

EPSON®
EXCEED YOUR VISION



FISCAL PRINTER

UPOS DRIVER USER GUIDE

REVISION HISTORY

Rev	Sheet	Comment	Date	Author
Preliminary		First edition	May, 11th 2016	S.Ravasi
B	9	Modified NumVatRates value	Oct, 26th 2016	S.Ravasi
C	9	Modified NumHeaderLines value	May, 10th 2017	S.Ravasi A.Molinari
D		Improvements in functions and installer process	Feb, 06th 2018	A.Molinari
E		RT printer specific commands	Jun, 06th 2018	S.Ravasi
F		Silent mode installation instructions; Log files management; RT printer peculiarities; Troubleshooting	Sep, 06th 2018	S.Ravasi
G		Added PrintRecTaxID() method	Oct, 10th 2018	S.Ravasi
H		Added Troubleshooting	Feb, 05th 2019	S.Ravasi
I		Modified CapPackageAdjustment value; Added new features of AdditionalHeader property; Added information of GetData() and GetTotalizer() methods; Added PrintRecPackageAdjustment() method; Added PrintRecPackageAdjustVoid(); Added new features of SetVatValue() method; Added events list; Modified the required framework version; Added management up to 10 devices per category; Added "Appendix A – Silent installation and devices manual setting"	Jun, 17th 2019	S.Ravasi
L		Added Linux documentation	Aug, 2nd 2019	A.Molinari
M		Modified NumVatRates value; Modified PredefinedPaymentLines note; Modified SetVatValue() description	Oct, 29th 2019	S.Ravasi
N		Modified GetData(MID_VOID) and GetData(REFUND) notes; Added Linux Troubleshooting "Special characters handling"	May, 05th 2020	S.Ravasi
O		GetTotalizer() method revisited for RT VOID and RETURN handling implemented some usability fixes added support for very high item amounts	Sep, 14 th 2020	A. Molinari

		(only OPOS impact) removed POS for NET single thread issue		
P		Added XML 2.0 features: password management, new modifiers, new payments	Oct, 26 th 2020	A.Molinari
Q		Rearranged payment tables according to XML 2.0, predefined payments flag defaulted to true at installation time	Jan 26 th 2021	A.Molinari
R		GetData(FISCAL_DOC) and GetData(FISCAL_DOC_VOID) are no longer supported; Changed description for GetTotalizer(ITEM), GetTotalizer(ITEM_VOID) and GetTotalizer(REFUND); Added customization for SetVatValue() method	Oct, 05th 2021	S.Ravasi
S		Changed description for GetTotalizer(ITEM), GetTotalizer(ITEM_VOID) and GetTotalizer(REFUND); Added "special" DirectIO(-113)	Jan, 27th 2022	S.Ravasi
T		Added "Fast Print NF" flag in SetupPOS application and "FastPrintNF" property in regsettings.xml file	Feb, 21th 2024	S.Ravasi

Index

1. Overview.....	6
2. Properties.....	7
3. Methods.....	17
4. Events.....	31
5. JAVAPOS driver.....	33
5.1 WINDOWS O.S.....	33
5.2 LINUX O.S.....	38
6. POS for .net driver.....	39
7. OPOS driver.....	44
8. Set and test the devices – SetupPOS application (only for Windows O.S.).....	49
9. Troubleshooting.....	55
9.1 WINDOWS O.S.....	55
9.2 LINUX O.S.....	56
10. Driver error code.....	57
11. Software Rev. History.....	58
Appendix A – Silent installation and devices manual setting.....	62

1. Overview

This document describes the Epson drivers based on UPOS standard.

The drivers available for fiscal printers are:

- JAVA POS driver
- POS for .net driver (only for Windows O.S.)
- OPOS driver (only for Windows O.S.)

All drivers implement the following classes:

- line-display
- cash-drawer
- fiscal-printer
- electronic-journal

Epson fiscal printer driver project is based on a common source code which generates different output: Java-POS driver, POS for .net driver and OPOS driver.

2. Properties

The driver is compliant to the standard UPOS ver. 1.13 for the 4 classes supported with the following capability:

Line Display

	<i>Value</i>	<i>Notes</i>
<i>Common</i>		
CapCompareFirmwareVersion	false	
CapPowerReporting	PR_NONE	
CapStatisticsReporting	false	
CapUpdateFirmware	false	
CapUpdateStatistics	false	
CheckHealthText	"Display driver internal test: successful"	Return a fixed string
Claimed		
DeviceEnabled		
FreezeEvents		If true, events are lost
PowerNotify		Set does not works
PowerState		
State		
DeviceControlDescription		
DeviceControlVersion		
DeviceServiceDescription		
DeviceServiceVersion		
PhysicalDeviceDescription		
PhysicalDeviceName		
<i>Specific</i>		
CapBlink	DISP_CB_NOBLINK	
CapBitmap	false	
CapBlinkRate	false	
CapBrightness	false	
CapCharacterSet	DISP_CCS_ASCII	
CapCursorType	DISP_CCT_NONE	
CapCustomGlyph	false	
CapDescriptors	false	
CapHMarquee	false	
CapICharWait	false	
CapMapCharacterSet	false	
CapReadBack	DISP_CRB_NONE	
CapReverse	DISP_CR_NONE	
CapScreenMode	false	
CapVMarquee	false	

BlinkRate	E_ILLEGAL	
CharacterSet	DISP_CS_ASCII	
CharacterSetList	"998"	
Columns	20	
CurrentWindow		
CursorColumn		
CursorRow		
CursorType	DISP_CT_NONE	
CursorUpdate		
CustomGlyphList		
DeviceBrightness	100	
DeviceColumns	20	
DeviceDescriptors	0	
DeviceRows	2	
DeviceWindows	0	
GlyphHeight	0	
GlyphWidth	0	
InterCharacterWait	0	
MapCharacterSet	false	
MarqueeFormat	DISP_MF_WALK	
MarqueeRepeatWait	0	
MarqueeType	DISP_MT_NONE	
MarqueeUnitWait	0	
MaximumX	0	
MaximumY	0	
Rows	2	
ScreenMode	0	
ScreenModeList	"2x20"	

Cash Drawer

	<i>Value</i>	<i>Notes</i>
<i>Common</i>		
CapCompareFirmwareVersion	false	
CapPowerReporting	PR_STANDARD	
CapStatisticsReporting	false	
CapUpdateFirmware	false	
CapUpdateStatistics	false	
CheckHealthText	"Drawer driver internal test: successful"	Return a fixed string
Claimed		
DeviceEnabled		
FreezeEvents		If true, events are lost
PowerNotify		
PowerState		

State		
DeviceControlDescription		
DeviceControlVersion		
DeviceServiceDescription		
DeviceServiceVersion		
PhysicalDeviceDescription		
PhysicalDeviceName		
<i>Specific</i>		
CapStatus		Setting on the SetupPOS application: true/false
CapStatusMultiDrawerDetect	false	
DrawerOpened		

Fiscal Printer

	<i>Value</i>	<i>Notes</i>
<i>Common</i>		
CapCompareFirmwareVersion	false	
CapPowerReporting	PR_STANDARD	
CapStatisticsReporting	false	
CapUpdateFirmware	false	
CapUpdateStatistics	false	
CheckHealthText	"Printer driver internal test: successful"	Return a fixed string
Claimed		
DeviceEnabled		
FreezeEvents		If true, events are lost
OutputID		
PowerNotify		
PowerState		
State		
DeviceControlDescription		
DeviceControlVersion		
DeviceServiceDescription		
DeviceServiceVersion		
PhysicalDeviceDescription		
PhysicalDeviceName		
<i>Specific</i>		
CapAdditionalHeader	true	
CapAdditionalLines	true	
CapAdditionalTrailer	true	
CapAmountAdjustment	true	
CapAmountNotPaid	true	
CapChangeDue	false	
CapCheckTotal	true	
CapCoverSensor	true	

CapDoubleWidth	true	
CapDuplicateReceipt	false	
CapEmptyReceiptIsVoidable	true	
CapFiscalReceiptStation	true	
CapFiscalReceiptType	true	
CapFixedOutput	false	
CapHasVatTable	true	
CapIndependentHeader	true	
CapItemList	false	
CapJrnEmptySensor	true	
CapJrnNearEndSensor	true	
CapJrnPresent	true	
CapMultiContractor	false	
CapNonFiscalMode	true	
CapOnlyVoidLastItem	true	
CapOrderAdjustmentFirst	false	
CapPackageAdjustment	true	
CapPercentAdjustment	false	
CapPositiveAdjustment	true	
CapPositiveSubtotalAdjustment	true	
CapPostPreLine	true	
CapPowerLossReport	false	
CapPredefinedPaymentLines		Setting on the SetupPOS application: true/false
CapReceiptNotPaid	true	
CapRecEmptySensor	true	
CapRecNearEndSensor	true	
CapRecPresent	true	
CapRemainingFiscalMemory	true	
CapReservedWord	true	
CapSetCurrency	false	
CapSetHeader	true	
CapSetPOSID	true	
CapSetStoreFiscalID	false	
CapSetTrailer	true	
CapSetVatTable	true	
CapSlpEmptySensor	false	
CapSlpFiscalDocument	false	
CapSlpFullSlip	false	
CapSlpNearEndSensor	false	
CapSlpPresent	false	
CapSlpValidation	false	
CapSubAmountAdjustment	true	
CapSubPercentAdjustment	false	
CapSubtotal	true	
CapTotalizerType	true	

CapTrainingMode	true	
CapValidateJournal	false	
CapXReport	true	
ActualCurrency	E_ILLEGAL	
AdditionalHeader		In RT mode, it can be used for printing refund and void, instead of DirectIO(). See note n.2
AdditionalTrailer		
AmountDecimalPlaces	4	
AsyncMode	false	Set does not work
ChangeDue	E_ILLEGAL	
CheckTotal		
ContractorId	FPTR_CID_SINGLE	
CountryCode	FPTR_CC_ITALY	
CoverOpen		
DateType	FPTR_DT_RTC	
DayOpened		
DescriptionLength	32/38	According to the FP model
DuplicateReceipt	E_ILLEGAL	
ErrorLevel	FPTR_EL_NONE	Never changes
ErrorOutID	0	Never changes
ErrorState	0	Never changes
ErrorStation	0	Never changes
ErrorString	""	Never changes
FiscalReceiptStation	FPTR_RS_RECEIPT	Available only FPTR_RS_RECEIPT
FiscalReceiptType		Available only FPTR_RT_SALES and FPTR_T_REFUND
FlagWhenIdle	false	Set does not work
JrnEmpty		
JrnNearEnd		
MessageLength	40/46	According to the FP model
MessageType	FPTR_MT_FREE_TEXT	Available only FPTR_MT_FREE_TEXT
NumHeaderLines	16	
NumTrailerLines	99	
NumVatRates	4/9	According to the FP model
PostLine		
PredefinedPaymentLines	"0,1,2,3,4,5,6,000,..."/ E_ILLEGAL	According to the CapPredefinedPaymentLines. See note n.1
PreLine		
PrinterState		
QuantityDecimalPlaces	3	
QuantityLength	7	

RecEmpty		
RecNearEnd		
RemainingFiscalMemory		
ReservedWord	"TOTALE"	
SlpEmpty		
SlpNearEnd		
SlipSelection	FPTR_SS_FULL_LENGTH	
TotalizerType		Available only FPTR_TT_DAY and FPTR_TT_GRAND
TrainingModeActive		

Electronic Journal

	<i>Value</i>	<i>Notes</i>
<i>Common</i>		
AutoDisable	false	
CapCompareFirmwareVersion	false	
CapPowerReporting	PR_STANDARD	
CapStatisticsReporting	false	
CapUpdateFirmware	false	
CapUpdateStatistics	false	
CheckHealthText	"EJ driver internal test: successful"	Return a fixed string
Claimed		
DataCount	0	
DataEventEnabled	false	
DeviceEnabled		
FreezeEvents		If true, events are lost
OutputID	0	
PowerNotify		
PowerState		
State		
DeviceControlDescription		
DeviceControlVersion		
DeviceServiceDescription		
DeviceServiceVersion		
PhysicalDeviceDescription		
PhysicalDeviceName		
<i>Specific</i>		
AsyncMode	false	Set does not work
CapAddMarker	false	
CapErasableMedium	false	
CapInitializeMedium	true	
CapMediumIsAvailable	true	

CapPrintContent	true	
CapPrintContentFile	false	
CapRetrieveCurrentMarker	false	
CapRetrieveMarker	false	
CapRetrieveMarkerByDateTime	true	
CapRetrieveMarkersDateTime	false	
CapStation	EJ_S_RECEIPT	
CapStorageEnabled	false	
CapSuspendPrintContent	false	
CapSuspendQueryContent	false	
CapWaterMark	false	
FlagWhenIdle	false	Set does not work
MediumFreeSpace		
MediumID		
MediumIsAvailable		
MediumSize		
Station	EJ_S_RECEIPT	Available only EJ_S_RECEIPT
StorageEnabled	true	Set does not work
Suspended	false	
WaterMark	false	Set does not work

Notes

1. PredefinedPaymentLines

If CapPredefinedPaymentLines is True, only predefined payment lines are allowed. Here is the description of the indexes used in the PrintRecTotal method for the description parameter.

Old management (single-digit code, representing both type and subtype)

- code "0" = "CONTANTE" (type 0, subtype 00)
- code "1" = "ASSEGNO" (type 1, subtype 01)
- code "2" = "CARTA di CREDITO" (type 2, subtype 01)
- code "3" = "TICKET" (type 3, subtype 01)
- code "4" = "CREDITO" (type 2, subtype 02)
- code "5" = "ALTRO PAGAMENTO" (type 2, subtype 03)
- code "6" = "BANCOMAT" (type 2, subtype 04)

New management (three-digit code, the first represents the type and the other two represent the subtype)

Payment fiscal categories: C = contante – E = elettronico – T = ticket – NR = non riscosso

- code "000" = "CONTANTE" (type 0, subtype 00, C)
- code "001" = "CONTANTE 1" (type 0, subtype 01, C)
- code "002" = "CONTANTE 2" (type 0, subtype 02, C)
- code "003" = "CONTANTE 3" (type 0, subtype 03, C)
- code "004" = "CONTANTE 4" (type 0, subtype 04, C)
- code "005" = "CONTANTE 5" (type 0, subtype 05, C)
- code "100" = "ASSEGNO" (type 1, subtype 00, C)
- code "200" = "CREDITO" (type 2, subtype 00, NR)
- code "201" = "CARTA di CREDITO" (type 2, subtype 01, E)
- code "202" = "CREDITO" (type 2, subtype 02, E)
- code "203" = "ALTRO PAGAMENTO" (type 2, subtype 03, E)
- code "204" = "BANCOMAT" (type 2, subtype 04, E)
- code "205" = "" (type 2, subtype 05, E)
- code "206" = "" (type 2, subtype 06, E)
- code "207" = "" (type 2, subtype 07, E)
- code "208" = "" (type 2, subtype 08, E)
- code "209" = "" (type 2, subtype 09, E)
- code "210" = "" (type 2, subtype 10, E)
- code "301" = "TICKET" (type 3, subtype 01, T)
- code "302" = "" (type 3, subtype 02, T)
- code "303" = "" (type 3, subtype 03, T)
- code "304" = "" (type 3, subtype 04, T)
- code "305" = "" (type 3, subtype 05, T)
- code "306" = "" (type 3, subtype 06, T)
- code "307" = "" (type 3, subtype 07, T)
- code "308" = "" (type 3, subtype 08, T)
- code "309" = "" (type 3, subtype 09, T)

- code "310" = "" (type 3, subtype 10, T)
- code "401" = "TICKET 01" (type 4, quantity 01, T)
- code "402" = "TICKET 02" (type 4, quantity 02, T)
- code "403" = "TICKET 03" (type 4, quantity 03, T)
- code "404" = "TICKET 04" (type 4, quantity 04, T)
- code "405" = "TICKET 05" (type 4, quantity 05, T)
- code "406" = "TICKET 06" (type 4, quantity 06, T)
- code "407" = "TICKET 07" (type 4, quantity 07, T)
- code "408" = "TICKET 08" (type 4, quantity 08, T)
- code "409" = "TICKET 09" (type 4, quantity 09, T)
- code "410" = "TICKET 10" (type 4, quantity 10, T)
- code "500" = "NON RISCOSSO" (type 5, subtype 01, NR)
- code "501" = "NON RISCOSSO BENI" (type 5, subtype 01, NR)
- code "502" = "NON RISCOSSO SERVIZI" (type 5, subtype 02, NR)
- code "503" = "NON RISCOSSO SEGUE FATTURA" (type 5, subtype 03, NR)
- code "504" = "" (type 5, subtype 04, NR)
- code "505" = "NON RISCOSSO SSN" (type 5, subtype 05, NR)
- code "600" = "SCONTO A PAGARE" (type 6, subtype 00, NR)
- code "601" = "BUONO MULTIUSO" (type 6, subtype 01, NR)

The payment description, printed on the receipt after "DETTAGLIO FORME DI PAGAMENTO", can be different than that assigned by the code: if added after the code, this description will be used.

Examples:

PrintRecTotal(10000, 10000, "1")	= "ASSEGNO	1,00"
PrintRecTotal(10000, 10000, "1Assegno xyz")	= "Assegno xyz	1,00"
PrintRecTotal(10000, 10000, "201")	= "CARTA di CREDITO	1,00"
PrintRecTotal(10000, 10000, "201C. di credito")	= "C. di credito	1,00"

New payment types have been added according to XML 2.0 fiscal rules, please refer to table in chapter 3, note 6.

2. AdditionalHeader

In case of RT printer, the AdditionalHeader property can be used for printing refund and void, instead of DirectIO() method.

The property must be set with the string required for open refund or void receipts (1 078 command) and then the contents of the receipt to be returned or canceled must be sent.

Description wording:

- "REFUND zzzz nnnn ddmmyyyy XXMY123456"
- "VOID zzzz nnnn ddmmyyyy XXMY123456"

zzzz => Zreport number of receipt which was printed the return product or receipt to void

nnnn => number of receipt which was printed the return product or receipt to void

dd => day of the receipt to be return or void

mm => month of the receipt to be return or void

yyyy => year of the receipt to be return or void

XXMY123456 => fiscal printer S/N that issued the original receipt

A refund example, in JavaPos:

```
fiscalprinter.setAdditionalHeader("REFUND 0170 0014 11032019 99MEA000213");  
fiscalprinter.beginFiscalReceipt(true);  
fiscalprinter.printRecRefund("T-Shirt REFUND", 200000, 3);  
fiscalprinter.printRecTotal(200000, 200000, "0Contante");  
fiscalprinter.endFiscalReceipt(false);
```

A void example, in JavaPos:

```
fiscalprinter.setAdditionalHeader("VOID 0170 0021 11032019 99IEY000213");  
fiscalprinter.beginFiscalReceipt(true);  
fiscalprinter.printRecRefund("T-Shirt VOID", 200000, 3);  
fiscalprinter.printRecTotal(200000, 200000, "0Contante");  
fiscalprinter.endFiscalReceipt(false);
```

3. Methods

The driver is compliant to the standard UPOS ver. 1.13 for the 4 classes supported with the following methods:

Line Display

	<i>Value</i>	<i>Notes</i>
<i>Common</i>		
Open		
Close		
Claim		
Release		
CheckHealth		No control is done on the devices
DirectIO	E_ILLEGAL	
CompareFirmwareVersion	E_ILLEGAL	
ResetStatistics	E_ILLEGAL	
RetrieveStatistics	E_ILLEGAL	
UpdateFirmware	E_ILLEGAL	
UpdateStatistics	E_ILLEGAL	
<i>Specific</i>		
ClearText		
DisplayText		
DisplayTextAt		
ScrollText	E_ILLEGAL	
ClearDescriptors	E_ILLEGAL	
SetDescriptor	E_ILLEGAL	
CreateWindow	E_ILLEGAL	
DestroyWindow	E_ILLEGAL	
RefreshWindow	E_ILLEGAL	
DefineGlyph	E_ILLEGAL	
ReadCharacterAtCursor	E_ILLEGAL	
DisplayBitmap	E_ILLEGAL	
SetBitmap	E_ILLEGAL	

Cash Drawer

	<i>Value</i>	<i>Notes</i>
<i>Common</i>		
Open		
Close		
Claim		
Release		
CheckHealth		No control is done on the devices
DirectIO	E_ILLEGAL	

CompareFirmwareVersion	E_ILLEGAL	
ResetStatistics	E_ILLEGAL	
RetrieveStatistics	E_ILLEGAL	
UpdateFirmware	E_ILLEGAL	
UpdateStatistics	E_ILLEGAL	
<i>Specific</i>		
OpenDrawer		
WaitForDrawerClose		The <i>beepDelay</i> parameter is not used

Fiscal Printer

	<i>Value</i>	<i>Notes</i>
<i>Common</i>		
Open		
Close		
Claim		
Release		
CheckHealth		No control is done on the devices
ClearOutput		Takes no action
DirectIO		For utilization, see note n.1
CompareFirmwareVersion	E_ILLEGAL	
ResetStatistics	E_ILLEGAL	
RetrieveStatistics	E_ILLEGAL	
UpdateFirmware	E_ILLEGAL	
UpdateStatistics	E_ILLEGAL	
<i>Specific</i>		
BeginFiscalDocument		Initiates invoice to the receipt station
BeginFiscalReceipt		The <i>printHeader</i> parameter is not used
BeginFixedOutput	E_ILLEGAL	
BeginInsertion	E_ILLEGAL	
BeginItemList	E_ILLEGAL	
BeginNonFiscal		
BeginRemoval	E_ILLEGAL	
BeginTraining		
ClearError		Takes no action
EndFiscalDocument		Terminates invoice to the receipt station
EndFiscalReceipt		The <i>printHeader</i> parameter is not used
EndFixedOutput	E_ILLEGAL	
EndInsertion	E_ILLEGAL	
EndItemList	E_ILLEGAL	
EndNonFiscal		
EndRemoval	E_ILLEGAL	
EndTraining		
GetData		Available: FIRMWARE, PRINTER_ID, DAILY_TOTAL, GRAND_TOTAL, MID_VOID, RECEIPT_NUMBER, REFUND,

		FISCAL_REC, FISCAL_REC_VOID, NONFISCAL_REC, DESCRIPTION_LENGTH, Z_REPORT, CURRENT_TOTAL, NOT_PAID For more information, see note n.2
GetDate		
GetTotalizer		Available: ITEM, DISCOUNT, REFUND, ITEM_VOID, GROSS, NET For more information, see note n.3
GetVatEntry		The <i>vatID</i> parameter accepts values from 1 to NumVatRates
PrintDuplicateReceipt	E_ILLEGAL	
PrintFiscalDocumentLine		Prints an invoice line to the receipt station
PrintFixedOutput	E_ILLEGAL	
PrintNormal		The <i>station</i> parameter must be set to FPTR_S_RECEIPT
PrintPeriodicTotalsReport		
PrintPowerLossReport	E_ILLEGAL	
PrintRecCash	E_ILLEGAL	
PrintRecItem		The <i>price</i> parameter is not used if the unit price is different from 0 (the amount is computed from the fiscal printer multiplying the unit price and the quantity). The <i>vatInfo</i> parameter represents the department number. The <i>unitName</i> parameter is not used. Set on the SetupPOS application to print the quantity line, even if it's 1
PrintRecItemAdjustment		The <i>vatInfo</i> parameter represents the department number.
PrintRecItemAdjustmentVoid		The <i>vatInfo</i> parameter represents the department number.
PrintRecItemFuel	E_ILLEGAL	
PrintRecItemFuelVoid	E_ILLEGAL	
PrintRecItemRefund		The <i>price</i> parameter is not used if the unit price is different from 0 (the amount is computed from the fiscal printer multiplying the unit price and the quantity). The <i>vatInfo</i> parameter represents the department number. The <i>unitName</i> parameter is not used
PrintRecItemRefundVoid		The <i>price</i> parameter is not used if the unit price is different from 0 (the amount is computed from the fiscal printer multiplying the unit price and the quantity). The <i>vatInfo</i> parameter represents the department number. The <i>unitName</i> parameter is not used
PrintRecItemVoid		The <i>price</i> parameter is not used if the unit price is different from 0 (the amount is computed from the fiscal printer

		multiplying the unit price and the quantity). The <i>vatInfo</i> parameter represents the department number. The <i>unitName</i> parameter is not used
PrintRecMessage		
PrintRecNotPaid		
PrintRecPackageAdjustment		The <i>vatAdjustment</i> parameter contains a single adjustment. See note n.4
PrintRecPackageAdjustVoid		The <i>vatAdjustment</i> parameter contains a single adjustment. See note n.4
PrintRecRefund		The <i>vatInfo</i> parameter represents the department number.
PrintRecRefundVoid		The <i>vatInfo</i> parameter represents the department number.
PrintRecSubtotal		The <i>amount</i> parameter is not used (the amount is computed from the printer) if the option check-total is not enabled
PrintRecSubtotalAdjustment		
PrintRecSubtotalAdjustVoid		
PrintRecTaxID		This method is NOT supported when the Fiscal Printer is in the Fiscal Receipt Total state and in the Fiscal Receipt Ending state. The <i>taxId</i> parameter must be 16 digits, without spaces, otherwise the EFPTR_TECHNICAL_ASSISTANCE exception is returned. The printer checks the code accuracy.
PrintRecTotal		The <i>total</i> parameter is not used (the amount is computed from the printer) if the option check-total is not enabled
PrintRecVoid		The <i>description</i> parameter is not used
PrintRecVoidItem		The <i>adjustmentType</i> and <i>adjustment</i> parameters are not used
PrintReport		
PrintXReport		
PrintZReport		
ResetPrinter		
SetCurrency	E_ILLEGAL	
SetDate		
SetHeaderLine		
SetPOSID		The <i>cashierID</i> parameter is not used. The <i>POSID</i> parameter can be input up to 3 chars, from 000 to 999, and can also be set by the SetupPOS application
SetStoreFiscalID	E_ILLEGAL	
SetTrailerLine		
SetVatTable		
SetVatValue		The <i>vatValue</i> parameter can be input up to 4 chars, from 0000 to 9999 (the last two

		digits represent decimal part; i.e. "750" = 7,50%, "1000" = 10,00%). The <i>vatID</i> parameter accepts values from 1 to NumVatRates and from 21 to 59 (excluding 30, 40, 50). To specify new department attributes, see note n.7. If the vatValue parameter is set with "n1", "n2", ... or "n11" (VAT exempt codes), the vatID parameter represents the department ID to associate.
VerifyItem	E_ILLEGAL	

Electronic Journal

	<i>Value</i>	<i>Notes</i>
<i>Common</i>		
Open		
Close		
Claim		
Release		
CheckHealth		No control is done on the devices
ClearInput		Takes no action
ClearOutput		Takes no action
DirectIO	E_ILLEGAL	
CompareFirmwareVersion	E_ILLEGAL	
ResetStatistics	E_ILLEGAL	
RetrieveStatistics	E_ILLEGAL	
UpdateFirmware	E_ILLEGAL	
UpdateStatistics	E_ILLEGAL	
<i>Specific</i>		
AddMarker	E_ILLEGAL	
CancelPrintContent	E_ILLEGAL	
CancelQueryContent	E_ILLEGAL	
EraseMedium	E_ILLEGAL	
InitializeMedium		The <i>mediumID</i> parameter is not used
PrintContent		
PrintContentFile	E_ILLEGAL	
QueryContent		
ResumePrintContent	E_ILLEGAL	
ResumeQueryContent	E_ILLEGAL	
RetrieveCurrentMarker	E_ILLEGAL	
RetrieveMarker	E_ILLEGAL	
RetrieveMarkerByDateTime		
RetrieveMarkersDateTime	E_ILLEGAL	
SuspendPrintContent	E_ILLEGAL	
SuspendQueryContent	E_ILLEGAL	

Notes

1. DirectIO

The method *directIO* gives to the application software the possibility to use the native commands of the fiscal printer.

Using the *directIO* method can generate problem to the UPOS driver, the status of the printer can be changed without the control of the driver.

To use the *directIO* method please refer to the Fiscal Printer communication protocol.

parameter	input	output
command	0	-
data	4 figures: first figure represents the H1 parameter while the other remaining 3 figures represent the parameter H2 (see the PDU protocol)	4 figures: first figure represents the H1 parameter while the other remaining 3 figures represent the parameter H2 (see the PDU protocol). The value 0 represents an ERROR.
obj	String to send to the printer	String with the reply from the printer, if data = 0 the content of the string gives the ERROR CODE.

Note:

DirectIO() command has a default timeout of 5000 milliseconds; it is possible to force a custom timeout up to 60000 milliseconds by sending a "special" DirectIO() command immediately BEFORE the one that needs the extended timeout.

The parameters are listed below, after execution of the intended command the timeout will go back to the default value.

parameter	input	output
command	-112	-
data	Timeout <milliseconds>	-
obj	<ignored>	-

2. GetData

Description of "dataItem" parameter:

FPTR_GD_CURRENT_TOTAL

Value of the current receipt. Increased or decreased in real time. Cleared after closing the commercial document.

(*) Can be retrieved only when a fiscal receipt is in progress, otherwise E_ILLEGAL exception is returned.

FPTR_GD_DAILY_TOTAL

Daily total. Increased after closing the fiscal receipt / commercial document. Cleared after printing the daily closure.

FPTR_GD_RECEIPT_NUMBER

Number of issued commercial documents. Indicates the periodic totalizer which is incremented only during the printing of a Z-01 daily fiscal closure (printZReport) and is reset by a Z-07 print.

FPTR_GD_REFUND

Total refund documents amount of the day. Increased after the refund document has been closed. Cleared after printing the daily closure.

FPTR_GD_NOT_PAID

Only in the case of partial payments. Increased or decreased in real time. Cleared after closing the fiscal receipt.

(*) Can be retrieved only when a fiscal receipt is in progress, otherwise E_ILLEGAL exception is returned.

FPTR_GD_MID_VOID

Total void documents amount of the day. Increased after the void document has been closed. Cleared after printing the daily closure.

FPTR_GD_Z_REPORT

Increased after printing the daily fiscal closure. It cannot be reset.

FPTR_GD_GRAND_TOTAL

Increased after printing the daily fiscal closure. It cannot be reset.

FPTR_GD_PRINTER_ID

Printer serial number (model+manufacturer+serial_number, eg EX99123456).

FPTR_GD_FIRMWARE

Fiscal printer firmware version (eg 10.01)

FPTR_GD_RESTART

Not supported.

FPTR_GD_REFUND_VOID

Not supported.

FPTR_GD_NUMB_CONFIG_BLOCK

Not supported.

FPTR_GD_NUMB_CURRENCY_BLOCK

Not supported.

FPTR_GD_NUMB_HDR_BLOCK

Not supported.

FPTR_GD_NUMB_RESET_BLOCK

Not supported.

FPTR_GD_NUMB_VAT_BLOCK

Not supported.

FPTR_GD_FISCAL_DOC

Not supported.

FPTR_GD_FISCAL_DOC_VOID

Not supported.

FPTR_GD_FISCAL_REC

Indicates the number of commercial documents of the day. Increased after the fiscal receipt has been closed. Cleared after printing the daily closure.

FPTR_GD_FISCAL_REC_VOID

Indicates the number of canceled commercial documents of the day. Increased after void the document. Cleared after printing the daily closure.

FPTR_GD_NONFISCAL_DOC

Not supported.

FPTR_GD_NONFISCAL_DOC_VOID

Not supported.

FPTR_GD_NONFISCAL_REC

Number of management documents issued on the day. Increased after the fiscal receipt closure. Cleared after printing the daily closure.

FPTR_GD_SIMP_INVOICE

Not supported.

FPTR_GD_TENDER

Not supported.

FPTR_GD_LINECOUNT

Not supported.

FPTR_GD_DESCRIPTION_LENGTH

32 characters on old printer models and 38 on new printer models (FP-81 II and FP-90 III including RT).

3. GetTotalizer

The TotalizerType property must be set before calling getTotalizer(). There are four possible values of which two are supported:

- FPTR_TT_DAY

Daily type – Supported.

Totalizers increased during the day and reset after the daily fiscal closure.

- FPTR_TT_DOCUMENT

Document type – Not supported.

- FPTR_TT_GRAND

Grand type – Supported.

Periodic totalizers. Increased only after the daily fiscal closure. Cleared by a Z-07 report.

- FPTR_TT_RECEIPT

Receipt type – Not supported.

The following totalizers reside in getTotalizer:

- FPTR_GT_GROSS

For each programmable VAT group (00 to 18), the sales documents total including VAT. The VAT group is the incoming topic.

- FPTR_GT_NET

For each programmable VAT group (00 to 18), the sales documents total excluding VAT.

- FPTR_GT_DISCOUNT

The total discount for all departments.

- FPTR_GT_DISCOUNT_VOID

Not supported.

- FPTR_GT_ITEM

For each programmable department (01 to 99), the sales documents total including VAT. The department number is the incoming topic.

It is possible to obtain this totalizer also for VAT index (00 to 59); the type of selection can be specified using the DirectIO(-113) command (see note below).

- FPTR_GT_ITEM_VOID

For each programmable department (01 to 99), the void documents total including VAT. The department number is the incoming topic.

It is possible to obtain this totalizer also for VAT index (00 to 59); the type of selection can be specified using the DirectIO(-113) command (see note below).

- FPTR_GT_NOT_PAID

Not supported.

- FPTR_GT_REFUND

For each programmable department (01 to 99), the refund documents total including VAT. The department number is the incoming topic.

It is possible to obtain this totalizer also for VAT index (00 to 59); the type of selection can be specified using the DirectIO(-113) command (see note below).

- FPTR_GT_REFUND_VOID

Not supported.

- FPTR_GT_SUBTOTAL_DISCOUNT

Not supported.

- FPTR_GT_SUBTOTAL_DISCOUNT_VOID

Not supported.

- FPTR_GT_SUBTOTAL_SURCHARGES

Not supported.

- FPTR_GT_SUBTOTAL_SURCHARGES_VOID

Not supported.

- FPTR_GT_SURCHARGE

Not supported.

- FPTR_GT_SURCHARGE_VOID

Not supported.

- FPTR_GT_VAT

Not supported.

- FPTR_GT_VAT_CATEGORY

Not supported.

Note:

ONLY for FPTR_GT_ITEM, FPTR_GT_ITEM_VOID and FPTR_GT_REFUND totalizers it is possible to choose the type of selection: by department or by VAT.

By sending a "special" DirectIO() command before the GetTotalizer(), you can select the type.

The default selection is by department.

The choice remains active until the DirectIO() is sent again, or when the driver is closed.

The parameters are listed below.

parameter	input	output
command	-113	-
data	0=department 1=vat	-
obj	<ignored>	-

4. PrintRecPackageAdjustment - PrintRecPackageAdjustVoid

The PrintRecPackageAdjustment() and the PrintRecPackageAdjustVoid() methods can be used to apply / cancel discounts or surcharges on a specific department.

On the other hand, the PrintRecItemAdjustment() and the PrintRecItemAdjustmentVoid() methods apply / cancel discounts or surcharges on the last item sold (and, consequently, to the connected department).

The *vatAdjustment* parameter contains a single adjustment.

PrintRecPackageAdjustment() and PrintRecPackageAdjustVoid() example, in JavaPos:

```
fiscalprinter.beginFiscalReceipt(true);  
fiscalprinter.printRecItem("Item dep.1", 200000, 1000, 1, 200000, "");  
fiscalprinter.printRecItemAdjustment(1, "Item dep.1 Discount", 5000, 1);  
fiscalprinter.printRecItemAdjustmentVoid(1, "Void Item dep.1 Disc.", 5000, 1);  
fiscalprinter.printRecItem("Item dep.2", 80000, 1000, 2, 80000, "");  
fiscalprinter.printRecItem("Item dep.3", 20000, 1000, 3, 20000, "");  
fiscalprinter.printRecSubtotal(300000);  
fiscalprinter.printRecPackageAdjustment(2, "Dep.1 Surcharge", "1,5000");  
fiscalprinter.printRecPackageAdjustment(1, "Dep.2 Discount", "2,10000");  
fiscalprinter.printRecPackageAdjustment(2, "Dep.3 Surcharge", "3,50000");  
fiscalprinter.printRecPackageAdjustVoid(2, "3,50000");  
fiscalprinter.printRecSubtotal(295000);  
fiscalprinter.printRecTotal(295000, 295000, "0Contante");  
fiscalprinter.endFiscalReceipt(false);
```

5. RT printer specific commands

RT printers have specific commands for RETURN and VOID receipts, that must be sent using DirectIO().

Below the list of commands to use:

- H1=1; H2=138 – RT status
- H1=9; H2=205 – Check document presence for refund and void receipts
- H1=1; H2=078 – Open return and void receipts
- H1=2; H2=050 – Reading return and void total and grand total

In the installation folder, you can find some examples of using these commands, with source code (see "SAMPLE" folder).

Other RT printers peculiarities:

- CapPredefinedPaymentLines property must be set to TRUE, using the SetupPOS application (this is currently the default setup since version V22).
PrintRecTotal "description" parameter must contain the index of one of the fiscal printer's predefined payment descriptions (see note on page 11)
- before printing any fiscal document, it's required to set the VAT table, using SetVatValue and SetVatTable methods
- descriptive lines up to 38 characters are accepted within fiscal receipt, but after the 25th character a line feed is performed by the printer
- the receipt number, "zzzz-nnnn", is composed of:
zzzz = number of daily closures + 1
nnnn = receipt progressive number

6. XML 2.0 new fiscal requirement implementation

- NEW payment types:

According with the new payment type management (three-digit code, the first represents the type and the other two represent the subtype), new codes are now available:

4xx => Ticket with number xx specifications (01-10)

5xx => NON riscossi (not cashed) with following subtypes:

00 = No payment (shared on BENI and SERVIZI types)

01 = No Payment ITEM (BENI)

02 = No Payment SERVICE (SERVIZI)

03 = Receipt followed to Invoice (segue Fattura)

04 = Invoiced (fatture da RT)

05 = SSN receipt (Only for Pharmacy usage)

6xx => Sconto a pagare

xx = 00 => generico

xx = 01 => BUONO MULTIUSO

- NEW modifiers types:

3 additional modifiers type (not covered inside UPOS standard) have been defined:

000 => **ACCONTO**

001 => **OMAGGIO**

002 => **BUONI MONOUSO**

their usage must be handled inside `printRecItemAdjustment()` method as follows:

```
fiscalprinter.printRecItemAdjustment(1, "000ACCONTO", 50000, 1);
```

- NEW GDPR compliance:

New regulations require that some functions implying RT memory access are password protected; the UPOS driver will handle the access automatically, by logging in at DeviceEnabled=TRUE time via a stored and protected password.

The RT default password will be used and stored at the first usage, the application software can modify the password via the provided directIO() method as follows:

```
int[]cmd[0] = 4038;  
StringBuffer sbcmd = new StringBuffer("0112345");  
fiscalprinter.directIO(0, icmd, sbcmd);
```

The driver will store the new password and use it for auto login.

7. SetVatValue

To set the new parameters related to the department, a customization of the SetVatValue () method has been introduced.

This is the syntax to use:

SetVatValue (int vatId, string param)

vatId = vat group number

param = "rrrrddtaa", where:

- rrrr = vat rate, 0000 to 9999, interpreted as 00.01 to 99.99% (two decimal places)
- dd = department number, 01 to 99
- t = sales type, 0 goods / 1 services
- aa = Ateco classification, 00 no Ateco handling / 01 to 03 Ateco index

Some SetVatValue() example in JavaPos:

- fiscalprinter.setVatValue(1, "220001000");

Vat group n. 1 is set to 22,00%

Department n. 01 is set to have: vat group = 1, sales type = goods, ateco = no handling

- fiscalprinter.setVatValue(0, "000013000");

Vat group n. 0 (N4 esente) is NOT set

Department n. 13 is set to have: vat group = 0, sales type = goods, ateco = no handling

- fiscalprinter.setVatValue(2, "100020100");

Vat group n. 2 is set to 10,00%

Department n. 20 is set to have: vat group = 2, sales type = services, ateco = no handling

- fiscalprinter.setVatValue(4, "050018001");

Vat group n. 4 is set to 05,00%

Department n. 18 is set to have: vat group = 4, sales type = goods, ateco = code 01

Default Vat table

VAT ID	VAT VALUE	VAT DESCRIPTION
0	0,00%	N4 - Esente
1	22,00%	22,00%
2	10,00%	10,00%
3	4,00%	4,00%
4	5,00%	5,00%
5	0,00%	0,00%
6	0,00%	0,00%
7	0,00%	0,00%
8	0,00%	0,00%
9	0,00%	0,00%
10	0,00%	N1 - Esclusa
11	0,00%	N2 - Non soggetta
12	0,00%	N3 - Non imponibile
13	0,00%	N5 - Regime del margine
14	0,00%	N6 - Operazione non IVA
15	0,00%	
16	0,00%	
17	0,00%	
18	0,00%	
19	0,00%	

Default Department table

DEPT ID	ASSOCIATED VAT ID	DEPT DESCRIPTION
1	1	PRODOTTO 1
2	2	PRODOTTO 2
3	3	PRODOTTO 3
4	4	PRODOTTO 4
5	1	PRODOTTO 5
6	1	PRODOTTO 6
7	1	PRODOTTO 7
8	1	PRODOTTO 8
9	1	PRODOTTO 9
10	10	PRODOTTO 10
11	11	PRODOTTO 11
12	12	PRODOTTO 12
13	0	PRODOTTO 13
14	13	PRODOTTO 14
15	14	PRODOTTO 15
16	0	PRODOTTO 16
17	0	PRODOTTO 17
18	0	PRODOTTO 18
19	0	PRODOTTO 19
20	0	PRODOTTO 20
21	0	PRODOTTO 21
...	0	PRODOTTO N
99	0	PRODOTTO 99

4. Events

The driver is compliant to the standard UPOS ver. 1.13 for the 4 classes supported with the following events:

Line Display

	<i>Value</i>	<i>Notes</i>
DataEvent	Not supported	
DirectIOEvent	Not supported	
ErrorEvent	Not supported	
OutputCompleteEvent	Not supported	
StatusUpdateEvent	Not supported	

Cash Drawer

	<i>Value</i>	<i>Notes</i>
DataEvent	Not supported	
DirectIOEvent	Not supported	
ErrorEvent	Not supported	
OutputCompleteEvent	Not supported	
StatusUpdateEvent		Supported status: SUE_POWER_ONLINE SUE_POWER_OFF_OFFLINE CASH_SUE_DRAWERCLOSED CASH_SUE_DRAWEROPEN

Fiscal Printer

	<i>Value</i>	<i>Notes</i>
DataEvent	Not supported	
DirectIOEvent	Not supported	
ErrorEvent	Not supported	
OutputCompleteEvent	Not supported	
StatusUpdateEvent		Supported status: SUE_POWER_ONLINE SUE_POWER_OFF_OFFLINE FPTR_SUE_REC_EMPTY FPTR_SUE_REC_NEAREMPTY FPTR_SUE_REC_PAPEROK

Electronic Journal

	<i>Value</i>	<i>Notes</i>
DataEvent	Not supported	
DirectIOEvent	Not supported	
ErrorEvent	Not supported	

OutputCompleteEvent	Not supported	
StatusUpdateEvent		Supported status: SUE_POWER_ONLINE SUE_POWER_OFF_OFFLINE EJ_SUE_MEDIUM_NEAR_FULL EJ_SUE_MEDIUM_FULL EJ_SUE_MEDIUM_REMOVED EJ_SUE_MEDIUM_INSERTED

5. JAVAPOS driver

5.1 WINDOWS O.S.

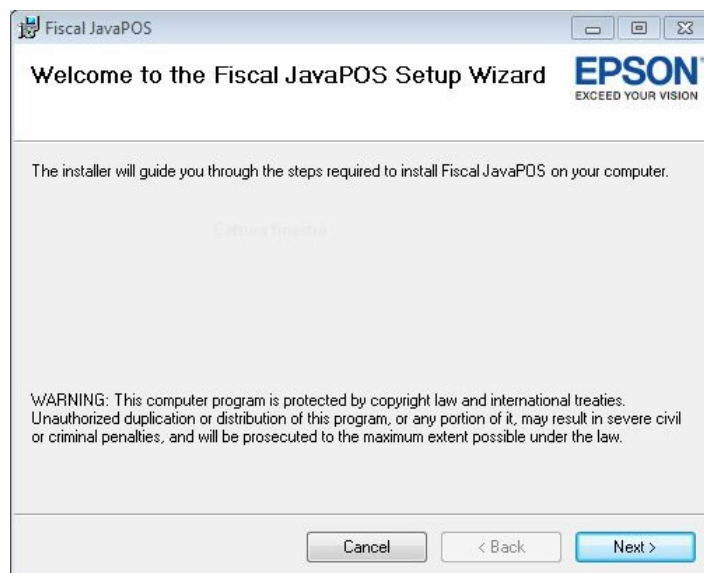
The JavaPOS driver, available for 32/64 bit systems and implemented at Upos level 1.13, needs that the following environment has been set in advance:

- Windows XP+
- JVM (Java Virtual Machine) 1.6+ (32/64 bit, according to the driver version)
- Microsoft framework 2.0

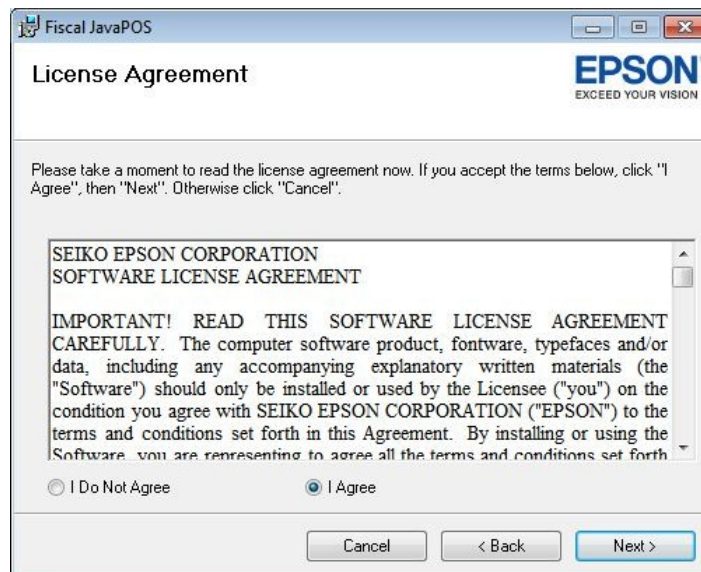
Following the installation of the environment, run the MSI installer file:

EpsonJavaPOSForFiscalSetup_xyy.msi

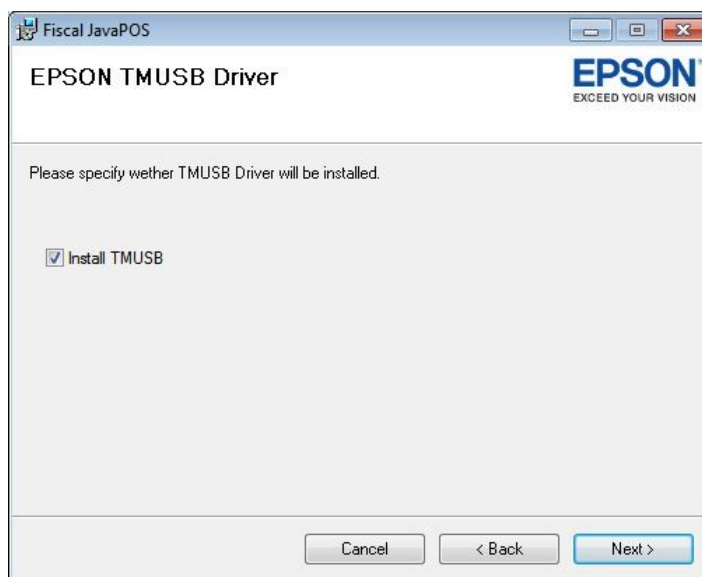
Click "Next"



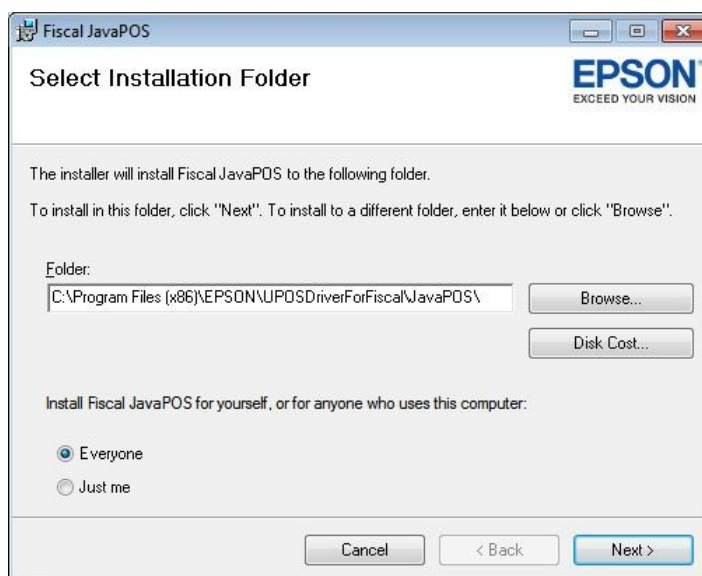
Select "I Agree" and click "Next"



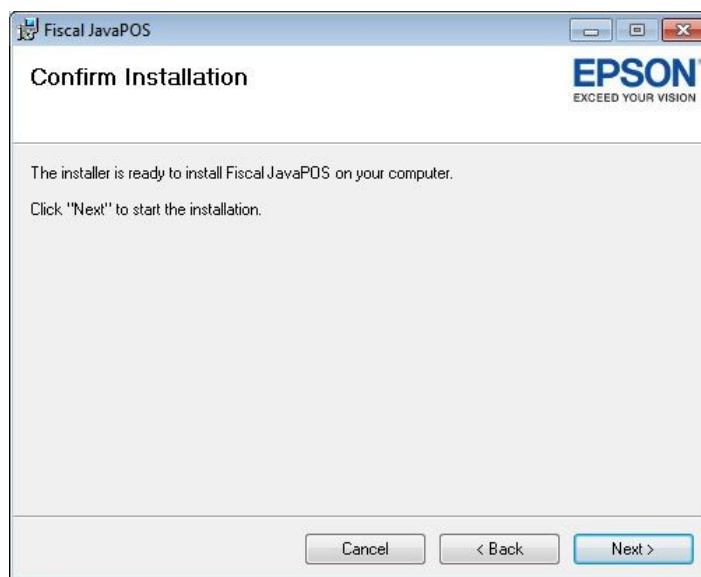
Select "Install TMUSB" if you will use USB connection and click "Next"



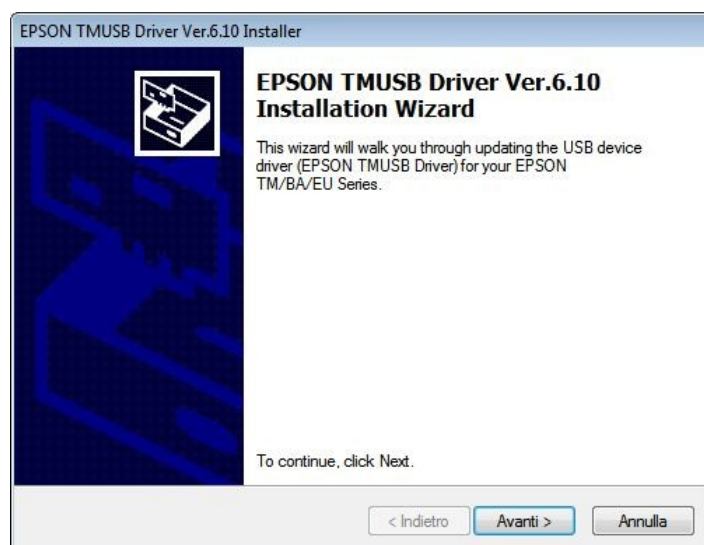
Select installation folder and click "Next"



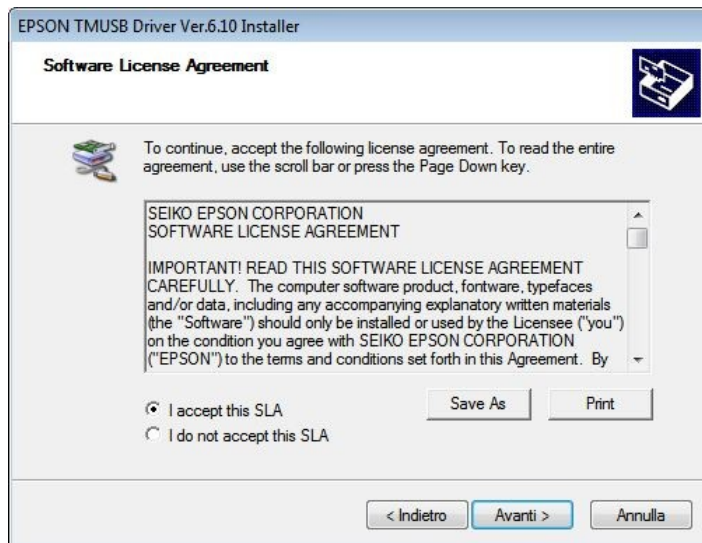
Click "Next"



If you chose to install the TMUSB driver, click "Avanti"



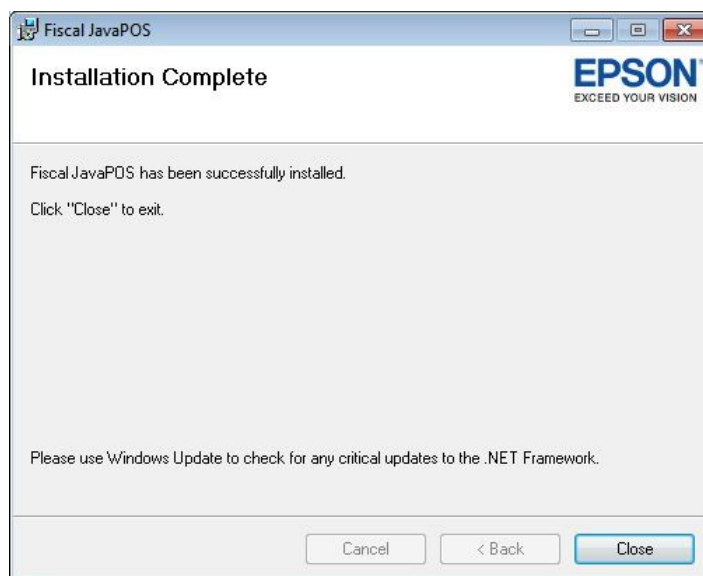
Select "I accept this SLA" and click "Avanti"



Click "Fine"



Click "Close" to complete the installation



RESTART the system

To run the setup program in **silent mode**, add `"/quiet"` command-line option, as shown below:

```
msiexec /i nomefile.msi /qn
```

Usage notes :

- a system path entry is added by the installer
- configuration is done via SetupPos.exe, handling jpos.xml directly
- in order to run use applications, a sample batch file is shown below

```
echo off
echo *****
echo PATH MUST BE ADAPTED TO THE SPECIFIC SYSTEM
echo *****
pause

set classpath="C:\Program Files\epson\UPOSDriverForFiscal\JavaPOS/*"

java -classpath .;%classpath% testClass
```

5.2 LINUX O.S.

The JavaPOS driver, available for 32/64 bit systems and implemented at Upos level 1.13, needs that the following environment has been set in advance:

- Linux / Centos distribution with kernel 2.6.32+
(this kernel is corresponding to Centos 6.7, the floor base for building the driver)
- JVM (Java Virtual Machine) 1.6+ (32/64 bit, according to the driver version)

After downloading :

Epson_JavaPOS_Fiscal_ADK_113[version_number]_for_Linux_x[32/64].zip

according to your system type, extract it and, while in a terminal window opened inside the created folder, please run :

```
sudo ./installJavaPOS.sh
```

(root privileges are required in order to create configuration and trace folders under /var)

Under EPSON_TEST folder a sample script running a test JavaPOS application is also provided.

Configuration instructions are supplied under EPSON_DOCS folder.

Usage notes :

- in order to run user applications, the sample batch file is shown below:

```
export LD_LIBRARY_PATH=./EPSON_LIBS
java -Djava.library.path=./EPSON_LIBS -cp
.:jpos113.jar:jcl.jar:xercesImpl.jar:epsonupos.frame.jar testAlloCE1
```

6. POS for .net driver

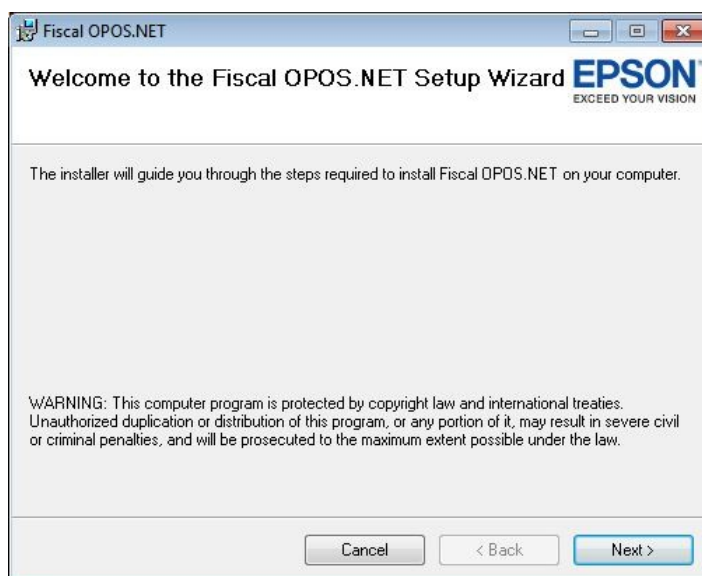
The POS for .net driver, available for 32/64 bit systems and implemented at Upos level 1.13, needs that the following environment has to set in advance:

- Windows XP+
- Microsoft framework 2.0
- Microsoft Point of Service 1.12 (see usage notes for 1.11)

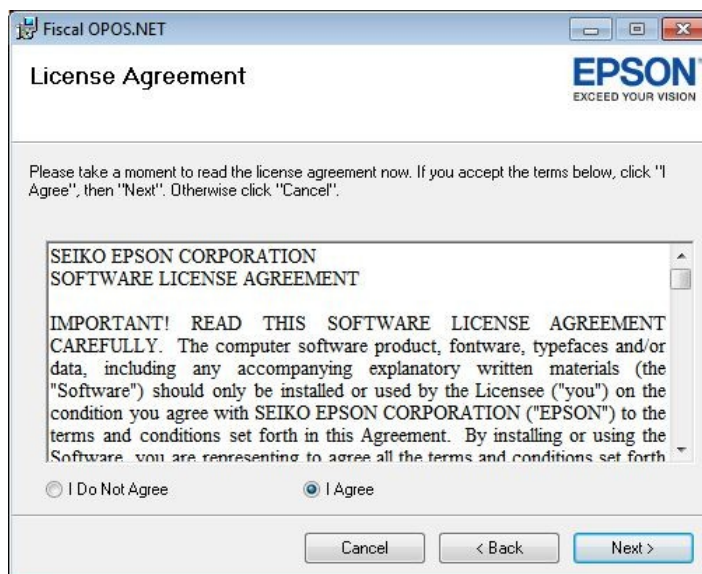
Following the installation of the environment, run the MSI installer file:

EpsonOPOSDotNetForFiscalSetup_xyy.msi

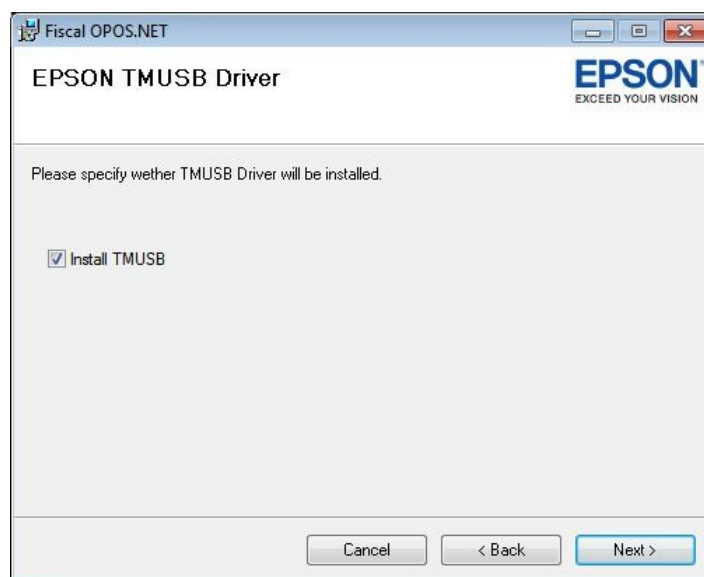
Click "Next"



Select "I Agree" and click "Next"



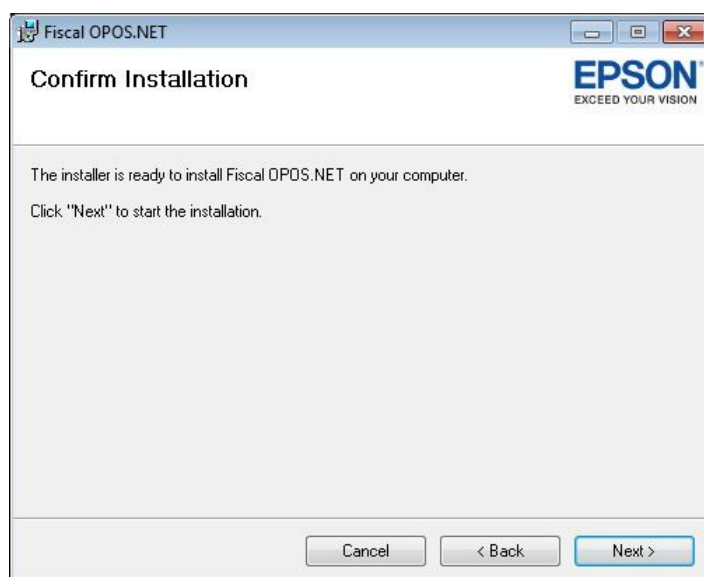
Select "Install TMUSB" if you will use USB connection and click "Next"



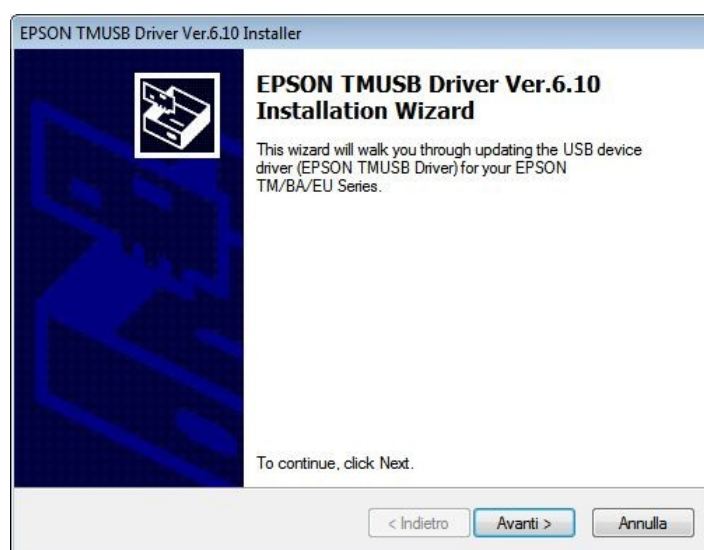
Select installation folder and click "Next"



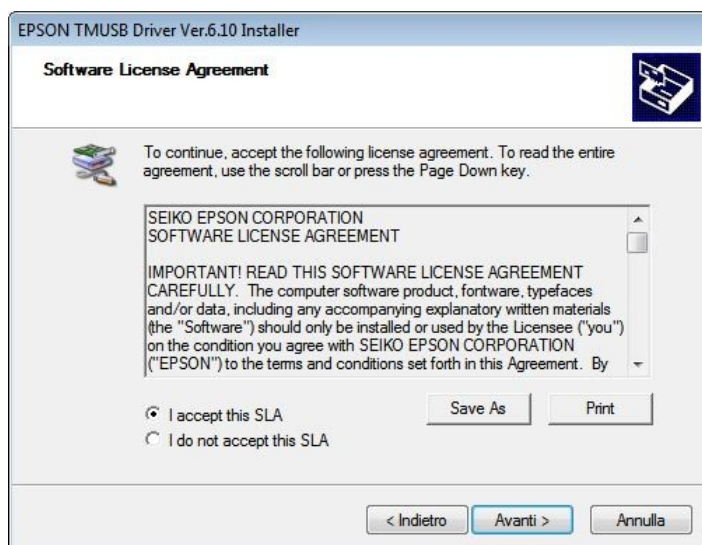
Click "Next"



If you chose to install the TMUSB driver, click "Avanti"



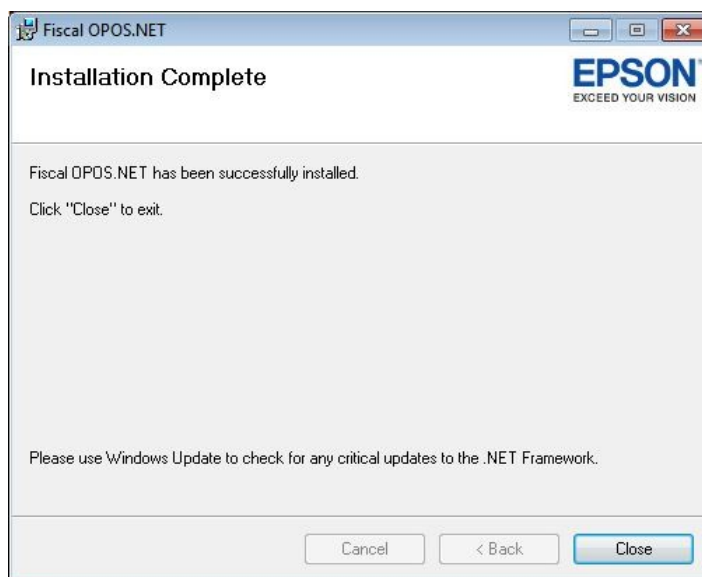
Select "I accept this SLA" and click "Avanti"



Click "Fine"



Click "Close" to complete the installation



RESTART the system

To run the setup program in **silent mode**, add `"/quiet"` command-line option, as shown below:

```
msiexec /i nomefile.msi /qn
```

Usage notes for Microsoft Point of Service 1.11:

If you need to downgrade to Microsoft Point of Service 1.11, you can replace the dll and the exe contained in the SAMPLE folder of installation path (dotNETService_111.zip) with the installed one.

7. OPOS driver

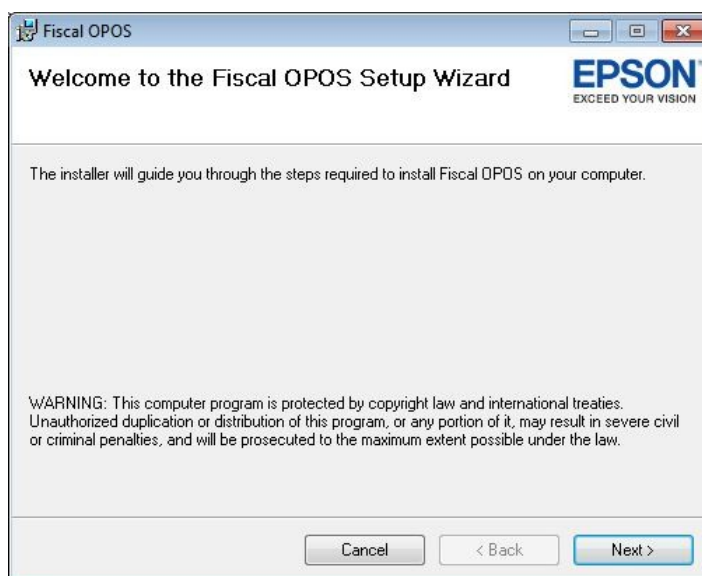
The OPOS driver, available for 32 bit systems and compatible with 64 bit systems, implemented at Upos level 1.13, needs that the following environment has to set in advance:

- Windows XP+
- Microsoft framework 2.0

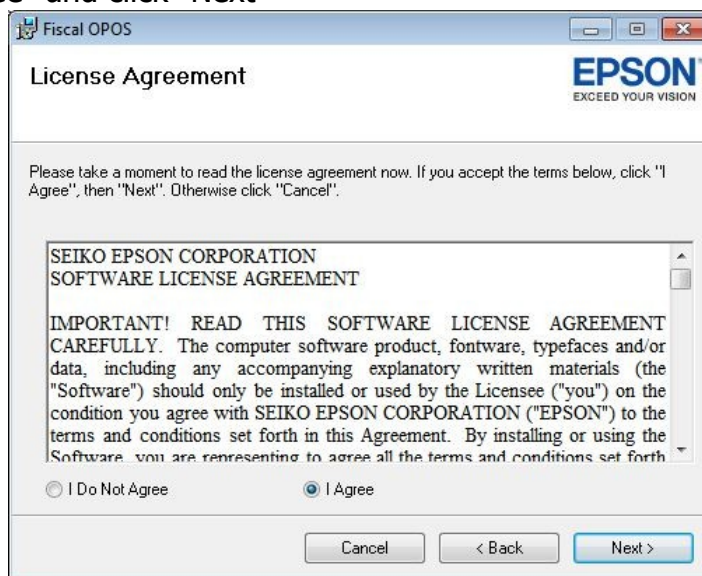
Following the installation of the environment, run the MSI installer file:

EpsonOPOSForFiscalSetup_x86.msi

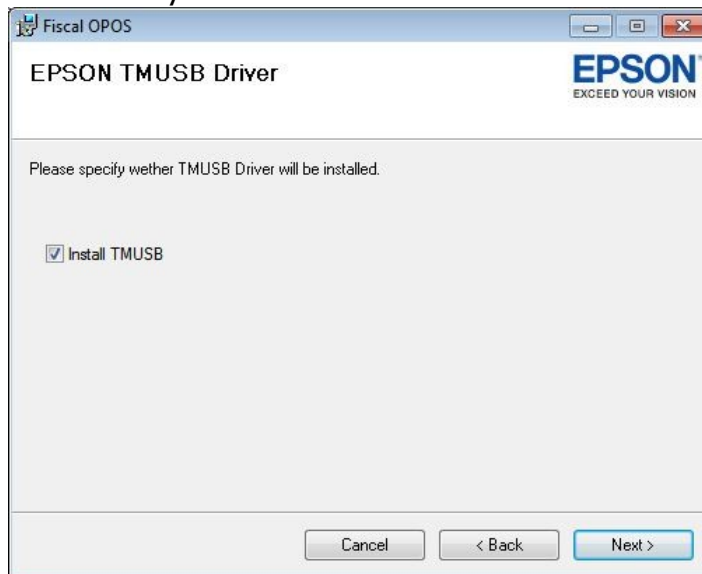
Click "Next"



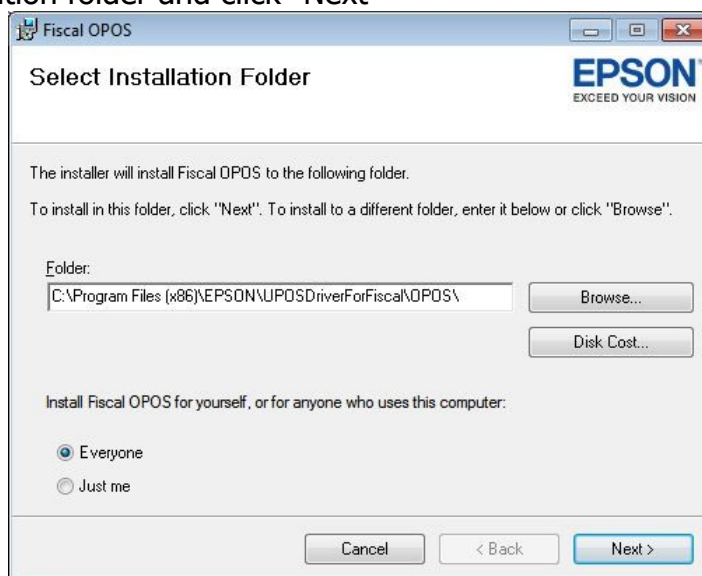
Select "I Agree" and click "Next"



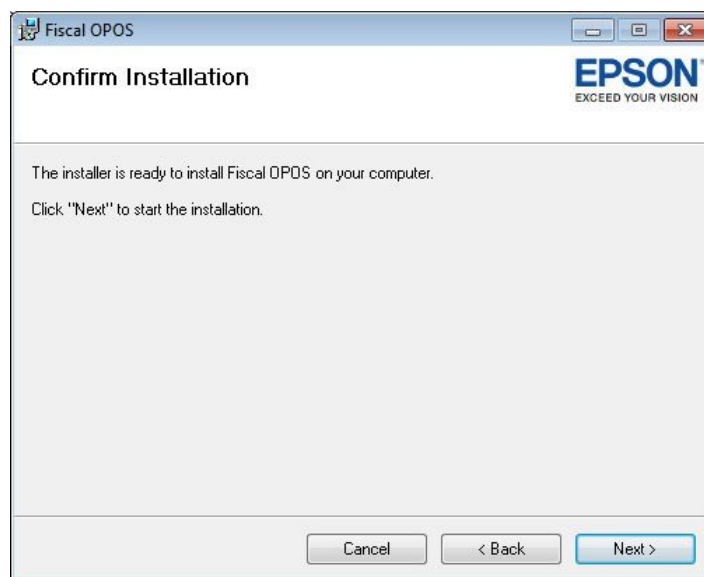
Select "Install TMUSB" if you will use USB connection and click "Next"



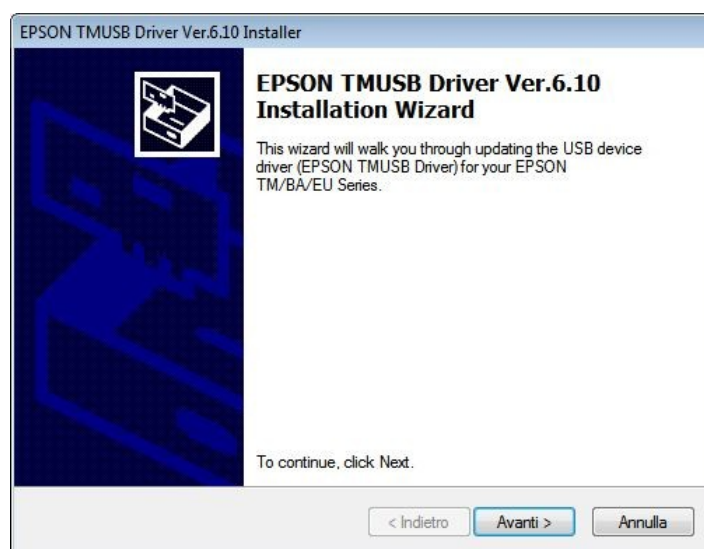
Select installation folder and click "Next"



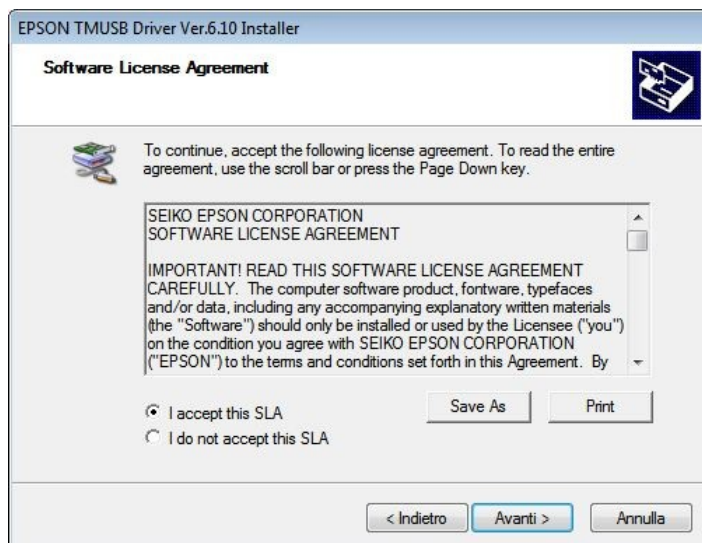
Click "Next"



If you chose to install the TMUSB driver, click "Avanti"



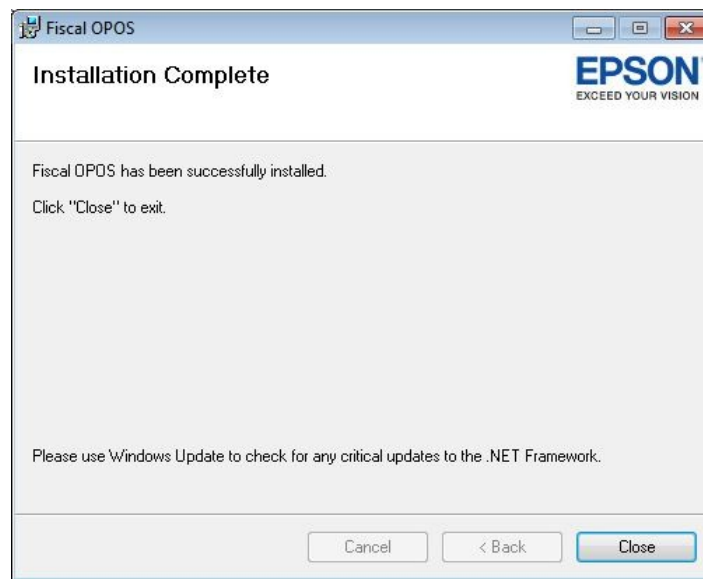
Select "I accept this SLA" and click "Avanti"



Click "Fine"



Click "Close" to complete the installation



RESTART the system

To run the setup program in **silent mode**, add "/quiet" command-line option, as shown below:

```
msiexec /i nomefile.msi /qn
```

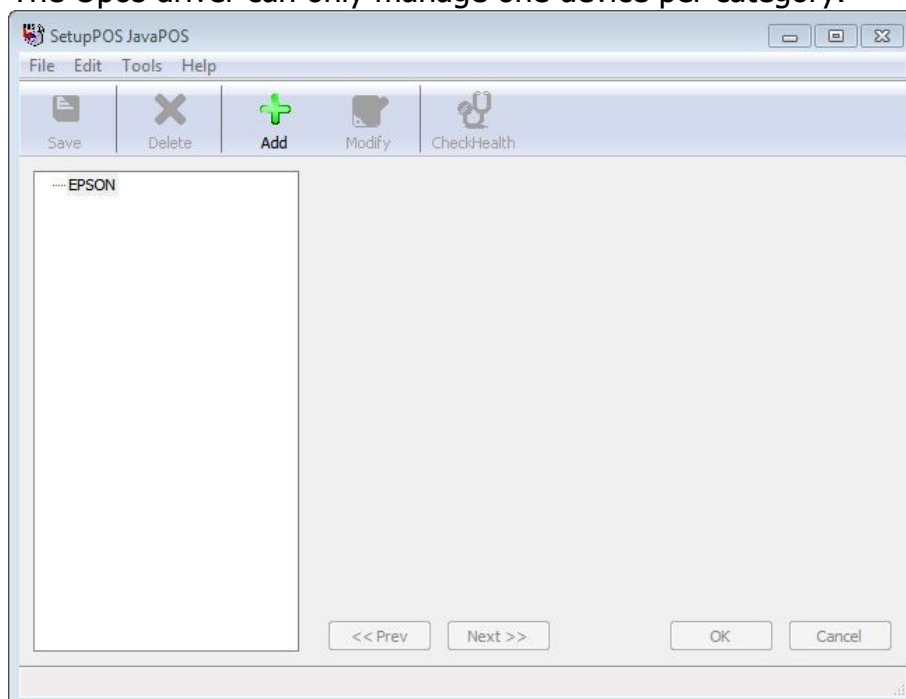
8. Set and test the devices – SetupPOS application (only for Windows O.S.)

To set the devices and test that the driver has been installed correctly, see the following instructions:

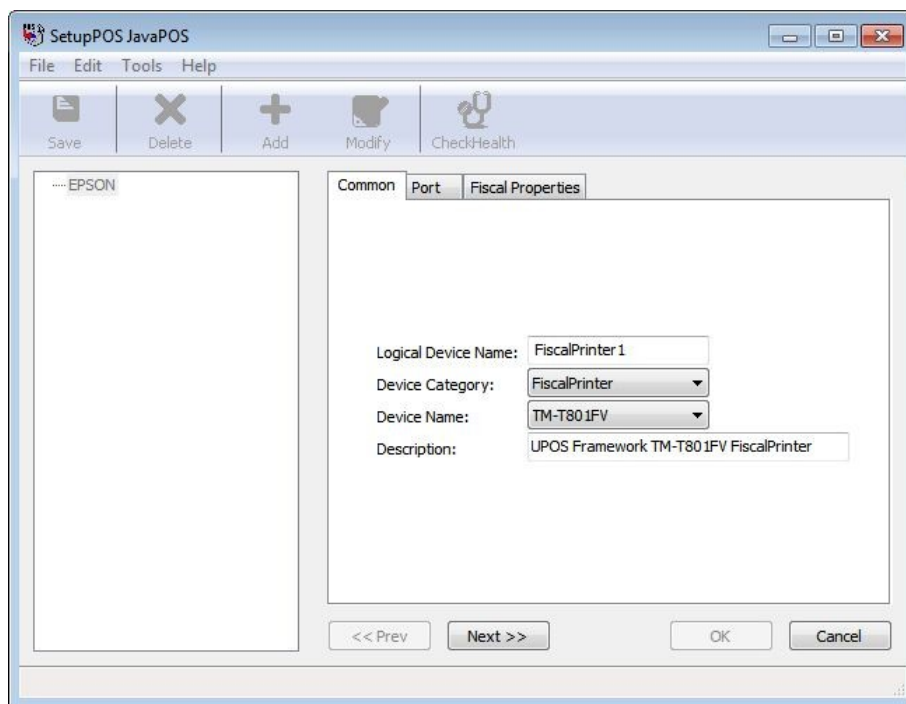
- Connect the serial/usb/tcp cable from fiscal printer
- Run the program: SetupPOS
- Click "Add" to add the devices (FiscalPrinter, LineDisplay, CashDrawer, ElectronicJournal).

The JavaPos and Pos for .net driver can manage up to 10 devices of the same category (10 FiscalPrinter, 10 LineDisplay, 10 CashDrawer, 10 ElectronicJournal).

The Opos driver can only manage one device per category.

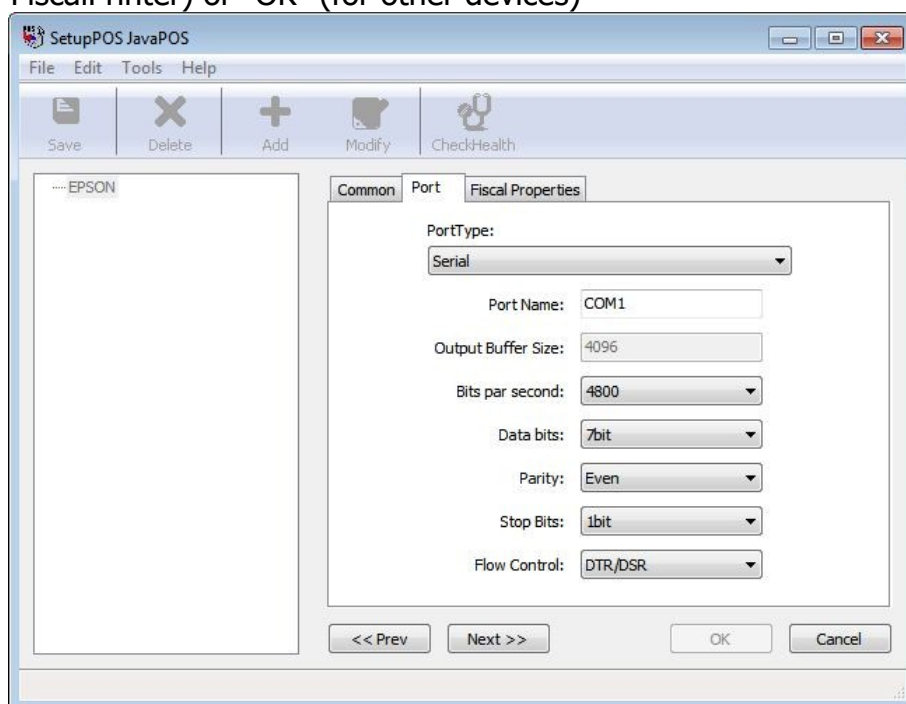


- Select the device category and configure the logical name, then click "Next"

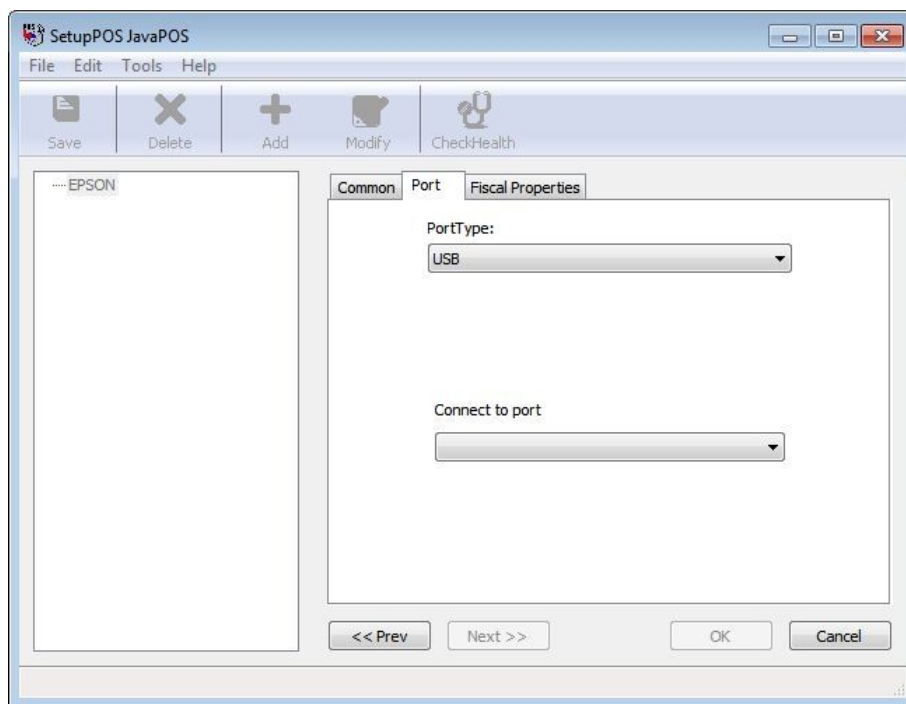


- Configure the printer connection.

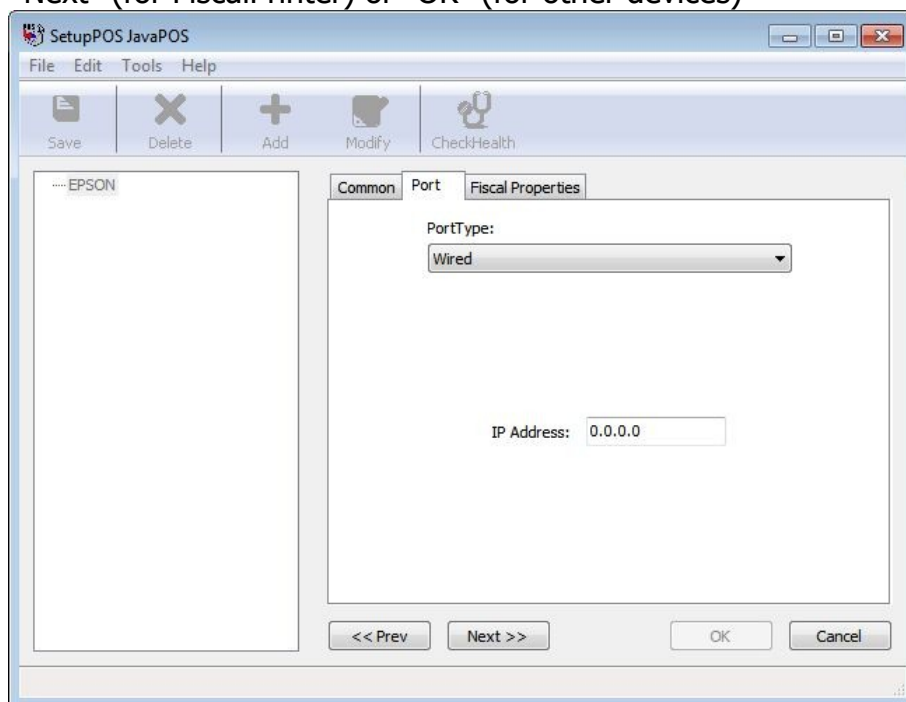
In case of serial port, select "Serial" and set its parameters (port name, bits per second, data bits, parity, stop bits, flow control), then click "Next" (for FiscalPrinter) or "OK" (for other devices)



- In case of USB port, select "USB" and the application automatically searches the available port, then click "Next" (for FiscalPrinter) or "OK" (for other devices)



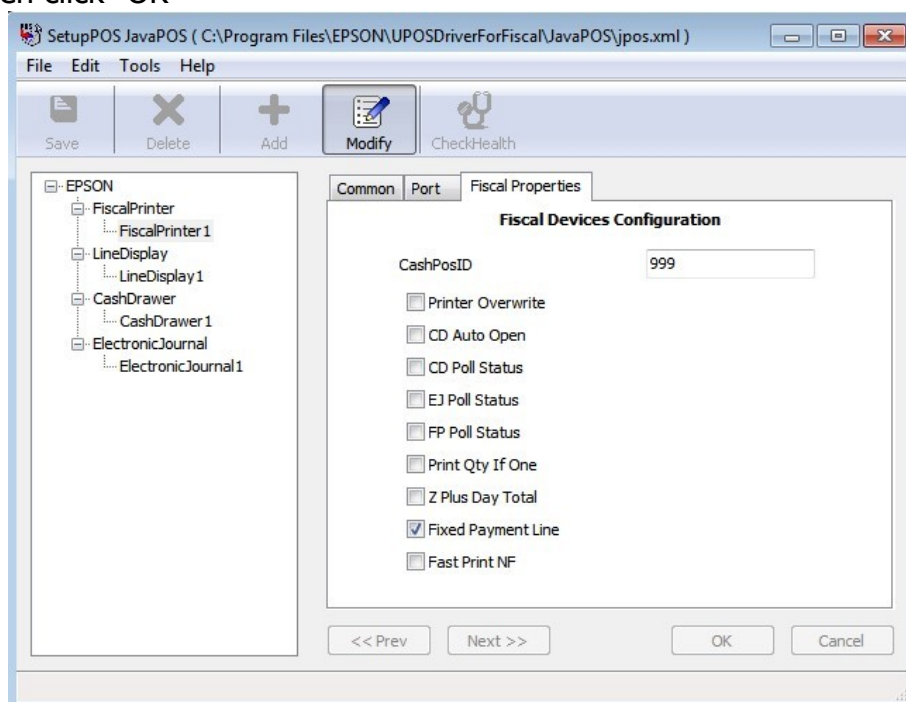
- In case of TCP port, select "Wired" and insert the IP address, then click "Next" (for FiscalPrinter) or "OK" (for other devices)



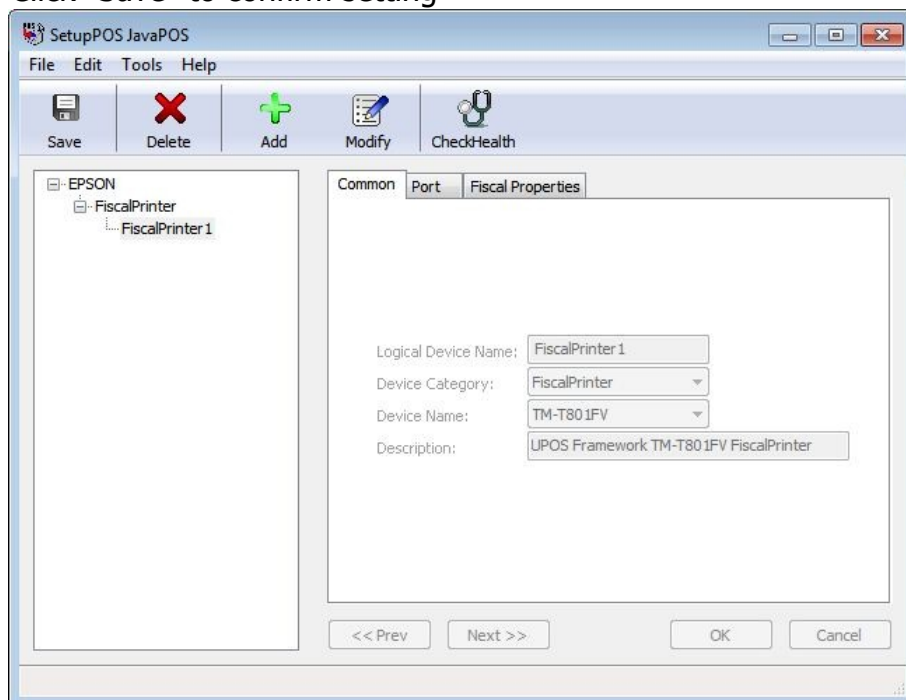
- In case of FiscalPrinter, you can set some properties:
 - Cash POS ID = printer ID (001 – 999), printed on the bottom of the fiscal receipt (refer to SetPOSID() method)
 - Printer Overwrite = if checked, any command sent to the line display from the application override the messages generated by the printer; if not checked, the messages generated by the printer override the messages sent from the application

- CD Auto Open = if checked, enables the automatic drawer opening; if not checked, disables the automatic drawer opening
- CD Poll Status = if checked, enables the sensor "open/close" in the cash drawer (CapStatus = true) and the driver manages a regular polling in order to generate events in case of device state changes; if not checked, disables the cash drawer sensor (CapStatus = false), no polling is performed and no events are returned
- EJ Poll Status = if checked, the driver manages a regular polling in order to generate events in case of device state changes; if not checked, no polling is performed and no events are returned
- FP Poll Status = if checked, the driver manages a regular polling in order to generate events in case of device state changes; if not checked, no polling is performed and no events are returned
- Print Qty If One = if checked, the driver prints the quantity line (before an item) even if is 1; if not checked, the driver does not print the quantity line if is 1
- Z Plus Day Total = if checked, the financial report is printed before the daily closure; if not checked, the financial report is not printed before the daily closure
- Fixed Payment Line = if checked, PrintRecTotal() method accepts a reference or a reference + description in the payment line description (refer to PredefinedPaymentLines property); if not checked, PrintRecTotal() method accepts free description in the payment line description
- Fast Print NF = if checked, the non-fiscal lines of management documents, sent via printNormal() method, will be sent in bulk to the printer via the ESC POS protocol

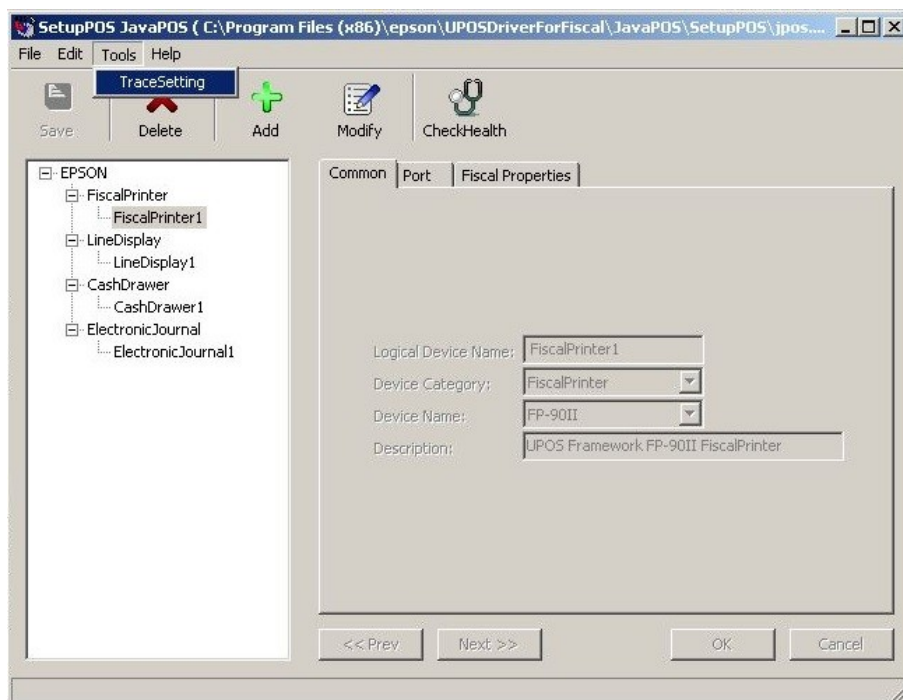
Then click "OK"



- Click "Save" to confirm setting



- Click "CheckHealth" to test the device. If no error is returned, the setting has been successful
- Click "Tools", then click "TraceSetting" to enable/disable tracing, set file name and max size. By default tracing is enabled and saved in path ProgramData\EPSON\UPOSDriverForFiscal\JavaPOS or OPOS.NET or OPOS). The driver generates another trace file (low level), EFIO_current.log, which is stored in path ProgramData\EPSON\EFIO\Trace and can not be customized



In the installation folder, you can find some examples of use of the driver, with source code (see "SAMPLE" folder):

- JavaPos driver has a Java code sample
- POS for .net driver has a C# code sample
- OPOS driver has a Visual Basic code sample

9. Troubleshooting

9.1 WINDOWS O.S.

Trace

In case of errors, it's possible to recover the trace files produced by the driver.

There are two levels of trace:

- the highest level, which generates by default the file "trace.log" in the ProgramData\EPSON\UPOSDriverForFiscal\JavaPOS or OPOS.NET or OPOS path (depending on the installed delivery).

This trace can be activated/deactivated and customized (file name and size) using the SetupPOS application (see previous chapter).

When the file reaches the maximum expected size, it's renamed to "Old_trace.log" and held until the new "trace.log" reaches its maximum size again.

- the lowest level, which generates the "EFIO_current.log" file in the ProgramData\EPSON\EFIO\Trace path.

This trace is always active and can not be customized.

Also in this case, when the file reaches the maximum expected size (5 MB), it's renamed to "EFIO_old.log" and held until the new "EFIO_current.log" reaches the maximum size again.

Polling

The driver has the possibility to perform an automatic polling, to monitor the status of Fiscal Printer, Cash Drawer and Electronic Journal devices (not provided for the Line Display).

Enabling this polling, by the SetupPOS application (see previous chapter), the driver generates Status Update Events during each device status change. For example, in case of Fiscal Printer's paper end, the Upos FPTR_SUE_REC_EMPTY event will be returned.

The application that interfaces with the driver can intercept these status changes and avoid creating its own polling, which usually involves the request of many properties, which could negatively affect the driver's performance.

Driver version

The driver version is visible in the DeviceServiceVersion property or in the "Version" property of the "Programs and features" in the "Control panel".

The first two fields correspond to the Upos version supported by the driver, the third is the specific version of the driver.

For example, the value "1.13.15" means:

- Upos version 1.13
- Epson driver version 15

The "Name" property indicates the installed environment: JavaPos / Opos / DotNet for 32 / 64 bit.

AppData folder permissions

For the correct operation of the driver it is recommended to give the permissions of the AppData folder to all the users.

9.2 LINUX O.S.

Trace

In case of errors, it's possible to recover the trace files produced by the driver.

There are two levels of trace:

- the highest level, which generates by default the file "trace.log" in the /var/epson/uposdriverforfiscal/JavaPOS folder.

This trace can be activated/deactivated and customized by editing RegSettings.xml file (see appendix A).

When the file reaches the maximum expected size, it's renamed to "Old_trace.log" and held until the new "trace.log" reaches its maximum size again.

- the lowest level, which generates the "EFIO_current.log" file in the /var/log/efio/trace folder.

This trace is always active and can not be customized.

Also in this case, when the file reaches the maximum expected size (5 MB), it's renamed to "EFIO_old.log" and held until the new "EFIO_current.log" reaches the maximum size again.

Polling

The driver has the possibility to perform an automatic polling, to monitor the status of Fiscal Printer, Cash Drawer and Electronic Journal devices (not provided for the Line Display).

Enabling this polling, by editing RegSettings.xml file (see appendix A), the driver generates Status Update Events during each device status change. For example, in case of Fiscal Printer's paper end, the Upos FPTR_SUE_REC_EMPTY event will be returned.

The application that interfaces with the driver can intercept these status changes and avoid creating its own polling, which usually involves the request of many properties, which could negatively affect the driver's performance.

Driver version

The driver version can be retrieved requesting the DeviceServiceVersion property.

The first two fields correspond to the Upos version supported by the driver, the third is the specific version of the driver.

For example, the value "1.13.15" means:

- Upos version 1.13
- Epson driver version 15

/var/epson and var/log/efio folders permissions

The setup script, run as root, tries to set the permissions for the correct operation of the driver; depending on the specific distribution setup, additional actions may be needed by the user.

Special characters handling

In case the special characters cannot be printed/displayed, check that the LOCALE is correctly set (for example "export LC_ALL=it_IT").

10. Driver error code

BASE CODE	EXTENDED CODE
101 closed	
102 claimed	
103 not-claimed	
104 no-service	
105 disabled	
106 illegal	
107 no-hardware	
108 offline	
109 no-exist	
110 exist	
111 failure	
112 timeout	
113 busy	
114 EXTENDED	
	201 cover-open
	202 JRN-empty
	203 REC-empty
	204 SLIP-empty
	205 SLIP-form
	206 missing-device(s)
	207 wrong-state
	208 technical-assistance
	209 clock-error
	210 fiscal-memory-full
	211 fiscal-memory-disconnected
	212 fiscal-totals-error
	213 bad-item-quantity
	214 bad-item-amount
	215 bad-item-description
	216 REC-total-overflow
	217 bad-VAT
	218 bad-price
	219 bad-date
	220 negative-total
	221 word-not-allowed
	222 bad-length
	223 Missing-set-currency
	224 Day-end-required

11. Software Rev. History

DATE : 26/10/2016
RELEASE : V7
AUTHOR : SR

- modified NumVatRates property value, according to the FP model and FW version, can be 4 or 9

DATE : 10/05/2017
RELEASE : V8
AUTHOR : SR/AM

- property NumHeaderLines upgraded to 16
- minor fixes

DATE : 09/07/2017
RELEASE : V9
AUTHOR : SR/AM

- minor fixes

DATE : 06/02/2018
RELEASE : V10
AUTHOR : SR/AM

- modified javapos installation procedure
- speedup open and close sequences
- solved compatibility issues in logo loading for OPOS
- added timeout setting for directIO()

DATE : 12/02/2018
RELEASE : V11
AUTHOR : SR/AM

- minor performance fixes

DATE : 19/02/2018
RELEASE : V12
AUTHOR : SR/AM

- minor performance fixes

DATE : 06/06/2018
RELEASE : V13
AUTHOR : SR/AM

- added description of RT specific commands, for return and void receipt
- minor performance fixes

DATE : 12/10/2018

RELEASE : V14
AUTHOR : SR/AM

- added PrintRecTaxID() method
- solved Status Update Events bug
- minor performance fixes

DATE : 05/02/2019
RELEASE : V15
AUTHOR : SR/AM

- included compatibility of RT printers
- minor performance fixes

DATE : 12/06/2019
RELEASE : V16
AUTHOR : SR/AM

- added PrintRecPackageAdjustment() method
- added PrintRecPackageAdjustVoid() method
- CapPackageAdjustment property returns True
- added the management of VAT exempt in the SetVatValue() method
- NumVatRates property, depending on the printer model, can return 4 or 9 or 60
- added management of AdditionalHeader property instead of DirectIO() method, to open refund and void receipts, when using RT models
- added driver version in the DeviceServiceVersion property
- added the ability to manage up to 10 devices of the same category (10 CashDrawer, 10 ElectronicJournal, 10 FiscalPrinter, 10 LineDisplay), only for JavaPos and Pos for .net drivers (not Opos)
- minor performance fixes

DATE : 20/09/2019
RELEASE : V17
AUTHOR : SR/AM

- added Linux JavaPos version
- minor performance fixes

DATE : 30/10/2019
RELEASE : V18
AUTHOR : SR/AM

- PrintRecTotal() method, changed predefined payment description for cash subtype 01 to 05 (from 001 to 005) and credit card subtype 00 (200)
- TrainingModeActive property, in case of RT return the "demo mode" state
- NumVatRates property, return only 4 or 9, but SetVatValue() method accepts values from 1 to NumVatRates and from 21 to 59 (excluding 30, 40, 50)
- minor performance fixes
- provided POS for .NET back compatibility to 1.11 (following specific procedure)

DATE : 05/05/2020
RELEASE : V19
AUTHOR : SR/AM

- GetData() method, in case of RT model, the MID_VOID returns the RT void documents daily total
- GetData() method, in case of RT model, the REFUND returns the RT refund documents daily total
- minor performance fixes

DATE : 14/09/2020
RELEASE : V20
AUTHOR : SR/AM

- GetTotalizer() method revisited for RT VOID and RETURN handling
- implemented some usability fixes
- added support for very high item amounts (only OPOS impact)
- removed POS for NET single thread issue
- unified windows installation process for using only MSIEXEC

DATE : 29/10/2020
RELEASE : V21
AUTHOR : AM

- Implementation of XML 2.0 requirements

DATE : 26/01/2021
RELEASE : V22
AUTHOR : AM

- Modified payment default description
- added some internal safety control
- updated POS for NET versioning

DATE : 05/10/2021
RELEASE : V23
AUTHOR : SR/AM

- modified GetTotalizer() method; FPTR_GT_ITEM, FPTR_GT_ITEM_VOID and FPTR_GT_REFUND returns gross total by vat
- modified GetData() method; FPTR_GD_FISCAL_DOC and FPTR_GD_FISCAL_DOC_VOID are no longer supported (illegal exception is returned)
- modified SetVatValue() method; added management of new parameters (department num, good / service, ateco code) for departments programming
- implemented some usability fixes

DATE : 27/01/2022
RELEASE : V24
AUTHOR : SR/AM

- added "special" DirectIO() for setting totalizer type (-113); next GetTotalizer(ITEM/ITEM_VOID/REFUND) will have the chosen type (0=department - 1=vatID)
- modified GetTotalizer(ITEM/ITEM_VOID/REFUND): if totalizer_type = 1 selection is made by vat id, if totalizer_type = 0 (default) selection is made by department
- modified DeviceEnabled=false; removed auto logout in order to always be logged in and have access to the www/dati-rt folder
- minor performance fixes

DATE : 22/06/2022
RELEASE : V25
AUTHOR : SR/AM

- minor performance fixes

DATE : 17/10/2022
RELEASE : V26
AUTHOR : SR/AM

- minor performance fixes

DATE : 30/11/2023
RELEASE : V27
AUTHOR : SR/AM

- minor performance fixes

DATE : 22/02/2024
RELEASE : V28
AUTHOR : SR/AM

- in SetupPOS application (and consequently in regsettings.xml file), added the “Fast Print NF” flag; if checked, the non-fiscal lines of management documents, sent via printNormal() method, will be sent in bulk to the printer via the ESC POS protocol
- minor performance fixes

Appendix A – Silent installation and devices manual setting

SILENT INSTALLATION (Windows O.S. only)

If the USB connection is needed, install Epson TMUSB in silent mode, following the instructions below.

1. Open a command prompt as ADMINISTRATOR.
2. Type the directory path of the Epson TMUSB setup.exe, then type a space and then type "/quiet".

For example:

"C:\TMUSB\Setup.exe" /quiet

3. Press Enter.

The Setup program runs.

DEVICES MANUAL SETTING

To generate and manage the devices (CashDrawer, ElectronicJournal, FiscalPrinter, LineDisplay) without using SetupPOS program, perform the following operations, according to the environment.

JAVAPOS (x86/x64)

- WINDOWS O.S.

1. Modify the **jpos.xml** file according to your needs (see specific paragraph, with the description of the file contents and the example in the "example_files" folder) and place it in a folder visible from the application (current directory or classpath).
2. Modify the **RegSettings.xml** file according to your needs (see specific paragraph, with the description of the file contents and the examples in the "example_files" folder) and place it in the path C:\ProgramData\EPSON\UPOSDriverForFiscal\JavaPOS (or according to the system variable %ProgramData%).
3. If exists, delete the **IOModulePort.xml** file, which is located in the path C:\ProgramData\EPSON\IOModuleTest (or according to the system variable %ProgramData%). It will be automatically generated by the driver when it is first used.
4. Restart the system.
5. To verify that the installation and the devices setting have been successful, run the JavaPos sample test, contained in the driver installation folder.

When first used, the driver automatically generates the C:\ProgramData\EPSON\IOModuleTest\IOModulePort.xml file, which contains the specifications of the printer connection (for example, in case of USB, it automatically recovers the ID of the USB port).

- LINUX O.S.

1. Modify the **jpos.xml** file according to your needs (see specific paragraph, with the description of the file contents and the example in the "example_files" folder) and place it in a folder visible from the application (current directory or classpath).
2. Modify the **RegSettings.xml** file according to your needs (see specific paragraph, with the description of the file contents and the examples in the "example_files" folder) and place it in the path /var/epson/uposdriverforfiscal/JavaPOS.
3. If exists, delete the **IOModulePort.xml** file, which is located in [user's home]/EPSON/IOModuleTest. It will be automatically generated by the driver when it is first used.
4. Restart the system.
5. To verify that the installation and the devices setting have been successful, run the JavaPos sample test, contained in the driver installation folder.

When first used, the driver automatically generates the :

[*user's home*]/EPSON/IOModuleTest/IOModulePort.xml file, which contains the specifications of the printer connection (for example, in case of USB, it automatically recovers the ID of the USB port).

POS FOR .NET (x86/x64)

1. Modify the **RegSettings.xml** file according to your needs (see specific paragraph, with the description of the file contents and the examples in the "example_files" folder) and place it in the path C:\ProgramData\EPSON\UPOSDriverForFiscal\OPOS.NET (or according to the system variable %ProgramData%).
2. Modify the **Configuration.xml** file according to your needs (see specific paragraph, with the description of the file contents and the example in the "example_files" folder) and place it in the path C:\ProgramData\Microsoft\Point Of Service\Configuration (or according to the system variable %ProgramData%).
3. If exists, delete the **IOModulePort.xml** file, which is located in the path C:\ProgramData\EPSON\IOModuleTest (or according to the system variable %ProgramData%). It will be automatically generated by the driver when it is first used.
4. Restart the system.
5. To verify that the installation and the devices setting have been successful, run the DotNet sample test, contained in the driver installation folder.

When first used, the driver automatically generates the C:\ProgramData\EPSON\IOModuleTest\IOModulePort.xml file, which contains the specifications of the printer connection (for example, in case of USB, it automatically recovers the ID of the USB port).

OPOS (x86)

1. According to the system (32 or 64 bit), modify the **OposData_x86.reg** / **OposData_x64.reg** file according to your needs (see specific paragraph, with the description of the file contents and the examples in the "example_files" folder).
2. Open a command prompt as ADMINISTRATOR.
3. Type "REG IMPORT", then type a space, then type the directory path of the OposData.reg.
For example, for 32 bit system type:
REG IMPORT "C:\Epson\DevicesManualSetting\OposData_x86.reg"

For example, for 64 bit system type:

```
REG IMPORT "C:\Epson\DevicesManualSetting\OposData_x64.reg"
```

4. Press Enter.

The import of registry keys starts.

5. Modify the **RegSettings.xml** file according to your needs (see specific paragraph, with the description of the file contents and the examples in the "example_files" folder) and place it in the path C:\ProgramData\EPSON\UPOSDriverForFiscal\OPOS (or according to the system variable %ProgramData%).
6. If exists, delete the **IOModulePort.xml** file, which is located in the path C:\ProgramData\EPSON\IOModuleTest (or according to the system variable %ProgramData%). It will be automatically generated by the driver when it is first used.
7. Restart the system.
8. To verify that the installation and the devices setting have been successful, run the Opos sample test, contained in the driver installation folder.

When first used, the driver automatically generates the C:\ProgramData\EPSON\IOModuleTest\IOModulePort.xml file, which contains the specifications of the printer connection (for example, in case of USB, it automatically recovers the ID of the USB port).

FILES DESCRIPTION

RegSettings.xml

Scheme and description of the file.

The file differs depending on the type of connection used (serial, usb or wired) and, for the FiscalPrinter device, has specific properties.

Serial connection

```
<?xml version="1.0" encoding="utf-8"?>
<RegSettings>
  <DeviceInformation>
    <CategorySetting Name="" Version="">
      <DeviceSetting Device="" LDN="" Desc="">
        <PropertySetting>
          <Property Name="FixedPaymentLine" Value=""/>
          <Property Name="ZPlusDayTotal" Value=""/>
          <Property Name="PrintQtyIfOne" Value=""/>
          <Property Name="FPPollStatus" Value=""/>
          <Property Name="EJPollStatus" Value=""/>
          <Property Name="CDPollStatus" Value=""/>
          <Property Name="CDAutoOpen" Value=""/>
          <Property Name="PrinterOverwrite" Value=""/>
          <Property Name="CashPosID" Value=""/>
          <Property Name="FastPrintNF" Value=""/>
        </PropertySetting>
        <PortSetting Value="Serial">
          <FlowControl Value="3"/>
          <StopBits Value="0"/>
          <DataBits Value="8"/>
          <Parity Value="0"/>
          <OutputBufferSize Value="4096"/>
          <BaudRate Value="57600"/>
          <PortName Value="COM1"/>
        </PortSetting>
      </DeviceSetting>
    </CategorySetting>
  </DeviceInformation>
  <CommonInformation>
    <CommonProperty Name="TraceFileMaxSize" Value=""/>
    <CommonProperty Name="TraceFileName" Value=""/>
    <CommonProperty Name="TraceEnable" Value=""/>
  </CommonInformation>
</RegSettings>
```

USB connection

```
<?xml version="1.0" encoding="utf-8"?>
<RegSettings>
  <DeviceInformation>
    <CategorySetting Name="" Version="">
      <DeviceSetting Device="" LDN="" Desc="">
        <PropertySetting>
```

```

        <Property Name="FixedPaymentLine" Value=""/>
        <Property Name="ZPlusDayTotal" Value=""/>
        <Property Name="PrintQtyIfOne" Value=""/>
        <Property Name="FPPollStatus" Value=""/>
        <Property Name="EJPollStatus" Value=""/>
        <Property Name="CDPollStatus" Value=""/>
        <Property Name="CDAutoOpen" Value=""/>
        <Property Name="PrinterOverwrite" Value=""/>
        <Property Name="CashPosID" Value=""/>
        <Property Name="FastPrintNF" Value=""/>
    </PropertySetting>
    <PortSetting Value="">
        <PortName Value=""/>
    </PortSetting>
</DeviceSetting>
</CategorySetting>
</DeviceInformation>
<CommonInformation>
    <CommonProperty Name="TraceFileMaxSize" Value=""/>
    <CommonProperty Name="TraceFileName" Value=""/>
    <CommonProperty Name="TraceEnable" Value=""/>
</CommonInformation>
</RegSettings>

```

Wired connection

```

<?xml version="1.0" encoding="utf-8"?>
<RegSettings>
    <DeviceInformation>
        <CategorySetting Name="" Version="">
            <DeviceSetting Device="" LDN="" Desc="">
                <PropertySetting>
                    <Property Name="FixedPaymentLine" Value=""/>
                    <Property Name="ZPlusDayTotal" Value=""/>
                    <Property Name="PrintQtyIfOne" Value=""/>
                    <Property Name="FPPollStatus" Value=""/>
                    <Property Name="EJPollStatus" Value=""/>
                    <Property Name="CDPollStatus" Value=""/>
                    <Property Name="CDAutoOpen" Value=""/>
                    <Property Name="PrinterOverwrite" Value=""/>
                    <Property Name="CashPosID" Value=""/>
                    <Property Name="FastPrintNF" Value=""/>
                </PropertySetting>
                <PortSetting Value="">
                    <IPAddress Value=""/>
                    <PortName Value=""/>
                </PortSetting>
            </DeviceSetting>
        </CategorySetting>
    </DeviceInformation>
    <CommonInformation>
        <CommonProperty Name="TraceFileMaxSize" Value=""/>
        <CommonProperty Name="TraceFileName" Value=""/>
        <CommonProperty Name="TraceEnable" Value=""/>
    </CommonInformation>
</RegSettings>

```

For each device that you want to use (CashDrawer, ElectronicJournal, FiscalPrinter, LineDisplay) an entire <CategorySetting> node must be created.

In case of use of more devices of the same category, an entire <DeviceSetting> node must be created.

COMMON FILEDS

<CategorySetting Name="" Version="">

Field name	Field type	Field value
Name	chosen from a fixed list	assign the specific class, based on the type of device CashDrawer = CashDrawer ElectronicJournal = ElectronicJournal FiscalPrinter = FiscalPrinter LineDisplay = LineDisplay
Version	fixed	01.13.00

<DeviceSetting Device="" LDN="" Desc="">

Field name	Field type	Field value
Device	chosen from a fixed list	Assign the specific value, according to the model of the device. In case of JavaPOS environment, it must match the one assigned in the jpos.xml. In case of DotNet environment, it must match the one assigned in the Configuration.xml. In case of Opos environment, it must match the one assigned in the OposData.reg. CashDrawer = FP-90IICD / FP-90IIICD / FP-81CD / FP-81IICD ElectronicJournal = FP-90IIEJ / FP-90IIIEJ / FP-81EJ / FP-81IIEJ FiscalPrinter = FP-90II / FP-90III / FP-81 / FP-81II LineDisplay = FP-90IILD / FP-90IIILD / FP-81LD / FP-81IILD
LDN	variable	Logical name to assign to the device (must be different for each device). In case of JavaPOS environment, it must match the one assigned in the jpos.xml. In case of DotNet environment, it must match the one assigned in the Configuration.xml. In case of Opos environment, it must match the one assigned in the OposData.reg.
Desc	chosen from a fixed list	assign the specific value, based on the model of the device. In case of JavaPOS environment, it must match the one assigned in the jpos.xml. In case of DotNet environment, it must match the one assigned in the Configuration.xml. CashDrawer = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II CashDrawer ElectronicJournal = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II ElectronicJournal FiscalPrinter = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II FiscalPrinter LineDisplay = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II LineDisplay

<CommonProperty Name="**TraceFileMaxSize**" Value=""/>

Field name	Field type	Field value
Value	variable	assign the maximum number of kilobytes that can reach the trace file

<CommonProperty Name="**TraceFileName**" Value=""/>

Field name	Field type	Field value
Value	variable	Assign the path and the name of the trace file (eg "trace.log")

<CommonProperty Name="**TraceEnable**" Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	specify whether or not to enable the generation of the trace file Trace enabled = TRUE Trace disabled = FALSE

ONLY FOR FISCAL PRINTER

<Property Name="**FixedPaymentLine**" Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	specify whether or not to enable the management of the fixed description of payments Fixed payments enabled = TRUE Fixed payments disabled = FALSE

<Property Name="**ZPlusDayTotal**" Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	specify whether or not enable printing of the X report together with Z report X report printing enabled = TRUE X report printing disabled = FALSE

<Property Name="**PrintQtyIfOne**" Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	specify whether or not to enable the printing of the quantity line if the quantity corresponds to only one unit Print quantity if 1 enabled = TRUE Print quantity if 1 disabled = FALSE

<Property Name="**FPollStatus**" Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	specify whether or not to enable polling on the status of the FiscalPrinter Polling enabled = TRUE Polling disabled = FALSE

<Property Name="**EJPollStatus**" Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	specify whether or not to enable polling on the status of the ElectronicJournal Polling enabled = TRUE Polling disabled = FALSE

<Property Name="**CDPollStatus**" Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	specify whether or not to enable polling on the status of the CashDrawer Polling enabled = TRUE Polling disabled = FALSE

<Property Name="**CDAutoOpen**" Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	specify whether or not to enable automatic opening of the drawer at the end of the receipt Automatic opening enabled = TRUE Automatic opening disabled = FALSE

<Property Name="**PrinterOverwrite**" Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	specify whether or not to enable overwriting of display messages generated automatically by the printer Overwriting enabled = TRUE Overwriting disabled = FALSE

<Property Name="**CashPosID**" Value=""/>

Field name	Field type	Field value
------------	------------	-------------

Value	chosen from a fixed list	specify the printer ID number, from 001 to 999
-------	--------------------------	--

<Property Name=**"FastPrintNF"** Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	specify whether or not the non-fiscal lines of management documents, sent via printNormal() method, will be sent in bulk to the printer via the ESC POS protocol ESC POS for printNormal() enabled = TRUE ESC POS for printNormal() disabled = FALSE

SERIAL CONNECTION

<**PortSetting** Value="">

Field name	Field type	Field value
Value	fixed	Serial

<**FlowControl** Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	assign the specific value, based on the FlowControl to be used Xon/Xoff = 1 DTR/DSR = 2 RTS/CTS = 3

<**StopBits** Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	assign the specific value, based on the StopBits to be used 1bit = 0 1,5bit = 1 2bit = 2

<**DataBits** Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	assign the specific value, based on the DataBits to be used 7bit = 7 8bit = 8

<**Parity** Value=""/>

Field name	Field type	Field value
------------	------------	-------------

Value	chosen from a fixed list	assign the specific value, based on the Parity to be used None = 0 Odd = 1 Even = 2
-------	--------------------------	--

<**OutputBufferSize** Value=""/>

Field name	Field type	Field value
Value	fixed	4096

<**BaudRate** Value="57600"/>

Field name	Field type	Field value
Value	chosen from a fixed list	assign the BaudRate value, between one of the following: 9600 – 19200 – 38400 – 57600 – 115200

<**PortName** Value=""/>

Field name	Field type	Field value
Value	variable	assign the serial port to be used (for example "COM1")

USB CONNECTION

<**PortSetting** Value="">

Field name	Field type	Field value
Value	fixed	USB

<**PortName** Value=""/>

Field name	Field type	Field value
Value	variable	assign the "USB1" port, making sure that NO OTHER Epson USB devices are connected to the system

WIRED CONNECTION

<**PortSetting** Value="">

Field name	Field type	Field value
Value	fixed	Wired

<**IPAddress** Value=""/>

Field name	Field type	Field value
Value	variable	assign the IP address to be used (for example "192.168.1.250")

<**PortName** Value=""/>

Field name	Field type	Field value
Value	variable	assign the IP address to be used (for example "192.168.1.250")

jpos.xml (only for JavaPos)

Scheme and description of the file.

```
<?xml version="1.0" encoding="UTF-8"?>
<JposEntries>
  <JposEntry logicalName="">
    <creation factoryClass="" serviceClass=""/>
    <vendor name="" url=""/>
    <jpos category="" version=""/>
    <product description="" name="" url=""/>
    <prop name="PhysicalDevice" type="String" value=""/>
    <prop name="DeviceDesc" type="String" value=""/>
    <prop name="DriverName" type="String" value=""/>
    <prop name="InstallFolderPath" type="String" value=""/>
  </JposEntry>
</JposEntries>
```

For each device that you want to use (CashDrawer, ElectronicJournal, FiscalPrinter, LineDisplay) an entire <JposEntry> node must be created.

In case of use of more devices of the same category, an entire <JposEntry> node must be created.

Pay particular attention to the correctness of the "InstallFolderPath" parameter.

<JposEntry logicalName="">

Field name	Field type	Field value
logicalName	variable	logical name to assign to the device (must be different for each device)

<creation factoryClass="" serviceClass=""/>

Field name	Field type	Field value
factoryClass	fixed	jp.co.epson.FiscalDriver.EpsonJposFiscalServiceInstanceFactory
serviceClass	chosen from a fixed list	assign the specific class, based on the type of device - CashDrawer = jp.co.epson.FiscalDriver.CashDrawer.CommonCashDrawerService - ElectronicJournal = jp.co.epson.FiscalDriver.EJ.CommonEJService - FiscalPrinter = jp.co.epson.FiscalDriver.FiscalPrinter.CommonFiscalPrinterService - LineDisplay = jp.co.epson.FiscalDriver.LineDisplay.CommonLineDisplayService

<vendor name="" url=""/>

Field name	Field type	Field value
name	fixed	SEIKO EPSON
url	fixed	empty ("")

<**jpos** category="" version=""/>

Field name	Field type	Field value
category	chosen from a fixed list	assign the specific class, based on the type of device - CashDrawer = CashDrawer - ElectronicJournal = ElectronicJournal - FiscalPrinter = FiscalPrinter - LineDisplay = LineDisplay
version	fixed	01.13.00

<**product** description="" name="" url=""/>

Field name	Field type	Field value
description	chosen from a fixed list	assign the specific class, based on the type of device - CashDrawer = EPSON CashDrawer Device Service - ElectronicJournal = EPSON ElectronicJournal Device Service - FiscalPrinter = EPSON FiscalPrinter Device Service - LineDisplay = EPSON LineDisplay Device Service
name	fixed	EPSON Services for JavaPOS(TM) Standard
url	fixed	empty ("")

<prop name="**PhysicalDevice**" type="String" value=""/>

Field name	Field type	Field value
value	chosen from a fixed list	assign the specific class, based on the device's model - CashDrawer = FP-90IICD / FP-90IIICD / FP-81CD / FP-81IICD - ElectronicJournal = FP-90IIEJ / FP-90IIIEJ / FP-81EJ / FP-81IIIEJ - FiscalPrinter = FP-90II / FP-90III / FP-81 / FP-81II - LineDisplay = FP-90IILD / FP-90IIILD / FP-81LD / FP-81IILD

<prop name="**DeviceDesc**" type="String" value=""/>

Field name	Field type	Field value
value	chosen from a fixed list	assign the specific class, based on the device's model - CashDrawer = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II CashDrawer - ElectronicJournal = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II ElectronicJournal - FiscalPrinter = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II FiscalPrinter - LineDisplay = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II LineDisplay

<prop name="**DriverName**" type="String" value=""/>

Field name	Field type	Field value
value	fixed	EPSON UPOS FiscalDriver

<prop name="**InstallFolderPath**" type="String" value=""/>

Field name	Field type	Field value
value	variable	full installation path to the driver root folder (for example "C:\Program Files\EPSON\UPOSDriverForFiscal\JavaPOS") Note: this value is only relevant for Windows O.S.

Configuration.xml (only for Pos for .NET)

Scheme and description of the file.

```
<?xml version="1.0" encoding="utf-8"?>
<PointOfServiceConfig Version="1.0">
  <ServiceObject Type="" Name="">
    <Device HardwarePath="">
      <LogicalName Name=""/>
      <Property Name="DeviceDesc" Value=""/>
      <Property Name="DriverName" Value=""/>
      <Property Name="InstallFolderPath" Value=""/>
      <Property Name="PhysicalDevice" Value=""/>
      <Property Name="LogicalName" Value=""/>
    </Device>
  </ServiceObject>
</PointOfServiceConfig>
```

For each device that you want to use (CashDrawer, ElectronicJournal, FiscalPrinter, LineDisplay) an entire **<ServiceObject>** node must be created.

In case of use of more devices of the same category, an entire **<Device>** node must be created.

<ServiceObject Type="" Name="">

Field name	Field type	Field value
Type	chosen from a fixed list	assign the specific category, based on the type of device CashDrawer = CashDrawer ElectronicJournal = ElectronicJournal FiscalPrinter = FiscalPrinter LineDisplay = LineDisplay
Name	chosen from a fixed list	assign the specific category name, based on the type of device CashDrawer = EPSON CashDrawer ElectronicJournal = EPSON ElectronicJournal FiscalPrinter = EPSON FiscalPrinter LineDisplay = EPSON LineDisplay

<Device HardwarePath="">

Field name	Field type	Field value
HardwarePath	variable	assign the value of the connection, depending on the type. For example: Serial = COM1 USB = USB1 Wired = 192.168.1.250

<LogicalName Name=""/>

Field name	Field type	Field value
Name	variable	logical name to assign to the device (must be different for each device)

<Property Name="**DeviceDesc**" Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	assign the specific value, based on the model of the device CashDrawer = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II CashDrawer ElectronicJournal = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II ElectronicJournal FiscalPrinter = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II FiscalPrinter LineDisplay = UPOS Framework FP-90II / FP-90III / FP-81 / FP-81II LineDisplay

<Property Name="**DriverName**" Value=""/>

Field name	Field type	Field value
Value	fixed	EPSON UPOS Fiscal Driver

<Property Name="**InstallFolderPath**" Value=""/>

Field name	Field type	Field value
Value	variable	full installation path to the driver root folder (for example "C:\Program Files\EPSON\UPOSDriverForFiscal\OPOS.NET")

<Property Name="**PhysicalDevice**" Value=""/>

Field name	Field type	Field value
Value	chosen from a fixed list	assign the specific value, based on the model of the device CashDrawer = FP-90IICD / FP-90IIICD / FP-81CD / FP-81IICD ElectronicJournal = FP-90IIEJ / FP-90IIIEJ / FP-81EJ / FP-81IIIEJ FiscalPrinter = FP-90II / FP-90III / FP-81 / FP-81II LineDisplay = FP-90IILD / FP-90IIILD / FP-81LD / FP-81IILD

<Property Name="**LogicalName**" Value=""/>

Field name	Field type	Field value
Value	variable	logical name to assign to the device (must be different for each device)

OposData_x86.reg / OposData_x64.reg (only for Opos)

Scheme and description of the file.

The file differs depending on the type of system used, 32 or 64 bit.

32 bit system

REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail]

[HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS]

[HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS**DeviceType**
"LogicalName"="DeviceModel"]

[HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS\DeviceType\DeviceModel]
@="ServiceObject"
"DriverName"="EPSON UPOS FiscalDriver"
"InstallFolderPath"="InstallFolderPath"

For each device that you want to use (CashDrawer, ElectronicJournal, FiscalPrinter, LineDisplay) an entire [HKEY_LOCAL_MACHINE\SOFTWARE\OLEforRetail\ServiceOPOS] key registry must be created.

Field name	Field type	Field value
DeviceType	chosen from a fixed list	assign the specific class, based on the type of device CashDrawer = CashDrawer ElectronicJournal = ElectronicJournal FiscalPrinter = FiscalPrinter LineDisplay = LineDisplay
LogicalName	variable	Logical name to assign to the device (must be different for each device)
DeviceModel	chosen from a fixed list	Assign the specific value, according to the model of the device. CashDrawer = FP-90IICD / FP-90IIICD / FP-81CD / FP-81IICD ElectronicJournal = FP-90IIEJ / FP-90IIIEJ / FP-81EJ / FP-81IIIEJ FiscalPrinter = FP-90II / FP-90III / FP-81 / FP-81II LineDisplay = FP-90IILD / FP-90IIILD / FP-81LD / FP-81IILD
ServiceObject	chosen from a fixed list	Assign the specific value, based on the type of device. CashDrawer = SOEPSON.CashDrawerService.1130 ElectronicJournal = SOEPSON.ElectronicJournalService.1130 FiscalPrinter = SOEPSON.FiscalPrinterService.1130 LineDisplay = SOEPSON.LineDisplayService.1130
InstallFolderPath	variable	full installation path to the driver root folder (for example "C:\\Program Files (x86)\\EPSON\\UPOSDriverForFiscal\\OPOS")

64 bit system

REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\OLEforRetail]

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\OLEforRetail\ServiceOPOS]

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\OLEforRetail\ServiceOPOS**DeviceType**]
"LogicalName" = "DeviceModel"

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\OLEforRetail\ServiceOPOS**DeviceType****DeviceModel**]

@ = **"ServiceObject"**

"DriverName" = "EPSON UPOS FiscalDriver"

"InstallFolderPath" = **"InstallFolderPath"**

For each device that you want to use (CashDrawer, ElectronicJournal, FiscalPrinter, LineDisplay) an entire [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\OLEforRetail\ServiceOPOS] key registry must be created.

Field name	Field type	Field value
DeviceType	chosen from a fixed list	assign the specific class, based on the type of device CashDrawer = CashDrawer ElectronicJournal = ElectronicJournal FiscalPrinter = FiscalPrinter LineDisplay = LineDisplay
LogicalName	variable	Logical name to assign to the device (must be different for each device)
DeviceModel	chosen from a fixed list	Assign the specific value, according to the model of the device. CashDrawer = FP-90IICD / FP-90IIICD / FP-81CD / FP-81IICD ElectronicJournal = FP-90IIEJ / FP-90IIIEJ / FP-81EJ / FP-81IIIEJ FiscalPrinter = FP-90IIF / FP-90IIIF / FP-81 / FP-81IIF LineDisplay = FP-90IILD / FP-90IIILD / FP-81LD / FP-81IILD
ServiceObject	chosen from a fixed list	Assign the specific value, based on the type of device. CashDrawer = SOEPSON.CashDrawerService.1130 ElectronicJournal = SOEPSON.ElectronicJournalService.1130 FiscalPrinter = SOEPSON.FiscalPrinterService.1130 LineDisplay = SOEPSON.LineDisplayService.1130
InstallFolderPath	variable	full installation path to the driver root folder (for example "C:\Program Files\EPSON\UPOSDriverForFiscal\JavaPOS")