

EPSON[®]
EXCEED YOUR VISION



Epson FP Mate

Development guide

REVISION HISTORY

| | | | | |
|---|-----|--|-------------------------------|-------------|
| A | All | preliminary | March, 1st 2005 | C. Guietti |
| B | All | First edition | March, 15th 2005 | C. Radaelli |
| C | All | Add some clarification sentences | May, 27 th 2005 | C. Radaelli |
| D | All | Management of display messages; Misspelling corrections; Review of the sample files | June, 9 th 2005 | C. Guietti |
| E | All | Added "Open Drawer" command Added modification of "Not allowed words" Added BarCode Printing | August, 5 th 2005 | C. Guietti |
| F | All | Added "Open Drawer" command also by file. Added "Command" file management | August, 10 th 2005 | C. Guietti |
| | | | | |
| | | | | |

Index

| | | |
|---------|---|----|
| 1 | Overview | 4 |
| 2 | List of related documents..... | 4 |
| 3 | Abbreviation | 4 |
| 4 | EpsonFpMateConfig..... | 5 |
| 4.1 | Input Options | 6 |
| 4.2 | Output Options..... | 7 |
| 4.3 | Printer Options | 8 |
| 5 | EpsonFpMate in Free Running mode..... | 9 |
| 6 | Input File | 11 |
| 6.1 | Input File specification | 11 |
| 6.2 | Printer selection..... | 11 |
| 6.3 | File body | 12 |
| 6.3.1 | Records managed regardless the kind of file. | 12 |
| 6.3.1.1 | Text message for customer display | 12 |
| 6.3.2 | File body for FISCAL RECEIPT..... | 13 |
| 6.3.2.1 | Sale | 13 |
| 6.3.2.2 | Good return from the customer | 13 |
| 6.3.2.3 | Sale cancel | 14 |
| 6.3.2.4 | Discount on the sale..... | 14 |
| 6.3.2.5 | Discount on the Subtotal..... | 15 |
| 6.3.2.6 | Print Subtotal..... | 15 |
| 6.3.2.7 | Additional Row Description..... | 16 |

| | | |
|---------|--|----|
| 6.3.2.8 | Print of Bar Codes | 17 |
| 6.3.2.9 | Payment section..... | 18 |
| 6.3.3 | Forbidden words in text messages on the fiscal Receipt. | 18 |
| 6.4 | File body for "Nota Di Credito" FISCAL RECEIP | 20 |
| 6.4.1 | Opening a "Nota Di Credito" FISCAL RECEIP | 20 |
| 6.4.2 | Closure of a "Nota Di Credito" FISCAL RECEIP | 20 |
| 6.5 | File body for NOT FISCAL RECEIPT | 21 |
| 6.5.1 | Begin Non fiscal receipt..... | 21 |
| 6.5.2 | Print Bar Code..... | 21 |
| 6.5.3 | Print row..... | 21 |
| 6.5.4 | Close Not fiscal receipt..... | 22 |
| 6.6 | File body for DAILY CLOSURE | 22 |
| 7 | Output File for Fiscal Receipt..... | 27 |
| 8 | Examples of TXT file..... | 28 |
| 8.1 | Example of fiscal receipt..... | 28 |
| 8.2 | Example of "Nota di Credito" fiscal receipt | 28 |
| 8.3 | Example of Z report command file..... | 29 |
| 8.4 | Example of Not fiscal receipt..... | 29 |
| 9 | Example of XML file..... | 31 |
| 9.1 | Example of fiscal receipt..... | 31 |
| 9.2 | Example of "Nota di Credito" fiscal receipt | 31 |
| 9.3 | Example of Zreport file command..... | 32 |
| 9.4 | Example of not fiscal receipt..... | 32 |
| 9.5 | Example of Exit application file command | 32 |

1 Overview

Aim of this manual is to provide information to the Sw designers who want to interface the Epson fiscal printer using the EpsonFPMate utility.

Epson FPMate is a software utility designed for the software houses who want to use Epson fiscal printer in a very easy way, all the communication protocol management are under the control of EpsonFPMate.

Epson FP printer can be easily managed by means of simple TXT file or XML one, based on the result of the print-out, EpsonFPMate return a file with the information relating the print-out of last fiscal receipt or the result of the last command.

Epson FPMate is provided with two utilities:

- EpsonFPMateConfig: used to set the environment
- EpsonFPMate: used to manage the fiscal print

EpsonFPMate has been developed under the Windows .net technology, using WIN98 or WIN2K operating system it is need to load on your computer the framework.

2 List of related documents

FP xxx manuale operatore
ActiveX user manual

3 Abbreviation

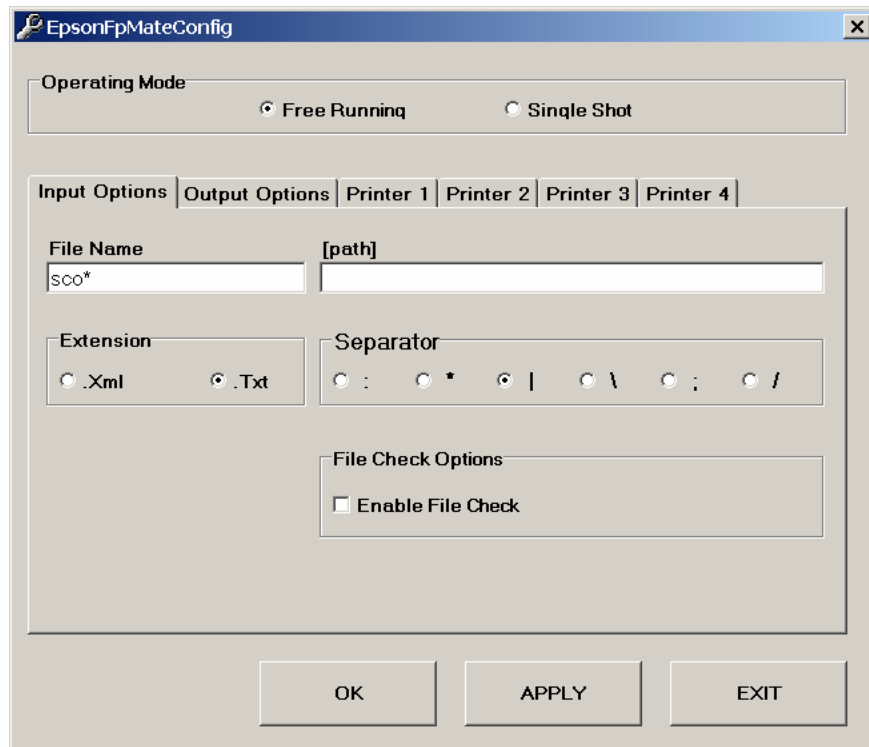
DGFE: Dispositivo Giornale di Fondo Elettronico (Electronic Journal)

4 EpsonFpMateConfig

Allows to create and edit the "Settings.xml" file that is used by the application EpsonFpMate.

There are 3 kinds of form to be set:

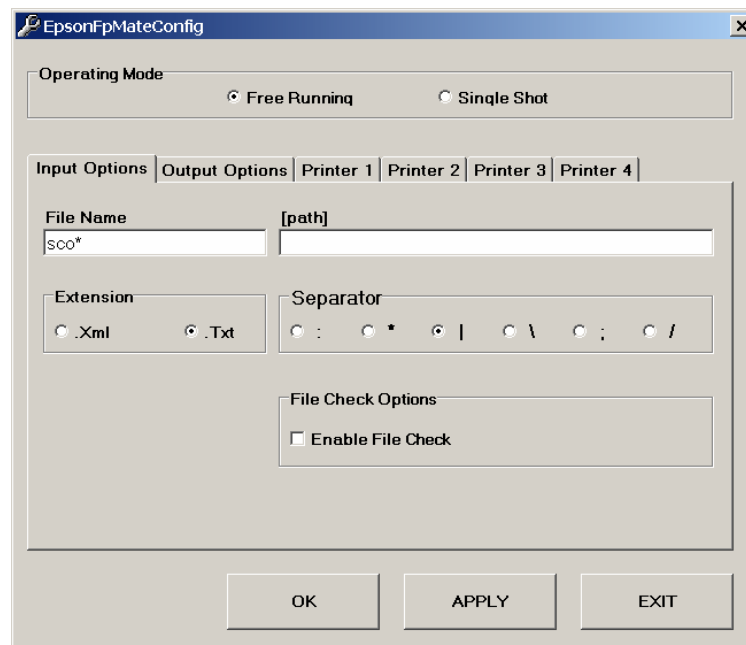
- Input option
- Output option
- Printer (1,2,3 e 4) setting



4.1 Input Options

It allows to define:

- Operating mode (FreeRunning or SingleShot).
 - In SingleShot mode, the application must be started by a command that have as argument the file that must be processed
 - In FreeRunning mode, the application, once started looks, once a second, for files to be processed in a programmed folder.
- In FreeRunning mode, it must be indicate the path where the input files will located and the rule defined for the name.
- In FreeRunning mode, the input files will be deleted after processing.



- Extension: input file format must be:
 - Xml
 - Txt, if selected TXT file format, it must be indicated also the separator to be used
- File Check options: MD5 check, if selected, requires that the input file must be write together with the check file created using the MD5 algorithm.

4.2 Output Options

By means of this folder can be selected the output file and condition to be work for the EpsonFPMate.

Output file: if enabled it must be indicated the path were the file must be saved.

Input Log File: if enabled it must be indicated where the file named InputLogFpMate with all the input files will be saved.

Output Log File: if enabled this file report all the error condition.

The screenshot shows the 'EpsonFpMateConfig' dialog box with the 'Output Options' tab selected. The 'Operating Mode' section has 'Free Running' selected. The 'Input Options' tab is also visible. The 'Output File' section has 'Enabled' checked and a path field. The 'Input Log File' section has 'Enabled' checked, 'File Name' set to 'InputLogFpMate', 'Extension' set to '.txt', and a path field. The 'Output Log File' section has 'Enabled' checked, 'File Name' set to 'OutLogFpMate', 'Extension' set to '.txt', and a path field. The 'Max Log File Size (kbytes)' is set to 1024. At the bottom are 'OK', 'APPLY', and 'EXIT' buttons.

EpsonFpMateConfig

Operating Mode
☒ Free Running ☐ Single Shot

Input Options Output Options Printer 1 Printer 2 Printer 3 Printer 4

☒ Enable Message Boxes Max Log File Size (kbytes) 1024

Output File
☒ Enabled [path]

Input Log File
☒ Enabled File Name InputLogFpMate Extension .txt
[path]

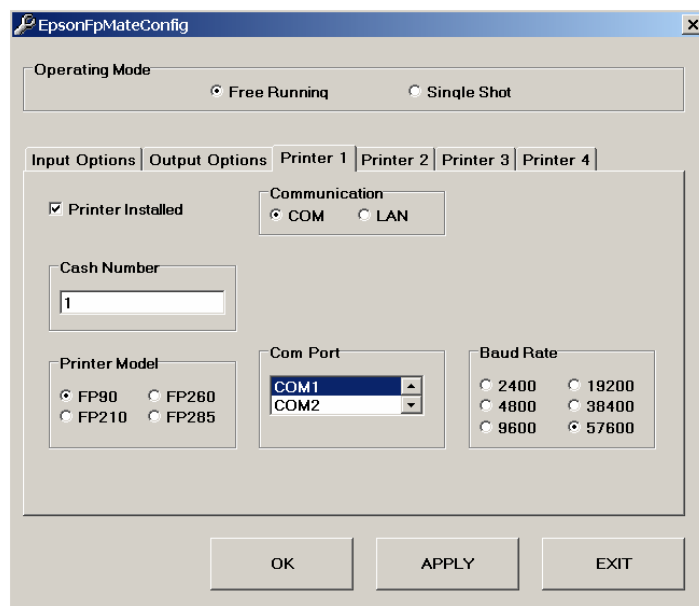
Output Log File
☒ Enabled File Name OutLogFpMate Extension .txt
[path]

OK APPLY EXIT

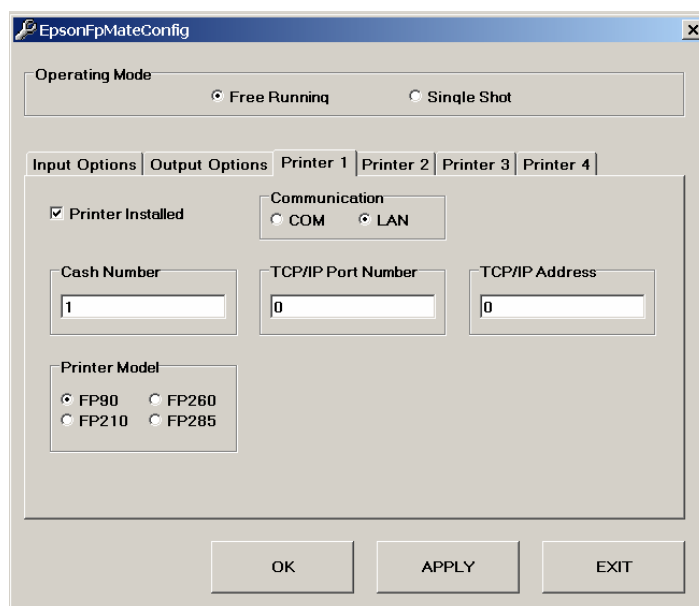
4.3 Printer Options

- It allows to manage up to four printers:
- For each printer, if installed, it must be selected:
 - The printer model:
 - Communication mode (RS232 or TCP/IP) and the related parameters
 - CashDesk Id number: range 0 - 999

In case of serial communication it must be selected the printer model and the communication parameters



In case of LAN interface communication it must be selected the printer model and the LAN PORT with IP address.



5 EpsonFpMate in Free Running mode

Once the system has been configured (by means of EpsonFpMateConfig) the application EpsonFpMate can be started.

The icon remains visible in the system tray.

Once started the application looks for files and, if found, it processes them sequentially.

According to the settings it generates output and/or log files.

The application can be closed using the appropriate button.



The application can be also used to perform local management. Local management can be enabled by means of the "Abilita Comandi" button.

In local management mode, it allows to send to the printer commands in order to perform:

- Fiscal closures
- Date and time settings
- Printing of DGFE
- Reading of
 - Printer status
 - DGFE
 - Fiscal amount
- Close a opened fiscal/Not fiscal receipt
- Void a fiscal receipt opened
- Open the Cash drawer

While in "Local Management Mode", the input files are not managed so they will be processed as soon as the application returns in normal mode.

By means of Giornale di Fondo relating option it is possible to read/print the content of DGFE.

Giornale di Fondo Elettronico

Modalità di Accesso

☒ Mediante Data di inizio e di fine

☐ Mediante Numero di inizio e di fine

Data di Inizio **Data di Fine**

14/03/2005 14/03/2005

Start Exit

6 Input File

EpsonFpMate is able to manage two different kind of file format: TXT or XML one. When used Txt files, setting operations require also to define a separator character used to separate the different fields of each record

Input file for must be composed by three parts:

- Input file specification
- Printer selection
- File body

6.1 Input File specification

One record (First record of the file). It allows to recognize what kind of file has to be managed

| Type of Input File | Keyword |
|-------------------------|----------------------|
| Printer set up (TBD) | printerPresetting |
| Fiscal receipt | printerFiscalReceipt |
| Daily closure (Zreport) | printerFiscalReport |
| Not fiscal receipt | printerNonFiscal |
| Read data (TBD) | printerDataRequest |
| Command | printerCommand |

Example:

- Txt Mode: printerFiscalReceipt
- Xml Mode: In Xml mode that keyword represents the root

6.2 Printer selection

One record (Second record of the file). It allows to select the printer that must be addressed. The record is optional, it must be used if more than one printer has to be managed.

- Txt Mode printer(FieldSeparator)(Num)
Es: Printer; 3
If we want to address printer N. 3 and the field separator character is semicolon.
- Xml Mode
<Printer Num="3" />

6.3 File body

The content of the receipt body depends on the kind of file has to be managed: fiscal receipt, fiscal not receipt or others.

6.3.1 Records managed regardless the kind of file.

| N | Kind of Record | keyword |
|---|-----------------------------|----------------|
| 1 | Message to customer display | displayText |
| 2 | Opening of the Cash Drawer | openDrawerText |

6.3.1.1 Text message for customer display

It sends to the customer display the text messages indicated.

- Txt Format (supposing that the separator used is the semicolon character)
displayText; Nope; DisplayTextMessage
- Xml Format
<displayText Ope="Nope" Text=" DisplayTextMessage"/>

where:

- **displayText** is the keyword for the operation
- **Nope** is Id number of the operator (range 1 ÷ 12)
- **DisplayTextMessage** represents the text that must be presented on the displayed. Max 40 characters.

6.3.1.2 Opening of the cash drawer

It sends a command to open the cash drawer

- Txt Format (supposing that the separator used is the semicolon character)
openDrawer; Nope
- Xml Format
< openDrawer Ope="Nope"/>

where:

- **openDrawer** is the keyword for the operation
- **Nope** is Id number of the operator (range 1 ÷ 12)

6.3.2 File body for FISCAL RECEIPT

One or more records of different kind. It is composed by a set of the following kind of record:

| N | Kind of Record | keyword |
|----|-------------------------------|----------------------------|
| 1 | Sale | printRecItem |
| 2 | Good Return | printRecRefund |
| 3 | Sale Cancel | printRecVoidItem |
| 4 | Discount on the Sale | printRecItemAdjustment |
| 5 | Discount on the Subtotal | printRecSubtotalAdjustment |
| 6 | Print of Subtotal | printRecSubtotal |
| 7 | Additional Row or Description | printRecMessage |
| 8 | Print of Bar Code | printBarCode |
| 9 | Payment Section | printRecTotal |
| 10 | Text for Customer Display | displayText |
| 11 | Opening of Cash Drawer | openDrawer |

6.3.2.1 Sale

It prints on the receipt the sale of an item. If it is the first sale, it starts the new fiscal receipt.

- Txt Format (supposing that the separator used is the semicolon character)
printRecItem; Nope; SaleDescription; xx,xxx; yy,yy; Ndep; Just
- Xml Format
<printRecItem Ope="Nope" Text="SaleDescription" Qty="xx,xxx" UnitCost="yyy,yy" Dep="Ndep" Just="Just"/>

where:

- **printRecItem** is the keyword for selling operations
- **Nope** is Id number of the operator (range 1 ÷ 12)
- **SaleDescription** is a description composed by max 32 characters
- **xx,xxx** represents the quantity (range 0,001 ÷ 9999,999)
- **yy,yy** represents the unit price (range 0,01 ÷ 9999999,999)
- **Ndep** represents the department number (range 1 ÷ 40)
- **Just** it is used to select the part of description (max chars)to present on the display (20 chars).
If Just = 1 it will present the first 20 characters, if just = 2 it will present the last 20 characters.

6.3.2.2 Good return from the customer

Good Return from the customer (refund required), it prints on the receipt the return of an item that must be re-found to the customer.

- Txt Format (supposing that the separator used is the semicolon character)
printRecRefund; Nope; RefundDescription; xx,xxx; yy,yy; Ndep; Just
- Xml Format
<printRecRefound Ope="Nope" Text="RefundDescription" Qty="xx,xxx" UnitCost="yyy,yy" Dep="Ndep" Just="Just"/>

where:

- **printRecRefund** is the keyword for refund operations
- **Nope** is Id number of the operator (range 1 ÷ 12)
- **RefundDescription** is a description composed by max 32 characters
- **xx,xxx** represents the quantity (range 0,001 ÷ 9999,999)
- **yy,yy** represents the unit price (range 0,01 ÷ 9999999,999)
- **Ndep** represents the department number (range 1 ÷ 40)
- **Just** it is used to select the part of description (max chars)to present on the display (20 chars).
If Just = 1 it will present the first 20 characters, if just = 2 it will present the last 20 characters.

6.3.2.3 Sale cancel

It voids one of the previous sale operations.

- Txt Format (supposing that the separator used is the semicolon character)
printRecVoidItem; Nope; VoidItemDescription; xx,xxx; yy,yy; Ndep; Just
- Xml Format
<printRecVoidItem Ope="Nope" Text=" VoidItemDescription" Qty="xx,xxx" UnitCost="yyy,yy"
Dep="Ndep" Just="Just"/>

where:

- **printRecVoidItem** is the keyword for void operations
- **Nope** is Id number of the operator (range 1 ÷ 12)
- **VoidItemDescription** is a description composed by max 32 characters
- **xx,xxx** represents the quantity (range 0,001 ÷ 9999,999)
- **yy,yy** represents the unit price (range 0,01 ÷ 9999999,999)
- **Ndep** represents the department number (range 1 ÷ 40)
- **Just** it is used to select the part of description (max chars)to present on the display (20 chars).
If Just = 1 it will present the first 20 characters, if just = 2 it will present the last 20 characters.

6.3.2.4 Discount on the sale

It applies a discount on the previous sale operation.

Txt Format (supposing that the separator used is the semicolon character)
printRecItemAdjustment; Nope; DiscountDescription; NU; yyy,yy; Ndep; Just

Xml Format

```
<printRecItemAdjustment      Ope="Nope"      Text="DiscountDescription"      Type="NU"
Amount="yyy,yy" Dep="Ndep" Just="Just"/>
```

where:

- **printRecItemAdjustment** is the keyword for the operation
- **Nope** is Id number of the operator (range 1 ÷ 12)
- **DiscountDescription** is a description composed by max 32 characters
- **NU** is an unused numeric field that can be left = 0
- **yyy,yy** represents the discount amount (range 0,01 ÷ 9999999,999)
- **Ndep** represents the department number (range 1 ÷ 40)

- **Just** it is used to select the part of description (max chars)to present on the display (20 chars). If Just = 1 it will present the first 20 characters, if just = 2 it will present the last 20 characters.

6.3.2.5 Discount on the Subtotal

It applies a discount on the subtotal of the receipt.

- Txt Format (supposing that the separator used is the semicolon character)
printRecSubtotalAdjustment;Nope; DiscountDescription;Prn;yyy,yy;Ndep;Just
- Xml Format
<printRecSubtotalAdjustment Ope="Nope" Text="DiscountDescription" Type="Prn"
Amount="yyy,yy" Dep="Ndep" Just="Just"/>

where:

- **printRecSubtotalAdjustment** is the keyword for the operation
- **Nope** is Id number of the operator (range 1 ÷ 12)
- **DiscountDescription** is a description composed by max 32 characters
- **Prn** indicates if it is required a printout of the subtotal (Prn=1) or it is not required (Prn=2)
- **yyy,yy** represents the discount amount (range 0,01 ÷ 9999999,999)
- **Ndep** represents the department number (range 1 ÷ 40)
- **Just** it is used to select the part of description (max chars)to present on the display (20 chars). If Just = 1 it will present the first 20 characters, if just = 2 it will present the last 20 characters.

6.3.2.6 Print Subtotal

This Keyword gives the possibility to print out or to show on the display the content of SUBOTATAL register.

- Txt Format (supposing that the separator used is the semicolon character)
printRecSubtotal|3|1
- Xml Format
<printRecSubtotal Ope="3" Type="1" />

where:

- **printRecSubtotal** is the keyword for the operation
- **Nope** is Id number of the operator (range 1 ÷ 12)
- **Kind of Subtotal:** 0=Print on printer & Display; 1=Just print on printer; 2=Just print on display; 3= get subtotal amount

6.3.2.7 Additional Row Description

It applies additional header, row or description on the receipt

- Txt Format (supposing that the separator used is the semicolon character)
printRecMessage; Nope; Type; Num; Font; RowText
- Xml Format
<printRecMessage Ope="Nope" Type="Type" Index="Num" Font="Font" Text="RowText"/>

where:

- **printRecMessage** is the keyword for the operation
- **Nope** is Id number of the operator (range 1 ÷ 12)
- **Type** defines the kind of row that must be added; 1=Additional header; 2= Additional row (before MF logotype); 3= Additional Promo (After MF logotype); 4=Additional Description (in the body of the receipt)
- **Num** indicates the number of the additional rows (Range 1 ÷ 9 for additional header (type 1); Range 1 ÷ 99 for additional promo and descriptions (types 2 and 3); No meaning for additional row (type 4)
- **Font** indicates the font that must be used for the row (1=Normal; 2=Bold, 3=Double height; 4=Bold Double height)
- **RowText** represents the text that must be printed. Max 40 characters. For type 4, it will be accepted up to 32 characters. Further characters will be ignored

Notes: Additional headers must be placed before the first selling operation.

6.3.2.8 Print of Bar Codes

It allows to print a Bar Code at the end of the receipt after the additional header, row or description on the receipt

- Txt Format (supposing that the separator used is the semicolon character)
printBarCode; Nope; Pos; BcWidth; BcHeight; HriEnabling; HriFont; CodeType; RowText
- Xml Format

```
<printBarCode Ope="Nope" Pos="Position" Width="BcWidth" Height="BcHeight"
Hri="HriEnabling" Font="HriFont" Tipo="CodeType" Text="BcText"/>
```

where:

- **printRecMessage** is the keyword for the operation
- **Nope** is Id number of the operator (range 1 ÷ 12)
- **Position** defines the starting position from the left margin (range 0 ÷ 511)
- **BcWidth** indicates the width of the bar-code (range 1 ÷ 8).
- **BcHeight** indicates the height of the bar-code in print dots (range 1 ÷ 255).
- **HriEnabling** enables or disables the characters string of the barcode (DISABLED, BELOW, ABOVE, TWICE)
- **HriFont** indicates the font that must be used for the Hri string (FontA, FontB, FontC)
- **CodeType** indicates the barcode system to be printed. It must be chosen among the following (UPC-A, UPC-B, EAN13, EAN8, ITF, CODABAR, CODE93, CODE128)
- **RowText** represents the barcode string to be printed

Notes:

- 1) The character set and the string length are defined by the standard related to each bar code system.
- 2) The starting position, the length of the string and the width of the barcode must be defined according to the roll paper size. If there is no room enough for the row, the barcode will not be printed.
- 3) According to the standard, using CODE128, it must be selected one of the following character set:
 - CODE A, To select CODE A, the first two characters of the RowText must be "{A"
 - CODE B, To select CODE B, the first two characters of the RowText must be "{B"
 - CODE C, To select CODE C, the first two characters of the RowText must be "{C"
 In case of CODE128, the first two characters are not part of the bar code itself so they are not printed. They just allow the selection of the character set

Important Note: BarCode Systems CODE93 and CODE128 are managed by FP90 printers since Fw release 3.00

6.3.2.9 Payment section

One or more records. Usually it is composed by one record, more than one record means that the payment is composed by several partial payments.

- Txt Format
printRecTotal; Nope; PaymentDescription; yyy,yy; Mode; Index; Just
- Xml Format
<printRecTotal Ope="Nope" Text="PaymentDescription" Amount="yyy,yy" Type="Mode"
Index="Ncard" Just="Just"/>

where:

- **printRecTotal** is the keyword for payment operations
- **Nope** is Id number of the operator (range 1 ÷ 12)
- **PaymentDescription** is a description composed by max 32 characters
- **yyy,yy** represents the amount (range 0,01 ÷ 9999999,999)
- **Mode** indicates the payment mode (0=Cash; 1=Check; 2=Credit/Credit Card; 4=Ticket)
- **Index** represents in case of Credit Cards or Ticket to select which kind of credit card or ticket has been used)
- **Just** it is used to select the part of description (max chars)to present on the display (20 chars). If Just = 1 it will present the first 20 characters, if just = 2 it will present the last 20 characters.

6.3.3 Text for Customer display

Text messages for the customer display can be sent while managing a fiscal receipt. This record is managed also in other kind of command file. Details of the command are given in the "*Records managed regardless the kind of file*" section.

6.3.4 Opening of Cash Drawer

Closure (payment completed) of fiscal receipt opens the cash drawer automatically. Anyway a command has been added to force the drawer opening. This record is managed also in other kind of command file. Details of the command are given in the "*Records managed regardless the kind of file*" section.

6.3.5 Forbidden words in text messages on the fiscal Receipt.

There are words that cannot be printed on fiscal receipt as additional text or description. The forbidden words are:

- Totale
- Sconto
- Importo
- Contante

These words can not be printed even if the characters are separated by not alphabetic characters (es: Cont-ant#e; s?c'o"n%t o,).

These word should not be present in any text message or description.

Fiscal printer must refuse any message that requires to print any of the above words on fiscal receipt.

EpsonFpMate performs a pre-processing operation on text message for fiscal receipt. In this way it avoids that the printer refuses a message returning an error code because in the text there is one or more "forbidden words".

Pre processing operation consists in the following operation:

- Check the presence of one forbidden words.
- If the word has been found, replace the last character with an "*" (Asterisk).
- Check if the word is followed by one or more of the replaced character and in the case replace all of them.

Examples:

Instead of "Totale" EpsonFpMate prints "Total*"

Instead of "Importo" EpsonFpMate prints "Import*"

Instead of "Sconto" EpsonFpMate prints "Scont*"

Instead of "Contante" EpsonFpMate prints "Contant*"

Instead of "Totale eaiou" EpsonFpMate prints "Total* *aiou"

Instead of "Sconto o o#oao" EpsonFpMate prints "Scont* * *#oao"

6.4 File body for "Nota Di Credito" FISCAL RECEIP

Since Release 3.00, FP90 allows to manage receipt related only to good refund activity. It means that the customer receive back the money he paid when he bought it.

The operation allowed in a "Nota di Credito" are almost the same than in a normal fiscal receipt and listed in the following table:

| N | Kind of Record | keyword |
|----|-------------------------------|----------------------------|
| 1 | Good Return | printRecRefund |
| 2 | Operation Cancel | printRecVoidItem |
| 3 | Discount on the Sale | printRecItemAdjustment |
| 4 | Discount on the Subtotal | printRecSubtotalAdjustment |
| 5 | Print of Subtotal | printRecSubtotal |
| 6 | Additional Row or Description | printRecMessage |
| 7 | Print of Bar Code | printBarCode |
| 8 | Payment Section | printRecTotal |
| 9 | Text for Customer Display | displayText |
| 10 | Opening of Cash Drawer | openDrawer |

For the details of the commands see in the file body for FISCAL RECEIP section.

6.4.1 Opening a "Nota Di Credito" FISCAL RECEIP

To open a "Nota di Credito" it is required to send as first command a "printRecMessage", type 4 that begins with the sentence "PRATICA DI RESO".

It is important to use capital font and separate the words by one blank character.

6.4.2 Closure of a "Nota Di Credito" FISCAL RECEIP

To open a "Nota di Credito" is closed by a payment command. It is ignored the amount indicated in the command line. The amount reported on the receipt is the one computed by the FP90 itself.

Important Note: "Nota di Credito" is managed by FP90 printers since Fw release 3.00

6.5 File body for NOT FISCAL RECEIPT

EpsonFpMate is able to manage the print of NOT fiscal receipt.
The keyword used to print out a NOT fiscal receipt is "printNonFiscalReceipt".

Following there is a list of command to be used in the NOT FISCAL file:

| N | Kind of Record | keyword |
|---|---------------------------|----------------|
| 1 | Open a NOT fiscal receipt | beginNonFiscal |
| 2 | Print a row | printNormal |
| 3 | Print of Bar Code | printBarCode |
| 4 | Close a fiscal receipt | endNonFiscal |
| 5 | Text for Customer Display | displayText |
| 6 | Opening of Cash Drawer | openDrawer |

6.5.1 Begin Non fiscal receipt

Used to start a not fiscal receipt, the format is a follow:

- Txt Format
beginNonFiscal;Nope
- Xml Format
<beginNonFiscal Ope="Nope" />

"Nope" means the operator field (range 1-12)

6.5.2 Print Bar Code

Printout of bar codes in not fiscal receipts uses the same command with the same rules described in the Fiscal Receipt section.

6.5.3 Print row

To printout row in a NOT fiscal receipt, the keyword is printNormal, the format is as follow:

- Txt Format
printNormal;Nope;Font;RowText
- Xml Format
<printNormal Ope="Nope" Font="Font" Text="RowText"/>
- "Nope" means the operator field (range 1-12)
- "Font" means the kind of font to be used: 1=normal, 2=bold, 3=double, 4 double and bold
- "RowText" means the row to be printed

6.5.4 Close Not fiscal receipt

To close a Not fiscal receipt must be used the following keyword: `endNonFiscal`, the format is as follow:

- Formato Txt
`endNonFiscal;Nope`
- Formato Xml
`<endNonFiscal Ope="Nope" />`
- "Nope" means the operator field (range 1-12)

6.5.5 Text for Customer display

Details of the command are given in the "*Records managed regardless the kind of file*" section.

6.5.6 Opening of Cash Drawer

Details of the command are given in the "*Records managed regardless the kind of file*" section.

6.6 File body for NOT FISCAL RECEIPT

EpsonFpMate is able to manage the print of NOT fiscal receipt.
The keyword used to print out a NOT fiscal receipt is "printNonFiscalReceipt".

Following there is a list of command to be used in the NOT FISCAL file:

| N | Kind of Record | keyword |
|---|---------------------------|----------------|
| 1 | Open a NOT fiscal receipt | beginNonFiscal |
| 2 | Print a row | printNormal |
| 3 | Print of Bar Code | printBarCode |
| 4 | Close a fiscal receipt | endNonFiscal |
| 5 | Text for Customer Display | displayText |
| 6 | Opening of Cash Drawer | openDrawer |

6.6.1 Begin Non fiscal receipt

Used to start a not fiscal receipt, the format is a follow:

- Txt Format
beginNonFiscal;Nope
- Xml Format
<beginNonFiscal Ope="Nope" />

"Nope" means the operator field (range 1-12)

6.6.2 Print Bar Code

Printout of bar codes in not fiscal receipts uses the same command with the same rules described in the Fiscal Receipt section.

6.6.3 Print row

To printout row in a NOT fiscal receipt, the keyword is printNormal, the format is as follow:

- Txt Format
printNormal;Nope;Font;RowText
- Xml Format
<printNormal Ope="Nope" Font="Font" Text="RowText"/>
- "Nope" means the operator field (range 1-12)
- "Font" means the kind of font to be used: 1=normal, 2=bold, 3=double, 4 double and bold
- "RowText" means the row to be printed

6.6.4 Close Not fiscal receipt

To close a Not fiscal receipt must be used the following keyword: `endNonFiscal`, the format is as follow:

- Formato Txt
`endNonFiscal;Nope`
- Formato Xml
`<endNonFiscal Ope="Nope" />`
- "Nope" means the operator field (range 1-12)

6.6.5 Text for Customer display

Details of the command are given in the "*Records managed regardless the kind of file*" section.

6.6.6 Opening of Cash Drawer

Details of the command are given in the "*Records managed regardless the kind of file*" section.

6.7 File body for COMMAND file

By means of file it is possible send specific commands to the fiscal printer.

| Type of print out | Keyword |
|---------------------------|-------------|
| Text for Customer Display | displayText |
| Opening of Cash Drawer | openDrawer |

6.7.1 Text for Customer display

Details of the command are given in the "*Records managed regardless the kind of file*" section.

6.7.2 Opening of Cash Drawer

Details of the command are given in the "*Records managed regardless the kind of file*" section.

6.8 File body for DAILY CLOSURE

By means of file it is possible to manage the daily closure with print out of the Zreport.

| Type of print out | Keyword |
|----------------------------------|-----------------|
| Normal daily closure | printZReport |
| With additional financial report | printZTotReport |
| X Daily closure (W/O reset) | printXReport |

Example for Normal daily closure

- Txt Format
printZReport;Nope
- Xml Format
<printZReport Ope="Nope" />

"Nope" means the operator field (range 1-12)

Example for daily closure with financial report

- Xml Format
<printZTotReport Ope="Nope" />
- Txt Format
printZTotReport;Nope

"Nope" means the operator field (range 1-12)

Example for daily closure with financial report

- Xml Format
<printXReport Ope="Nope" />
- Txt Format
printXReport;Nope

"Nope" means the operator field (range 1-12)

7 Output File for Fiscal Receipt

When selected the option **"CreateOutputFile"** the application generates an output file.

In case of Xml files the name of the output file is achieved putting the characters "Out" before the name of the file.

In case of Txt files the name of the output file is achieved changing the extension with the extension "out".

Output file reports:

- Date and time;
- Printer number
- Id Cash Number
- Result of the operation (ok or ko)
- In case of Result = ok
 - Receipt Number
 - Total amount of the receipt
- In case of Result = ko
 - Error Code
 - Short description of the error

8 Examples of TXT file

8.1 Example of fiscal receipt

Following a simple file in TXT format used to print out fiscal receipt is reported:

```
printerFiscalReceipt
Printer|1
printRecMessage|1|1|1|1|Prima Riga Aggiuntiva Tipo 1
printRecMessage|1|1|2|1|Seconda Riga Aggiuntiva Tipo 1
printRecMessage|1|4|1|1|Prima Riga Aggiuntiva Tipo 4
printRecItem|1|Vendita Reparto Iva 10%|1|100,0|15|1
printRecMessage|1|4|1|1|Seconda Riga Aggiuntiva Tipo 4
printRecItem|1|Vendita Reparto Iva 20%|1,234|100,0|15|1
printRecVoidItem|1|Vendita Reparto Iva 20%|1,234|100,0|15|1
printRecItem|1|Vendita Reparto Iva 4%|2,5|100,17|3|2
printRecMessage|1|4|1|1|Terza Riga Aggiuntiva Tipo 4
printRecRefund|2|Restituzione Imballo|10,000|5,0|15|1
printRecItem|3|Vendita Reparto Esente Iva|12,13|216,17|3|2
printRecItemAdjustment|3|Sconto su singolo prodotto|0|123,45|1|2
printRecMessage|1|2|1|1|Prima Riga Aggiuntiva Tipo 2
printRecMessage|1|2|2|1|Seconda Riga Aggiuntiva Tipo 2
printRecMessage|1|3|1|1|Prima Riga Aggiuntiva Tipo 3
printRecMessage|1|3|2|1|Seconda Riga Aggiuntiva Tipo 3
printRecItem|3|Vendita Reparto esente IVA|12,13|216,17|3|2
printRecSubtotalAdjustment|3|Sconto sul subtotal|1|300,12|3|2
printRecSubtotal|3|1
printBarCode|1|10|2|66|TWICE|FontA|CODE39|01234567ABCDEF
printRecTotal|4|Pagamento con carta di credito|6000,0|0|0|2
displayText|1|Customer Display Printed Fisc Receipt
```

8.2 Example of “Nota di Credito” fiscal receipt

Following a simple file in TXT format used to print out fiscal receipt for money refund is reported:

```
printerFiscalReceipt
Printer|1
printRecMessage|1|4|1|1|PRATICA DI RESO BETA 1
printRecRefund|2|Refund1 (good return)|10,000|5,0|1|1
printRecRefund|2|Refund2 (good return)|13,000|6,0|1|1
printRecRefund|2|Refund3 (good return)|16,000|7,0|1|1
printRecItemAdjustment|3|Discount applied to the product|0|1,23|1|2
printRecMessage|1|2|1|1|First Additional Row Type 2
printRecMessage|1|2|2|1|Second Additional Row Type 2
printRecMessage|1|3|1|1|First Additional Row Type 3
printRecMessage|1|3|2|1|Second Additional Row Type 3
printRecSubtotal|3|1
printBarCode|1|10|2|66|BELOW|FontA|CODE93|0123456789ABCDEFG
printRecTotal|4|Payment cash|0,0|0|0|2
displayText|1|Arrivederci e grazie Avanti il prossimo
```

8.3 Example of Z report command file

```
printerFiscalReport
Printer|1
printZReport|1
displayText|1|Messaggio su display cliente
```

8.4 Examples of file to send commands to the printer

```
printerCommand
Printer|1
openDrawer|1
displayText|1|Customer Display Message
```

```
printerCommand
Printer|1
openDrawer|1
```

```
printerCommand
Printer|1
openDrawer|1
displayText|1|Customer Display Message
```

8.5 Example of Not fiscal receipt

```
printerNonFiscal
beginNonFiscal|1|
printNormal|1|1|Non Fiscal Receipt Row N. 1 Font 1
printNormal|1|2|
printNormal|1|2|BarCode System CODE39
printBarCode|1|10|2|66|TWICE|FontA|CODE39|01234567ABCD
printNormal|1|2|
printNormal|1|2|Non Fiscal Receipt Row N. 2 Font 2
printNormal|1|3|Non Fiscal Receipt Row N. 3 Font 3
printNormal|1|2|
printNormal|1|2|BarCode System CODE93
printBarCode|1|10|2|66|TWICE|FontA|CODE93|01234567ABCDEF
printNormal|1|2|
printNormal|1|4|Non Fiscal Receipt Row N. 4 Font 4
printNormal|1|1|Non Fiscal Receipt Row N. 5 Font 1
printNormal|1|2|Non Fiscal Receipt Row N. 6 Font 2
printNormal|1|2|
printNormal|1|2|BarCode System CODE128 CODE A
printBarCode|1|10|2|66|TWICE|FontA|CODE128|{A01234567ABCDEF
printNormal|1|2|
printNormal|1|3|Non Fiscal Receipt Row N. 7 Font 3
printNormal|1|2|
printNormal|1|2|BarCode System CODE128 CODE B
printBarCode|1|10|2|66|TWICE|FontA|CODE128|{B01234567ABCDEF
printNormal|1|2|
printNormal|1|4|Non Fiscal Receipt Row N. 8 Font 4
printNormal|1|2|
printNormal|1|2|BarCode System CODE128 CODE C
```

```
printBarCode|1|10|2|66|TWICE|FontA|CODE128|{C01234567ABCDEF  
printNormal|1|2|  
displayText|4|Customer Display   Printed Not Fisc Doc  
endNonFiscal|1|
```

9 Examples of XML file

9.1 Example of fiscal receipt

```
<?xml version="1.0" encoding="utf-16" ?>
<printerFiscalReceipt>
  <Printer Num="1" />
  <displayText Ope="1" Text="Messagge on customer display" />
  <printRecMessage Ope="1" Text="First Additional Row Type 1" Type="1" Index="1" Font="1" />
  <printRecMessage Ope="1" Text="Second Additional Row Type 1" Type="1" Index="2" Font="1" />
  <beginFiscalReceipt Ope="1" />
  <printRecMessage Ope="1" Text="First Additional Row Type 4" Type="4" Index="1" Font="1" />
  <printRecItem Ope="1" Text="Selling Item 1 VAT 10%" Qty="1" UnitCost="100,0" Dep="15" Just="1" />
  <printRecMessage Ope="1" Text="Second Additional Row Type 4" Type="4" Index="1" Font="1" />
  <printRecItem Ope="2" Text="Selling Item 2 VAT 20%" Qty="1,234" UnitCost="100,0" Dep="15" Just="1" />
  <printRecVoidItem Ope="2" Text="Void selling Item2" Qty="1,234" UnitCost="100,0" Dep="15" Just="1" />
  <printRecItem Ope="4" Text="Selling Item 3 VAT 20%" Qty="2,5" UnitCost="100,17" Dep="3" Just="2" />
  <printRecMessage Ope="1" Text="Third Additional Row Type 4" Type="4" Index="1" Font="1" />
  <printRecRefund Ope="2" Text="Refound (good return)" Qty="10,000" UnitCost="5,0" Dep="15" Just="1" />
  <printRecItem Ope="3" Text="Selling Item 4 VAT 20%" Qty="12,13" UnitCost="216,17" Dep="3" Just="2" />
  <printRecItemAdjustment Ope="3" Text="Discount applied to the product" Type="1" Amount="123,45" Dep="3" Just="2" />
  <printRecMessage Ope="1" Text="First Additional Row Type 2" Type="2" Index="1" Font="1" />
  <printRecMessage Ope="1" Text="Second Additional Row Type 2" Type="2" Index="2" Font="1" />
  <printRecMessage Ope="1" Text="First Additional Row Type 3" Type="3" Index="1" Font="1" />
  <printRecMessage Ope="1" Text="Second Additional Row Type 3" Type="3" Index="2" Font="1" />
  <printRecItem Ope="3" Text="Selling Item 5 20%" Qty="12,13" UnitCost="216,17" Dep="3" Just="2" />
  <printRecSubtotalAdjustment Ope="3" Text="Discount applied to the subtotal" Type="1" Amount="300,12" Dep="3" Just="2" />
  <printRecSubtotal Ope="3" Type="1" />
  <printBarCode Ope="1" Position="10" Width="2" Height="66" Hri="TWICE" Font="FontA" Tipo="CODE39" Text="0123456789" />
  <printRecTotal Ope="4" Text="Payment cash" Amount="6000,0" Type="0" Index="0" Just="2" />
  <displayText Ope="1" Text="Message on customer display" />
</printerFiscalReceipt>
```

9.2 Example of "Nota di Credito" fiscal receipt

```
<?xml version="1.0" encoding="utf-16" ?>
<printerFiscalReceipt>
  <Printer Num="1" />
  <printRecMessage Ope="1" Text="PRATICA DI RESO BETA 1" Type="4" Index="1" Font="1" />
  <printRecRefund Ope="2" Text="Refund1 (good return)" Qty="10,000" UnitCost="5,0" Dep="1" Just="1" />
  <printRecRefund Ope="2" Text="Refund2 (good return)" Qty="13,000" UnitCost="6,0" Dep="2" Just="1" />
  <printRecRefund Ope="2" Text="Refund3 (good return)" Qty="16,000" UnitCost="7,0" Dep="3" Just="1" />
  <printRecItemAdjustment Ope="2" Text="Discount applied to the product" Type="1" Amount="1,23" Dep="3" Just="2" />
  <printRecMessage Ope="1" Text="First Additional Row Type 2" Type="2" Index="1" Font="1" />
  <printRecMessage Ope="1" Text="Second Additional Row Type 2" Type="2" Index="2" Font="1" />
  <printRecMessage Ope="1" Text="First Additional Row Type 3" Type="3" Index="1" Font="1" />
  <printRecMessage Ope="1" Text="Second Additional Row Type 3" Type="3" Index="2" Font="1" />
  <printRecSubtotal Ope="3" Type="1" />
  <printBarCode Ope="1" Pos="10" Width="2" Height="66" Hri="BELOW" Font="FontA" Tipo="CODE93" Text="0123456789ABCDEFGH" />
  <printRecTotal Ope="4" Text="Payment cash" Amount="0,0" Type="0" Index="0" Just="2" />
  <displayText Ope="4" Text="Arrivederci e grazie Avanti il prossimo" />
</printerFiscalReceipt>
```

9.3 Example of Zreport file command

```
<?xml version="1.0" encoding="utf-16" ?>
<printerFiscalReport>
  <Printer Num="1" />
  <printZReport Ope="1" />
  <displayText Ope="1" Text=" Messagge on      customer display " />
</printerFiscalReport>
```

9.4 Examples of file to send commands to the printer

```
<?xml version="1.0" encoding="utf-16" ?>
<printerCommand>
  <Printer Num="1" />
  <openDrawer Ope="1" />
</printerCommand>
```

```
<?xml version="1.0" encoding="utf-16" ?>
<printerCommand>
  <Printer Num="1" />
  <displayText Ope="4" Text="Customer Display Message" />
</printerCommand>
```

```
<?xml version="1.0" encoding="utf-16" ?>
<printerCommand>
  <Printer Num="1" />
  <openDrawer Ope="1" />
  <displayText Ope="4" Text="Customer Display Message" />
</printerCommand>
```

9.5 Example of not fiscal receipt

```
<?xml version="1.0" encoding="utf-16" ?>
<printerNonFiscal>
  <Printer Num="1" />
  <displayText Ope="1" Text="Messaggio su   display cliente" />
  <beginNonFiscal Ope="1" />
  <printNormal Ope="1" Font="1" Text="Non Fiscal Receipt Row N. 1 Font 1"
  <printNormal Ope="1" Font="2" Text="Non Fiscal Receipt Row N. 2 Font 2"
  <printNormal Ope="1" Font="3" Text="Non Fiscal Receipt Row N. 3 Font 3"
  <printBarCode Ope="1" Position="10" Width="2" Height="66" Hri="ABOVE" Font="FontA" Tipo="CODE39"
    Text="0123456789" />
  <printNormal Ope="1" Font="4" Text="Non Fiscal Receipt Row N. 4 Font 4"
  <displayText Ope="1" Text="Messaggio su   display cliente" />
  <printNormal Ope="1" Font="1" Text="Non Fiscal Receipt Row N. 5 Font 1"
  <printNormal Ope="1" Font="2" Text="Non Fiscal Receipt Row N. 6 Font 2"
  <printNormal Ope="1" Font="3" Text="Non Fiscal Receipt Row N. 7 Font 3"
  <printBarCode Ope="1" Position="10" Width="2" Height="66" Hri="BELOW" Font="FontB" Tipo="CODE128"
    Text="{A0123456789}" />
  <printNormal Ope="1" Font="4" Text="Non Fiscal Receipt Row N. 8 Font 4"
  <endNonFiscal Ope="1" />
  <displayText Ope="1" Text=" Messagge on      customer display " />
</printerNonFiscal>
```

9.6 Example of Exit application file command

```
<?xml version="1.0" encoding="utf-16" ?>
<applicationExit />
```