaymentSTS.xml
   ```

   and correct the value of all files mentioned there. We urgently recommend to have a correct and well configured log4j environment running, before proceeding. Only logging allows any insight in what's going on in LUIS ePayment SharedTerminalServer and helps finding configuration problems.

   Migrate **$LUIS_EPAY_STS_HOME** in your Java classpath.

   Have a look at the **LicenceFile=<filename>** entry in the **<tid>.cfg** file. The filename is relative to the **cfgPath** parameter of the startup call (see 4.3.1).

6. Start the LUIS ePayment SharedTerminalServer by invoking

   ```
   bin/start.bat
   ```

   or by a manual call of Ant with input file

   ```
   bin/build_win.xml
   ```

4.3 Invocation/Startup

4.3.1 Startup the server

The DOS command script

```
bin/start.bat
```

starts the Apache Ant script

```
bin/build_win.xml
```

which invokes the Java runtime environment and

```
        de.luis.kioskComponents.ePayment.sharedTerminalServer
                                                   .PaySharedTerminalServer
```

as the **Main** method class.

You can use the DOS script or only the Ant script, or you can directly invoke the Java class. All approaches have the following parameters:

de.luis.kioskComponents.ePayment.sharedTerminal.PaySharedTerminalServer

-start [-cfgPath <path>] [-port <portNr>]

to start the server as an application

and specify a directory, where the configuration files reside (default = "**../etc**")

and define a TCP/IP port number for the Java RMI registry different from **1099** (= default).

Never invoke **-start**, when running the server as a service/daemon.

-stop

to stop the server, running as an application.

Never invoke **-stop**, when running the server as a service/daemon.

-configure [-libPath <path>] [-cfgPath <path>] [-port <portNr>]

to create a configuration file for running LUIS ePayment SharedTerminalServer as a Windows service or Linux daemon. This configuration file is the requirement for the later call of **-install**.

Parameter **-libPath** defines the directory, where the *.jar files are located (default = "**../lib**").

You may also specify with **-cfgPath** a directory, where the configuration files reside (default = "**../etc**")

and define with **-port** a TCP/IP port number for the Java RMI registry different from **1099** (= default).

You need administrator rights here.

-install [-libPath <path>] [-cfgPath <path>]

to install and run the server as a Windows service or Linux daemon

and specify the directory, where the *.jar files are located (default = "**../lib**").

You can define the directory of the configuration files (default = "**../etc**") to let log4j find its property file.

You need administrator rights here.

`-uninstall [-libPath <path>]`

to uninstall the server service/daemon

and specify the directory, where the *.jar files are located (default = "**../lib**").

You need administrator rights here.

Samples:

```
bin/start.bat -start -cfgPath ../etc -port 1099
```

starts the SharedTerminalServer as Application

```
bin/start.bat -start
```

stops the Application

```
bin/start.bat -configure -libPath ../lib -cfgPath ../etc -port 1099
```

creates a configuration file, which is the precondition to start as daemon/service

```
bin/start.bat -install -libPath ../lib -cfgPath
```

uses the previously created configuration file and starts as daemon/service

Installing and running as a service/daemon creates a **`LUIS ePayment SharedTerminalServer Service`**, which will be automatically started, every time, the operating system is booting. But take care to have log4j logging facility correctly configured, because a service/daemon can only communicate through its logfiles. We therefore strictly recommend to run the SharedTerminalServer as a service/daemon only after successfull testing of the installation by running it as an application, not as as service/daemon.

Please keep in mind, that you need administrator rights, to configure, install or de-install services/daemons. Example, if you are using Windows7 or above, configure and install the service with cmd.exe, running under admin rights. After installation the service itself starts and runs with standard service rights.

Sometimes the firewall might hinder the start of the Java RMI registry. So if the **`LUIS ePayment SharedTerminalServer`** stops while starting RMI registry (have a look at the logfile), check your firewall settings and allow the access to and from port 1099 for Java applications.

If no license file is found during a client login to the server, have a look at the **`LicenceFile=<filename>`** entry in the **`<tid>.cfg`** file. The filename is relative to the **`cfgPath`** Parameter.

4.3.2 Startup the client

Starting the client is invoking your Java or .net implementation.

5 Documentation

All interface classes and methods of LUIS ePayment are described in a javadoc, which can be found in the archive

```
LUIS ePayment javadoc.zip
```

Start with the PaySession class and have a look at method

```
PaySession.login()
```

In case of a SharedTerminalServer environment start looking at the method

```
PaySharedSession.login()
```

6 Samples

It is a good idea to have a look at the samples enclosed with LUIS ePayment. All samples can be found in the archive

```
LUIS ePayment samples.zip
```

The samples do always the same:

- login
- query the settings
- make some administrative calls
- start a card payment
- logout

LUIS ePayment is shipped with a Java and C# sample.

7 Configuration

All configuration files need read and write access. These files get read only once during the first need within LUIS ePayment. The configuration files get written as well by LUIS ePayment. So don't use these files to save additional information; it will get lost.

7.1 Standalone client installation

A typical configuration file for LUIS ePayment looks like this:

```
#----------------------
# Device configuration
#----------------------
    #-----------------
    # Common settings
    #-----------------
        TerminalID                  = 69112345
        Device                      = iCT250
        AllowAdministration         = false
        Currency                    = 0x0978
        DisplayLanguage             = de
        UseInternalPrinter          = false
        Timeout                     = 45
        TimeoutMat7000              = 65
    #---------------------------
    # Interface ePayment <-> PC
    #---------------------------
        DeviceProtocol              = ZVT
        DeviceConnector             = LAN
        #-------
        # RS232
        #-------
            COMPort                 = COM1
            SerialPortBaudrate      = 9600
            SerialPortDataBits      = 8
            SerialPortParity        = N
            SerialPortStopBits      = 2.0
        #-----
        # USB
        #-----
            USBEndpointRead         = 0
            USBEndpointWrite        = 0
            USBProductID            = 0
            USBVendorID             = 0
        #-----
        # LAN
        #-----
            IPLocal                 = 192.168.1.100
            PortLocal               = 5577
            IPGateway               = 192.168.1.1
            IPSubnetMask            = 255.255.255.0
    #--------------------------------------
    # Interface device <-> networkProvider
    #--------------------------------------
        NetworkConnection           = LAN
        #------
        # ISDN
        #------
            Call1                   = 221534660
            Call2                   = 221534660
            Call3                   = 221534660
            Call4                   = 221534660
            Call5                   = 0
            UserData                = POSM
        #-----
        # LAN
        #-----
            IP1                     = 17.111.131.52
            IP2                     = 217.111.131.52
            PORT1                   = 500
            PORT2                   = 500
    #---------------------------------
    # Interface device <-> cardReader
    #---------------------------------
        CardReaderConnection        = Controller
    #-----------------------------
    # Interface device <-> PINPad
    #-----------------------------
        PinpadConnection            = Controller
    #----------
    # Security
    #----------
        MerchantPIN                 = DZRiyHWFXLT8cD7Yiy...aLmm7cI8gGeBtQ=
        ServicePIN                  = aJoLBWjk8AAYx5iPUa...hRO8TZWVz1V9Gg=
#--------------------
# ePayment settings
#--------------------
    Simulation                      = false
    CheckRemovalProtection          = false
    DisplayMessages                 = true
    #-------
    # Files
    #-------
        LicenseFile                 = ../etc/LUIS_ePaymentLicense.dat
        LogConfig                   = ../etc/LUIS_ePayment.xml
        MerchantJournal             = ./haendlerjournal.log
        MerchantJournalEncryption   = false
        PINBlacklist                = ../etc/blacklist.cfg
    #----------
    # Internal
    #----------
        HighBusyCounter             = 55
        LowBusyCounter              = 3
        ArchivePath                 = ./
        DisplayTimeShowText         = 3
        LastCashCrop                = 1405435924545
        MaxPINTries                 = 3
        MerchantData1               =
        MerchantData2               =
        MerchantData3               =
        MerchantData4               =
        MerchantData5               =
        Card                        = 0x60
        VU_Nr                       =
    #----------------------
    # SharedTerminalServer
    #----------------------
        SharedTerminalIP            =
        SharedTerminalPort          = 1099
```

7.1.1 Common Settings

Key	Description
TerminalID	Mandatory.

	The 8-digit ID of the EFT. Set this field to 00000000 if you dont't know it. It will be replaced with the current EFT's ID. An eft ID is necessary during eft initialisation.
`Device`	Mandatory. Specifies the type of the EFT. Possible values are {Artema Modular, MAT7000, Artema Hybrid, iCT250, iWL250, iPP480, H5000, VX520, VX680, OPP-C60, UX100, ZVT}
`AllowAdministration`	Boolean flag to allow (=true) or deny (=false) the administration functions on the EFT keyboard. Default=true.
`Currency`	Configures the currency of the EFT as a hexadecimal value. Actually only 0x0978 (=EUR) is supported.
`DisplayLanguage`	Defines the language of the display texts on the EFT. Default=de. Samples: en, de, fr, it (see http://ftp.ics.uci.edu/pub/ietf/http/related/iso639.txt)
`UseInternalPrinter`	Boolean flag to activate (=true) or deactivate (=false) the internal printer of the EFT. Default=false.
`Timeout`	The timeout for a network command.
`TimeoutMat7000`	The timeout for the low level communication with the device (RS232 or TCP/IP) . Default = 65.

7.1.2 Interface ePayment <-> PC

Key	Description
`DeviceProtocol`	Mandatory. Defines the kind of protocol to communicate with the EFT. Possible values are {ZVT, KRONE}. Default=KRONE.
`DeviceConnector`	The way, the eft is connected to LUIS ePayment. This is a mandatory field.

	Possible values are {RS232, USB, LAN}. Default=RS232.

7.1.3 RS232

Key	Description
COMPort	The COM port of the EFT. Samples: COM1, /dev/ttyS1.
SerialPortBaudrate	Defines the baudrate. Possible values = {9600, 115200}. Default=9600.
SerialPortDataBits	Defines the amount of data bits. Possible values = {5, 6, 7, 8}. Default=8.
SerialPortParity	Defines the parity. Possible values = {'N'=none, 'O=odd, 'E'=even, 'M'=mark, 'S'=space}. Default='N'.
SerialPortStopBits	Defines the amount of stop bits. Possible values = {1.0, 1.5, 2.0}. Default for KRONE protocol = 1.0. Default for EFT protocol = 2.0.

7.1.4 USB

USB is currently not supported.

7.1.5 LAN

Key	Description
IPLocal	Mandatory. Local IP of the EFT (DHCP is not supported). Value of 'NetworkConnection' must be set to 'LAN'.
PortLocal	Mandatory.

	Local port of the EFT (DHCP is not supported). Typical values: {ZVT protocol: 20007; KRONE protocol: 22000; Ingenico: 5577; CCV: 20007). Default= 20007.
IPGateway	Mandatory. Gateway IP of EFT (DHCP is not supported). Value of 'NetworkConnection' must be set to 'LAN'.
IPSubnetMask	Subnet Mask of EFT (DHCP is not supported). DEFAULT=255.255.255.0. Value of 'NetworkConnection' must be set to 'LAN'.

7.1.6 Interface device <-> networkProvider

Key	Description
NetworkConnection	Mandatory. Definition of the kind of connection to the network provider. Possible values are {LAN, ISDN, ISDNviaSeriell}.

7.1.7 ISDN

Key	Description
Call1	Mandatory. 1.ISDN phone number of the networkprovider. Value of 'NetworkConnection' must be set to 'ISDN' or 'ISDNviaSeriell'.
Call2	Mandatory. 2.Datex-P number of the network provider. Value of 'NetworkConnection' must be set to 'ISDN' or 'ISDNviaSeriell'.
Call3	Mandatory. 2.ISDN phone number of the networkprovider. Value of 'NetworkConnection' must be set to 'ISDN' or 'ISDNviaSeriell'.
Call4	Mandatory. 2.Datex-P number of the network provider.

	Value of 'NetworkConnection' must be set to 'ISDN' or 'ISDNviaSeriell'.
`Call5`	Mandatory. Phone prefix, if EFT is connected to a phone system. Value of 'NetworkConnection' must be set to 'ISDN' or 'ISDNviaSeriell'.
`UserData`	Specifies the Datex-P connection. Value of 'NetworkConnection' must be set to 'ISDN' or 'ISDNviaSeriell'.

7.1.8 LAN

Key	Description
`IP1`	Mandatory for KRONE protocol. 1.IP of the network provider. Value of 'NetworkConnection' must be set to 'LAN'.
`IP2`	Mandatory for KRONE protocol. 2.IP of the network provider. Value of 'NetworkConnection' must be set to 'LAN'.
`PORT1`	Mandatory for KRONE protocol. 1.Port of the network provider. Value of 'NetworkConnection' must be set to 'LAN'.
`PORT2`	Mandatory for KRONE protocol. 2.Port of the network provider. Value of 'NetworkConnection' must be set to 'LAN'.

7.1.9 Interface device <-> cardReader

Key	Description
`CardReaderConnection`	Specifies the way, the card reader is connected to the EFT. Possible values are {Controller, PINPad, PC}

7.1.10 Interface device <-> PINPad

Key	Description

`PinpadConnection`	Specifies the way, the PINpad is connected to the EFT. Possible values are {Controller}

7.1.11 Security

Key	Description
`MerchantPIN`	Mandatory. The merchant password of the EFT.
`ServicePIN`	Mandatory. The service password of the EFT.

7.1.12 ePayment settings

Key	Description
`Simulation`	Set this flag to run LUIS ePayment in simulation mode (without EFT device).
`CheckRemovalProtection`	Boolean flag to activate (=true) or deactivate (=false) the state request on the PINPad after a payment transaction. Default=false.
`DisplayMessages`	Configures the EFT to send display messages (=true) or not (=false) during transactions. These display messages are then available in LUIS ePayment, or not. Default=true.

7.1.13 Files

Key	Description
`LicenseFile`	Mandatory. Defines the full filename of the LUIS ePayment license file. Default = ./LUIS_ePaymentLicense.dat.
`LogConfig`	Full qualified path to a XML file, which contains the log4j configuration für LUIS ePayment.
`MerchantJournal`	Filename of the merchant journal. If the file doesn't exist, it will be created. An empty value leads to no merchant journal.

	Default=<empty>
MerchantJournalEncryption	Boolean flag, if merchant journal is encrypted (=true) or not (=false). Default=false.
PINBlacklist	Internal value. Do not change.

7.2 SharedTerminalServer

7.2.1 Client side

Because the client only has to know the address of the LUIS ePayment SharedTerminalServer, the client side configuration file is very small. It must contain the address of the server and some local path entries for logfiles.

A typical client config file in a SharedTerminalServer environment looks like this:

```
#--------------------
# ePayment settings
#--------------------
    Simulation                      = false
    LogConfig                       = ./LUIS_ePayment.xml
    MerchantJournal                 = C:/journal files/merchantJournal.xml
    MerchantJournalEncryption       = false
    #----------------------
    # sharedTerminalServer
    #----------------------
        SharedTerminalIP            = 192.168.1.100
        SharedTerminalPort          = 1099
```

Key	Description
Simulation	Set this flag to run LUIS ePayment in simulation mode (without EFT device).
LogConfig	Full qualified path (use '/' instead of '\') to a XML file, which contains the log4j configuration für LUIS ePayment (see also remark to paths bellow).
MerchantJournal	Filename of the merchant journal (use '/' instead of '\') . If the file doesn't exist, it will be created. An empty value leads to no merchant journal. Default=<empty>

	(see also remark to paths bellow)
`MerchantJournalEncryption`	Boolean flag, if merchant journal is encrypted (=true) or not (=false). Default=false.
`SharedTerminalServerIP`	Mandatory for sharedTerminalServer. Defines the IP of the machine, where LUIS ePayment SharedTerminalServer is running. If this value is set, LUIS ePayment is running in SharedTerminalServer mode.
`SharedTerminalServerPort`	Mandatory for sharedTerminalServer. Sets the port of the Java RMI registry, where LUIS ePayment SharedTerminalServer is running. Default= 1099.

Remark to paths:

All paths are on the client PC side. Please remark, that a path only allows '/' in Java. If you are using a '\', it must be either escaped '\\' or replaced by '/', in Windows and Linux alike.

A blank within a path is allowed. The path must not be embraced by single or double quotes ('...' or “...“).

7.2.2 Server side

The configuration is EFT specific and results in a separate config file. It is a good idea to name these files corresponding to their terminal ID. Each configuration file has the structure of a standalone config file, described in chapter 7.1.