

EPSON FISCAL DRIVER SPECIFICATION

1	INTRODUCTION.....	4
2	LIBRARIES.....	5
2.1	FOR WINDOWS (32 AND 64 BITS).....	5
2.2	FOR LINUX (32 AND 64 BITS).....	5
2.3	FOR ANDROID.....	5
3	INSTALLATION.....	6
3.1	LINUX.....	6
3.2	WINDOWS.....	6
3.3	ANDROID.....	6
4	CODE ERROR.....	7
5	PROGRAMMING INTERFACE.....	8
5.1	PROPERTIES.....	8
5.1.1	ComPort.....	8
5.1.2	BaudRate.....	8
5.1.3	State.....	8
5.1.4	LastError.....	9
5.1.5	FiscalStatus.....	9
5.1.6	PrinterStatus.....	9
5.1.7	ReturnCode.....	10
5.1.8	ProtocolType.....	10
5.1.9	ExtraFieldsCount.....	10
5.1.10	APIVersion.....	10
5.1.11	SentFrame.....	11
5.1.12	ReceivedFrame.....	11
5.2	METHODS.....	12
5.2.1	setComPort.....	12
5.2.2	getComPort.....	12
5.2.3	setBaudRate.....	13
5.2.4	getBaudRate.....	13
5.2.5	getState.....	14
5.2.6	getLastError.....	14
5.2.7	getPrinterStatus.....	15
5.2.8	getFiscalStatus.....	15
5.2.9	getReturnCode.....	16
5.2.10	OpenPort.....	16
5.2.11	ClosePort.....	17
5.2.12	Purge.....	17
5.2.13	AddDataField.....	18
5.2.14	SendCommand.....	18

5.2.15	<i>GetExtraField</i>	19
5.2.16	<i>GetSentFrame</i>	20
5.2.17	<i>GetReceivedFrame</i>	20
5.2.18	<i>GetAPIVersion</i>	21
5.2.19	<i>getPermission</i>	22
5.2.20	<i>getUSBHostPort</i>	22
5.2.21	<i>setProtocolType</i>	23
5.2.22	<i>getProtocolType</i>	23
6	EXAMPLES.....	24

1 Introduction

This document describes the properties and methods of the EpsonFiscalDriver.

2 Libraries

2.1 For Windows (32 and 64 bits)

- EpsonFiscalDriver.dll – Dynamic Link Library (DLL)
- EpsonFiscalDriver.lib – Static Library (LIB)
- comEpsonFiscalDriver.ocx – Component Object Model (COM)

2.2 For Linux (32 and 64 bits)

- libEpsonFiscalDriver.a – Static lib (LIB)
- libEpsonFiscalDriver.so – Dynamic lib (LIB)

2.3 For Android

- EpsonFiscalDriver.aar – Android archive library(aar)

3 Installation

3.1 Linux

Static Linking – Just link the lib (.a file) during the application building process.

Dynamic Linking – Link and distribute the lib (.so file) with your executable file.

3.2 Windows

Install the drivers package running EpsonFiscalDriver_01.09.00.exe.

After that, OCX can be referenced and used by the application.

The dynamic/static libraries (DLL and LIB) are used by the application in the same way as Linux ones.

Note that, in all cases, EpsonFiscalDriver_01.09.00.exe should also be installed in the PC where the application is executed.

3.3 Android

In your project add EpsonFiscalDriver.aar as dependencies and after that can be imported and used by the application.

This is a mandatory import: `import epon.epsonfiscaldriver.EpsonFiscalDriver.`

4 Code Error

Error Code (dec)	Description
0	Success
1	Port in use
2	Invalid port number
3	Port already open
4	Port already close
5	Initialization error
6	Port closed
7	Maximum length reached
8	Invalid data field
9	Operation in progress
10	Write error
11	Timeout Error
12	Byte 0x15 received
13	Invalid index

5 Programming Interface

5.1 Properties

The properties are only available for COM (.ocx) library.

5.1.1 ComPort

Defines the communication port where the printer is connected.

5.1.1.1 Access

Read and Write.

5.1.1.2 Value type

Numeric.

5.1.1.3 Notes

The value of the port can be: 0=USB, 1=Com1, 2=Com2...

.

5.1.2 BaudRate

Defines the communication speed.

5.1.2.1 Access

Read and Write.

5.1.2.2 Value type

Numeric.

5.1.2.3 Notes

The value of the baudrate can be: 9600(default), 19200, 38400, 57600 and 115200.

5.1.3 State

Gets the communication state.

5.1.3.1 Access

Read.

5.1.3.2 Value type

Numeric.

5.1.3.3 Notes

Close=0, Idle=1, Busy=2

5.1.4 **LastError**

Gets the last communication error.

5.1.4.1 Access

Read.

5.1.4.2 Value type

Numeric.

5.1.4.3 Notes

See the error codes table.

5.1.5 **FiscalStatus**

Gets the last fiscal status.

5.1.5.1 Access

Read.

5.1.5.2 Value type

Numeric.

5.1.5.3 Notes

Only the least significant 16 bits are valid.

5.1.6 **PrinterStatus**

Get the last printer status.

5.1.6.1 Access

Read.

5.1.6.2 Value type

Numeric.

5.1.6.3 Notes

Only the least significant 16 bits are valid.

5.1.7 ReturnCode

Get the last command's return code.

5.1.7.1 Access

Read.

5.1.7.2 Value type

Numeric.

5.1.7.3 Notes

Only the least significant 16 bits are valid.

5.1.8 ProtocolType

Defines the protocol type(compatible or extended).

5.1.8.1 Access

Read and Write.

5.1.8.2 Value type

Numeric.

5.1.8.3 Notes

Compatible protocol use value 0.

Extended protocol use value 1(default).

5.1.9 ExtraFieldsCount

Gets the command's extra fields number.

5.1.9.1 Access

Read

5.1.9.2 Value type

Numeric.

5.1.9.3 Notes

Nothing.

5.1.10 APIVersion

Gets the API version.

5.1.10.1 Access

Read.

5.1.10.2 Value type

String

5.1.10.3 Notes

Version format is XX.YY.ZZ

5.1.11 SentFrame

Gets the last frame (protocol frame) that was sent to the printer.

5.1.11.1 Access

Read.

5.1.11.2 Value type

String

5.1.11.3 Notes

Nothing.

5.1.12 ReceivedFrame

Gets the last frame (protocol frame) received from the printer.

5.1.12.1 Access

Read.

5.1.12.2 Value type

String

5.1.12.3 Notes

Nothing.

5.2 Methods

5.2.1 setComPort

This method configures the port number to be used. It is not available in the COM (.ocx) library.

Prototype:

void setComPort(int Port)

Inputs:

Variable	Type	Value	Description
Port	int	Numeric	0 – USB 1 – COM1 2 – COM2 x - COMx

Outputs:

Nothing.

Returns:

Nothing.

Prerequisites:

None.

5.2.2 getComPort

This method opens the communication port. It is not available in the COM (.ocx) library.

Prototype:

int getComPort(void)

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Port number.

Prerequisites:

None.

5.2.3 setBaudRate

This method configures the communication baudrate. It is not available in the COM (.ocx) library.

Prototype:

```
void setBaudRate( int BaudRate )
```

Inputs:

Variable	Type	Value	Description
BaudRate	int	numeric	9600 (default), 19200, 38400, 57600, 115200

Outputs:

Nothing.

Returns:

Nothing.

Prerequisites:

None.

5.2.4 getBaudRate

This method returns the value of the communication baudrate. It is not available in the COM (.ocx) driver.

Prototype:

```
int getBaudRate( void )
```

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Communication baudrate.

Prerequisites:

None.

5.2.5 getState

This method returns the communication state. It is not available in the COM (.ocx) library.

Prototype:

int getState(void)

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Communication state.

Close=0, Idle=1, Busy=2

Prerequisites:

None.

5.2.6 getLastError

This method returns the last communication error. It is not available in the COM (.ocx) library.

Prototype:

int getLastError(void)

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Last communication error.

See the error table.

Prerequisites:

None.

5.2.7 getPrinterStatus

This method returns the last printer status. It is not available in the COM (.ocx) library.

Prototype:

int getPrinterStatus(void)

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Only the least significant 16 bits are valid.

Prerequisites:

None.

5.2.8 getFiscalStatus

This method returns the last fiscal status. It is not available in the COM (.ocx) library.

Prototype:

int getFiscalStatus(void)

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Only the least significant 16 bits are valid

Prerequisites:

None.

5.2.9 getReturnCode

This method returns the last command's return code. It is not available in the COM (.ocx) library.

Prototype:

int getReturnCode(void)

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Only the least significant 16 bits are valid

Prerequisites:

None.

5.2.10 OpenPort

This method opens the communication port.

Prototype:

void OpenPort(void)

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Nothing.

Prerequisites:

Port must be closed

Communication port and baudrate must be previously issued.

5.2.11 ClosePort

This method closes the communication port.

Prototype:

void ClosePort(void)

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Nothing.

Prerequisites:

Port must be opened.

5.2.12 Purge

This method cleans output buffer.

Prototype:

void Purge(void)

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Nothing.

Prerequisites:

None.

5.2.13 AddDataField

This method adds a new field (binary data) for the next command.

Prototype:

```
void AddDataField( char *in_buffer, int in_buffer_length )
```

Inputs:

Variable	Type	Value	Description
in_buffer	char	Binary	Max size buffer is 8192
in_buffer_length	int	Numeric	Max length is 82192

Outputs:

Nothing.

Returns:

Nothing.

Prerequisites:

None.

5.2.14 SendCommand

This method sends a command to the printer.

Prototype:

```
void SendCommand( void )
```

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Nothing.

Prerequisites:

None.

5.2.15 GetExtraField

This method returns one of the last command's extra fields.

Prototype:

DLL/LIB:

```
void GetExtraField( int field_number, char* output_buffer, int output_buffer_length, int* output_buffer_final_length)
```

COM:

```
String GetExtraField( int field_number)
```

Inputs

DLL/LIB:

Variável	Tipo	Conteúdo	Descrição
field_number	int	Numeric	
output_buffer_length	int	Numeric	Max length is 51200

Inputs

COM:

Variável	Tipo	Conteúdo	Descrição
field_number	int	Numeric	

Outputs

DLL/LIB:

Variável	Tipo	Conteúdo	Descrição
output_buffer	char	Binary	Max size buffer is 51200
output_buffer_final_length	int	Numeric	

Returns

DLL/LIB: Nothing.

COM: String is OLEString type.

Prerequisites:

None.

5.2.16 GetSentFrame

Returns the protocol frame that was sent to the printer. It is not available in the COM (.ocx) library.

Prototype:

```
void GetSentFrame( char* output_buffer, int output_buffer_length, int* output_buffer_final_length )
```

Inputs:

Variável	Tipo	Conteúdo	Descrição
output_buffer_length	int	Numeric	Max length is 51200

Outputs:

Variável	Tipo	Conteúdo	Descrição
output_buffer	char	Binary	Max size buffer is 51200
output_buffer_final_length	int	Numeric	

Returns:

Nothing.

Prerequisites:

None.

5.2.17 GetReceivedFrame

Returns the protocol frame that was received from the printer. It is not available in the COM (.ocx) library.

Prototype:

```
void GetReceivedFrame( char* output_buffer, int output_buffer_length, int* output_buffer_final_length )
```

Inputs:

Variável	Tipo	Conteúdo	Descrição
output_buffer_length	int	Numeric	Max length is 51200

Outputs:

Variável	Tipo	Conteúdo	Descrição
output_buffer	char	Binary	Max size buffer is 51200
output_buffer_final_length	int	Numeric	

Returns:

Nothing.

Prerequisites:

None.

5.2.18 GetAPIVersion

This method returns the API version number. It is not available in the COM (.ocx) library.

Prototype:

void GetAPIVersion(char* api_version)

Inputs:

Nothing.

Outputs:

Variável	Tipo	Tam. Min	Tam. Max	Conteúdo	Descrição
api_version	char	1	8	Alfanumeric	XX.YY.ZZ

Returns:

Nothing.

Prerequisites:

None.

5.2.19 getPermission

This method returns a string with information about device and request a permission to use USB port. It is only available in ANDROID.

Prototype:

String getPermission(void)

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

String with device information.

Prerequisites:

The device must have a USB host(check in specification device). Use the function getUSBHostPort to check it.

The function getPermission is mandatory and must to be used before the function OpenPort. After execution will be appears a dialog box. Press "OK" to allow the use of USB port.

5.2.20 getUSBHostPort

This method check if device has a USB Host. It is only available in ANDROID.

Prototype:

boolean getUSBHostPort(void)

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

True – device has a USB host

False – device has not a USB host

Prerequisites:

If the device has not a USB Host the communication will fail.

5.2.21 setProtocolType

This method configures the protocol type. It is not available in the COM (.ocx) library.

Prototype:

```
void setProtocolType( int Protocol )
```

Inputs:

Variable	Type	Value	Description
Protocol	int	numeric	0(compatible protocol) 1(extended protocol)

Outputs:

Nothing.

Returns:

Nothing.

Prerequisites:

None.

5.2.22 getProtocolType

This method returns the value of the protocol type. It is not available in the COM (.ocx) driver.

Prototype:

```
int getProtocolType( void )
```

Inputs:

Nothing.

Outputs:

Nothing.

Returns:

Protocol type.

Prerequisites:

None.

6 Examples

Code examples can be found into the **.\Examples** folder, showing the use of the methods and properties. It is very important to correctly check the return codes and exceptions.