

# PosPay client service installation and integration Linux



[www.payex.no](http://www.payex.no)

## Documentation

PayEx Solutions AS  
PO.Box 143  
N-0323 Oslo

Switch: +47 22 03 63 00

[support@payex.com](mailto:support@payex.com)

### Document information:

<b>AUTHOR:</b>	
----------------	--

## 1 Changes

### Change log:

Date:	Description:	By:
2012-05-24	Initial version	Helge Dahl
2012-10-03	New installation guide for version 4.2.5	Stian Pettersen
2013-01-07	Version 4.4.x	Helge Dahl
2013-05-10	Version 4.6.x	Helge Dahl

**Table of contents**

**1 CHANGES .....1**

**2 INTRODUCTION .....3**

**3 PREREQUISITES.....3**

**4 INSTALLATION.....3**

**5 UPGRADE .....6**

**6 SILENT INSTALL .....6**

**7 TESTING THE INSTALLATION .....6**

**8 RUNNING THE CLIENT SERVICE.....7**

**9 INTEGRATION WITH JAVA API.....7**

## 2 Introduction

This document describes PosPay client service installation and upgrade. In addition it contains basic "getting started information" for integrating through the Java API.

## 3 Prerequisites

The following files are required:

- myntinstaller-x.x.x-Linux32.jar ( File may have different name depending on version and build )

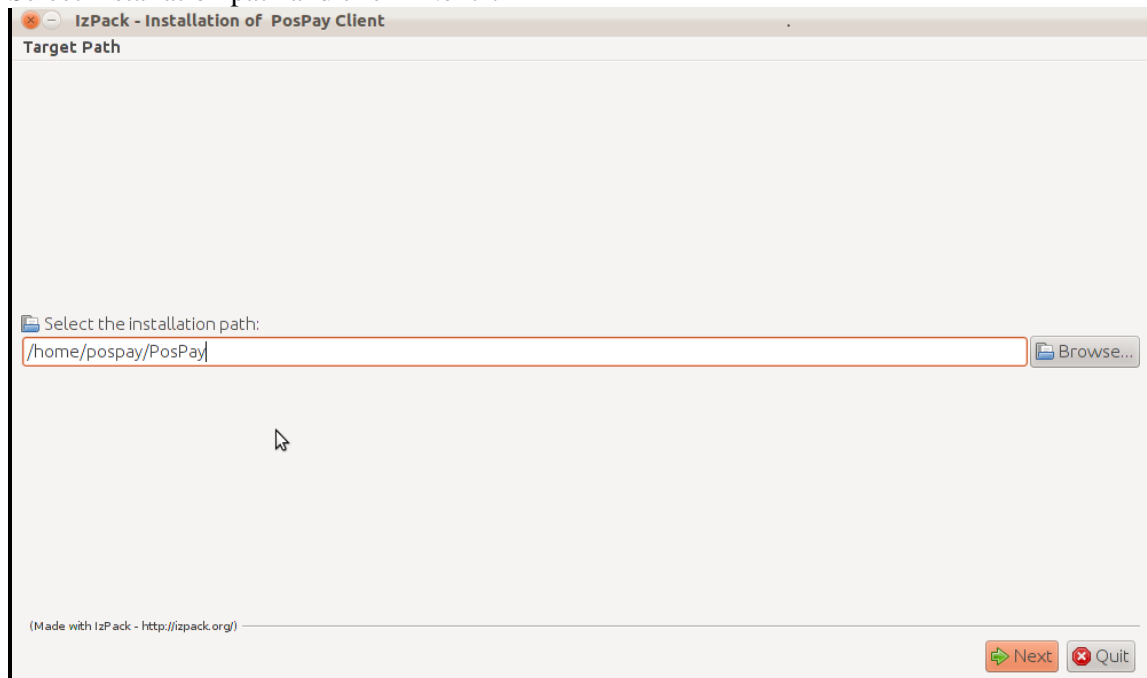
The following files are optional:

- pospay.properties

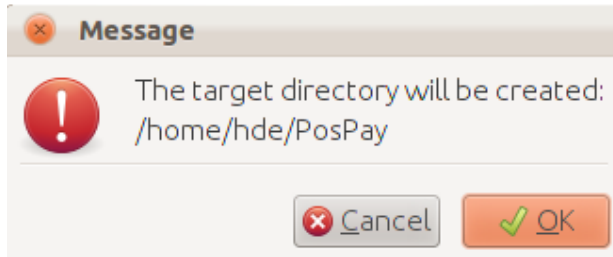
A Java 6 or newer virtual machine must be installed on the host pc to execute the installer.

## 4 Installation

1. Run myntinstaller-x.x.x-Linux32.jar installer using `java -jar` command from preinstalled Java virtual machine.
2. Select installation path and click "Next".



3. Click "OK".

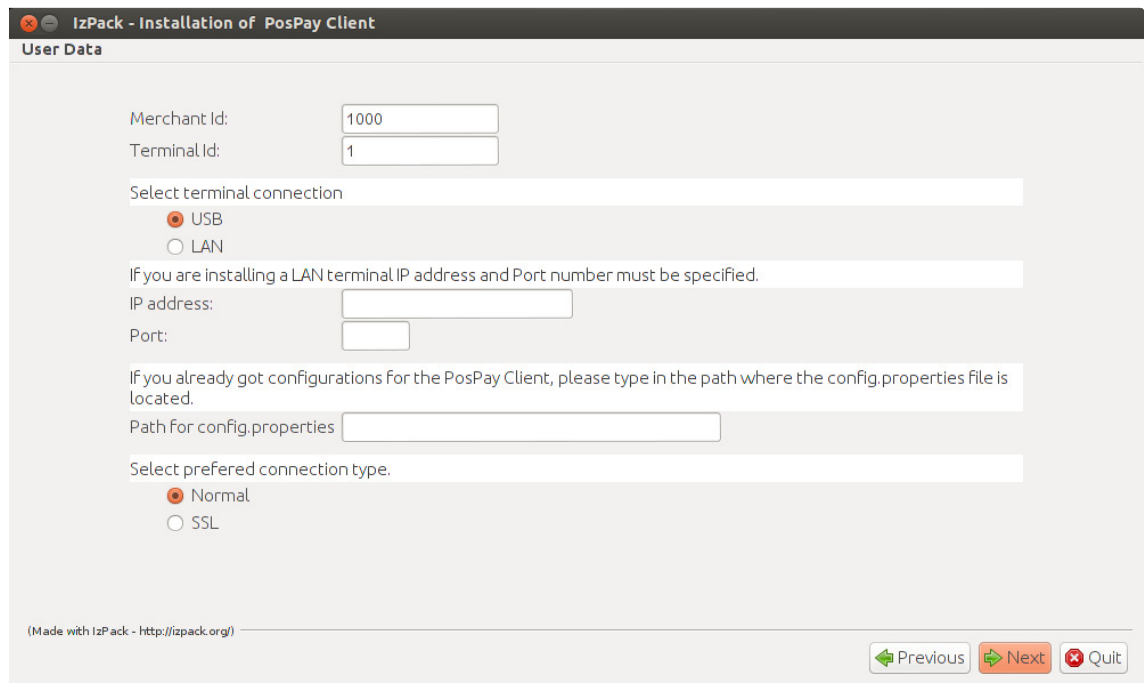


4. Enter merchant id and terminal id.

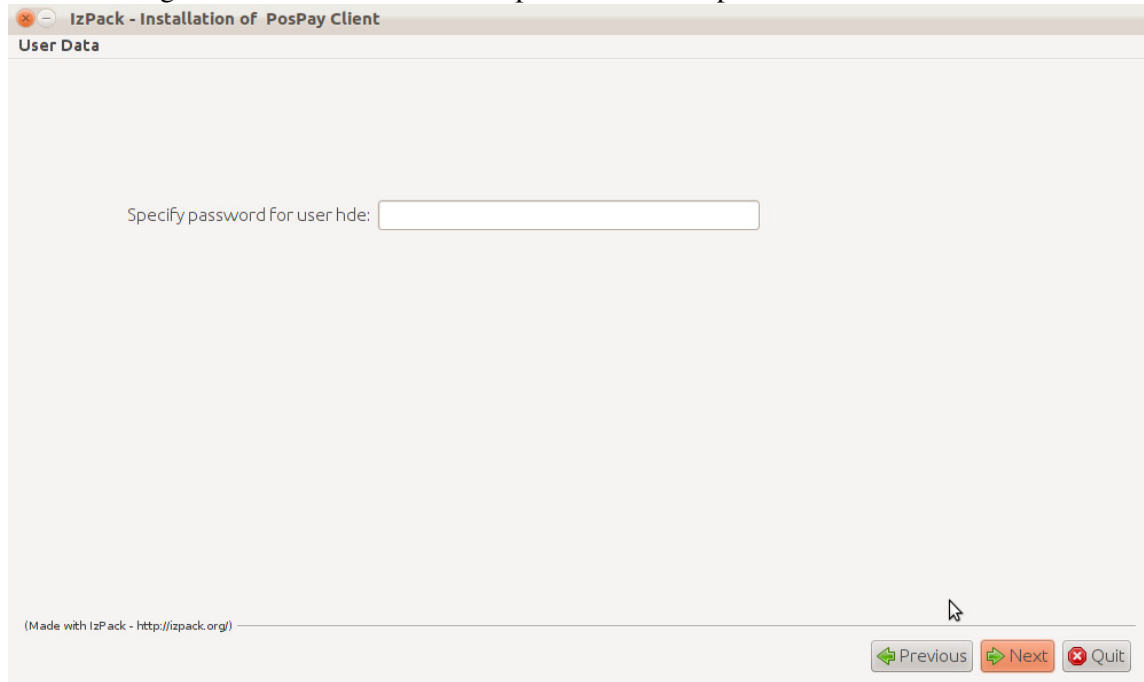
Select terminal connection type USB or LAN. If you are installing for use with a LAN terminal, enter ip address of terminal. Leave port empty ( default ).

If you have a base configuration file for install (pospay.properties), enter fully qualified filename. All properties specified in configuration file will be included in the created config.properties.

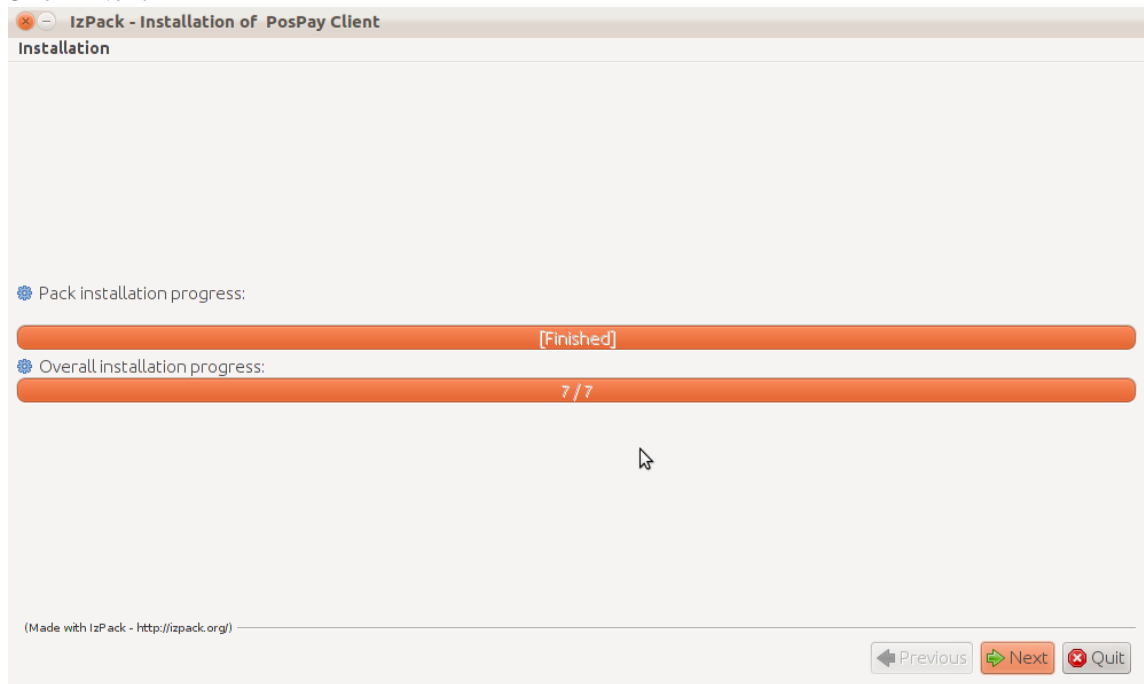
Finally select connection type Normal is for WAN connections to PayEx host, SSL is for open internet connections.



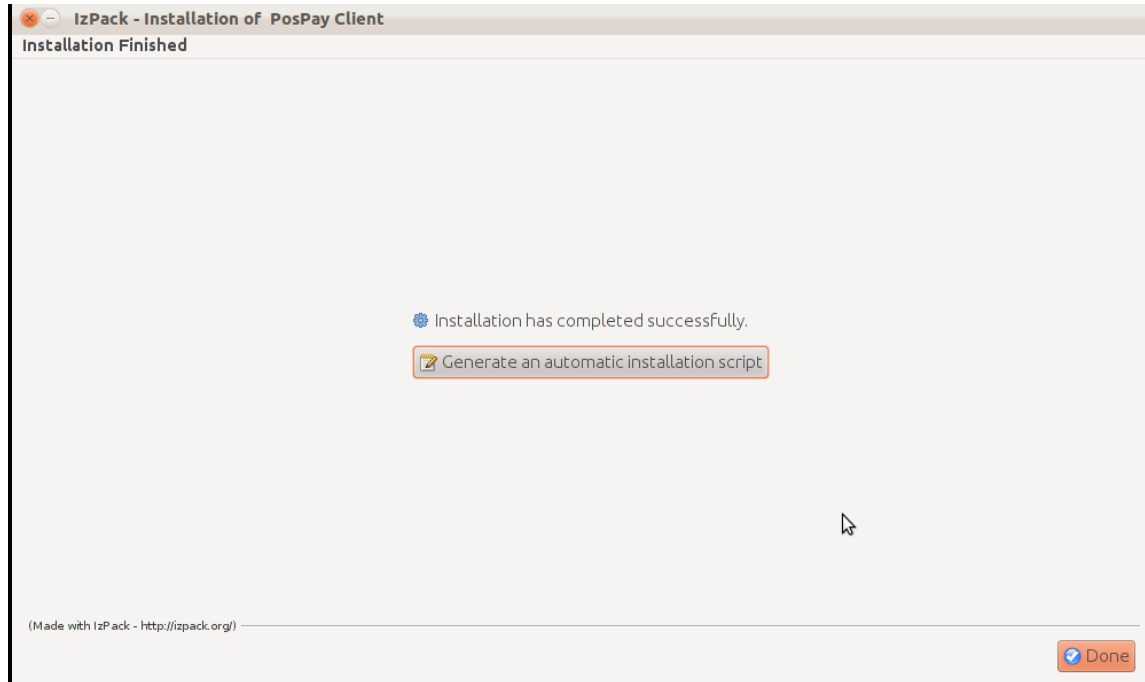
5. The executing shell will now ask for sudo password. Enter password for sudo user.



6. Click "Next"



7. Click "Done".



## 5 Upgrade

Upgrade is performed by the same installer and same execution method as install described in section 5. Select PosPay installation folder as installation path and installer will automatically detect the existing installation and perform an upgrade.

## 6 Silent install

Create an installation properties file, install.properties, containing the following properties:

**INSTALL\_PATH**=<Installation path for the PosPay application>

**merchant-id**=<Merchant id>

**terminal-id**=<Terminal id>

**terminal-connection**=<usb or lan>

**ip-address**=<Ip address of terminal>

**port**=<Terminal port>

**pospay-properties**=<PosPay properties file>

**connection-type**=<normal or ssl>

**javavm**=<Fully qualified path to jvm jre. (Only necessary for installer with no JRE) >

**sudo-password**=<Sudo password>

Run myntinstaller-x.x.x-Linux32.jar installer using the following command from preinstalled Java virtual machine: `java -jar myntinstaller-x.x.x-Linux32.jar -options install.properties`

## 7 Testing the installation

1. Run command `"/etc/init.d/pospay-daemon start"` or to run as a Linux service `"service pospay-daemon start"`.

2. Check logs in <PPP\_HOME>/logs for any errors.
3. Start <PPP\_HOME>/bin/runpcpos.sh. This will start the PCPos terminal test application.
4. Run a purchase transaction and verify that terminal behaves as expected.

## 8 Running the client service

During the installation process a Linux daemon named pospay-daemon is created. This Linux daemon controls the PosPay client service. It supports 4 commands: start, stop, restart and status. Logs are written to <PPP\_HOME>/logs. Optionally the daemon can be executed as a Linux service using the service command.

The API components are able to control startup and shutdown of the PosPay client service. The default behaviour is to let API initialization and deinitialization control the lifecycle of the service. If however this is not the preferred behaviour, this can be disabled by setting the property `pospay.client.api.init.use.pospayservice=false` in <PPP\_HOME>/config/api.properties. The service must then be manually started and stopped.

## 9 Integration with Java API

The Java API consists of the jar file **pospay-client-api.jar**. In addition, several java library files must be included in classpath of the integrating virtual machine. All library files can be found in the sw/java folder of the PosPay installation. Following is a list of jar files on which the Java API depends:

- ezmorph.jar
- commons-beanutils.jar
- commons-codec-1.3.jar
- commons-collections.jar
- commons-lang-2.1.jar
- commons-logging.jar
- log4j.jar
- jdom.jar
- httpclient.jar
- httpcore.jar
- json-lib.jar

When executing the integrating virtual machine, set the following system property:  
`-DPPP_HOME=<POSPAY_INSTALL_HOME>`

Refer to the document `PosPay_Client_API_PA20.pdf` for further integration details.