EPSON

EXCEED YOUR VISION

ePOS Fiscal Print Solution

Development Guide

REVISION HISTORY

A	All	Preliminary.	27th April 2012	P. Barnett
B	All	Revised the complete document.	2nd May 2012	P. Barnett
C	All	Converted to DOCX and removed false titles.	2nd May 2012	C. Radaelli
D	All	Small correction.	18th May 2012	P. Barnett
E	All	Removed personalised receipt, class I and Class III references. Removed FP-H6000 slip references (beginInsertion etc.). Removed Falcon and FP-90 series one fiscal printer references. Corrected errors in example files. Added QRCode attributes to printBarCode sub-element.	7th June 2012	P. Barnett
F	All	Added XML responses for each file type Added fiscalReceiptNumber and fiscalReceiptAmount response elements	9th July 2012	C. Radaelli
G	All	Added Direct IO response. Added incomplete file response. Improved document readability. Corrected response errors. printFiscalDocAmount sub-element removed (not currently supported.	23rd July 2012	P. Barnett
H	16	addDevice and readDevices naming altered	24th Jan 2013	P . Barnett

I	All	Important printBarCode notes added. Credit note printRecTotal amount attribute note regarding firmware < 3.007 added. Corrected displayText order in example XML files. Added HTML code table in UTF-8 Encoding chapter. Added "Success, Code and Status Cross Reference Table" section. Added the following sub-elements: printContentByDate printContentByNumbers queryContent replaced by two new sub-elements: queryContentByDate and queryContentByNumbers Rewritten printGraphicCoupon. Added setLogo. Corrected basic end point addresses. Query string default timeout reduced from 60 to 10 seconds unitPrice, amount and payment zero note added. Credit or credit card indexes corrected (should be 0 – 10). Added CODABAR start and stop characters note. Minor improvements made throughout the document.	30th July 2013	P. Barnett

J	Various	Enhanced chapter 5.8.4 printRecItemVoid description. Fiscal receipt sample XML storno alias printRecVoidItem now printRecItemVoid. Updated Aliases for Backward Compatibility. Added operator note (attribute can be omitted). printDuplicateReceipt moved from printerFiscalReceipt to printerCommand XML file type. Added 5, 6 and 7 messageType attribute values to printRecMessage sub-element plus example lines added to sample files.	11th Oct 2013	P. Barnett
K	Various	Corrected sample XML for invoice based on last receipt and Direct IO. Updated Direct invoice sample XML. Added cross-domain note. Added barcode number aliases.	16th Dec 2013	P. Barnett
L	Various	Corrected Credit Note attributes: "amount" should be "payment" and "description" should be "message".	19th May 2014	P. Barnett

M	Various	"UTF-8 Encoding" section renamed to "UTF-8 Encoding and Character Escaping". Character escaping table expanded. Free and Last Receipt invoices now supported from firmware version 3.00D. Inserted Box office ticket printing sections and examples. Added timeout argument to epos.send method. Added res_add JavaScript library argument. Added fiscalReceiptDate and fiscalReceiptTime response lines. Updated queryContentByDate and queryContentByNumbers sections. Updated Command Response File section. resetPrinter description expanded. Added saveDevices and readVersion. Updated addDevice and readDevices. Added qRCodeDataType attribute. Added printRecMessage in non-fiscal receipts. Added messageType 8 in printRecMessage. Added clearEFTPOSBuffer printRecMessage attribute. Added authorizeSales, setDate and getDate printerCommand sub-elements. Added receiptBasedFreeInvoice invoice type. printRecVoid now in printerCommand files. Added INVALID XML COMMAND error message. Document revised throughout.	18th Nov 2014	P. Barnett
N	30	Added FP-81 II S and T models. clearEFTPOSBuffer requires firmware >= 4.012. Document title renamed to "ePOS Fiscal Print Solution Development Guide". Filename altered accordingly.	9th Jan 2015	P. Barnett
O	Various	"Rounding" changed to "rounding down" since extra decimal places are truncated. Added "printerStatus and fpStatus" section Corrected first letter upper case errors on some attributes throughout the document. Removed operator attribute from the following methods as was never required: setDate getDate beginTraining endTraining setLogo (in printerCommand file) Mentioned 50+ operator offset if quantity line printing when equal to 1 is desired. Removed saveDevices as no longer necessary.	22nd Dec 2015	P. Barnett

P	Various	queryContentByDate and queryContentByNumbers line limits updated. setDate year updated. Can be either 2 or 4 digits long. query string timeout field updated. directIO timeout attribute added. printerCommands file type added. EFT POS ERROR message added to error table. authorizeSales response added. Automatic Programming chapter added. Added FP NO ANSWER NETWORK error.	20th Dec 2016	P. Barnett
Q	Various	Added RT status request (new attribute). Added commercial refund document and commercial void document. Added RT examples. Document updated throughout.	13th Mar 2018	P. Barnett

Index

1 Overview

1.1 Description

The aim of this manual is to provide information to software designers who want to interface Epson fiscal and non-fiscal TM printers (latterly via a fiscal printer) using the fiscal ePOS-Print solution. It covers both "MF" and "RT" printer variants.

1.2 List of Related Documents

- Epson FP Manuale Operatore
- Active X / OCX user manual
- Communication Protocol
- Protocollo di Comunicazione NDA Biglietteria
- Intelligent Printer Printing System ePOS-Print Technical Data
- Gestione Fatture Ricevute Fiscali
- FP Gestione Logo
- Fiscal Printer Intelligent Features Guide

1.3 Abbreviations

- DGFE: Italian acronym for EJ (MF printers only)

- EJ: Electronic Journal
  (in Italian known as either the Dispositivo Giornale di Fondo Elettronico (DGFE) or Memoria Permanente di Dettaglio (MPD))

- FP: Fiscal Printer
  (in Italian known as either a Misuratore Fiscale (MF) or Registratore Telematico (RT))

- MF: Misuratore Fiscale or Memoria Fiscale

- MPD: Italian acronym for EJ (MF printers only)

- MPR: Italina acronym for fiscal memory (RT printers only)

- RT: Registratore Telematico (Data transfer fiscal printer)

1.4 Structure

The Fiscal ePOS-Print solution consists of three components:

- EpsonFPMate service
- Fiscal-ePOS-Print XML
- Fiscal ePOS-Print API

1.4.1 EpsonFPMate Service

The service is essentially an XML parser which resides inside intelligent or RT fiscal printers that reads and interprets fiscal ePOS-Print XML for printing, reporting, configuration and reading.

1.4.2 Fiscal ePOS-Print XML

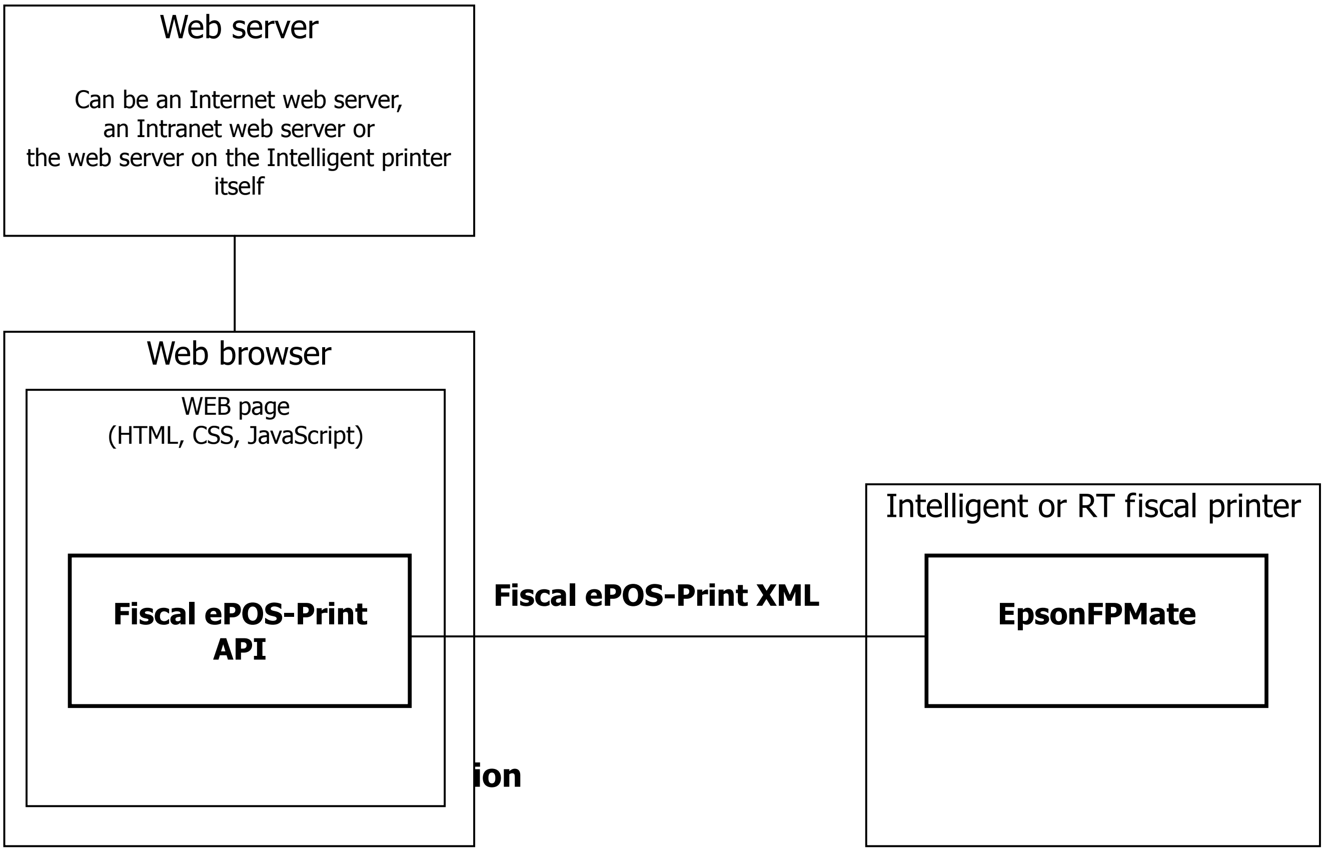
This encompasses the XML command system that defines the main fiscal printer functions in XML. All XML elements and attributes are described in this document.

1.4.3 Fiscal ePOS-Print API

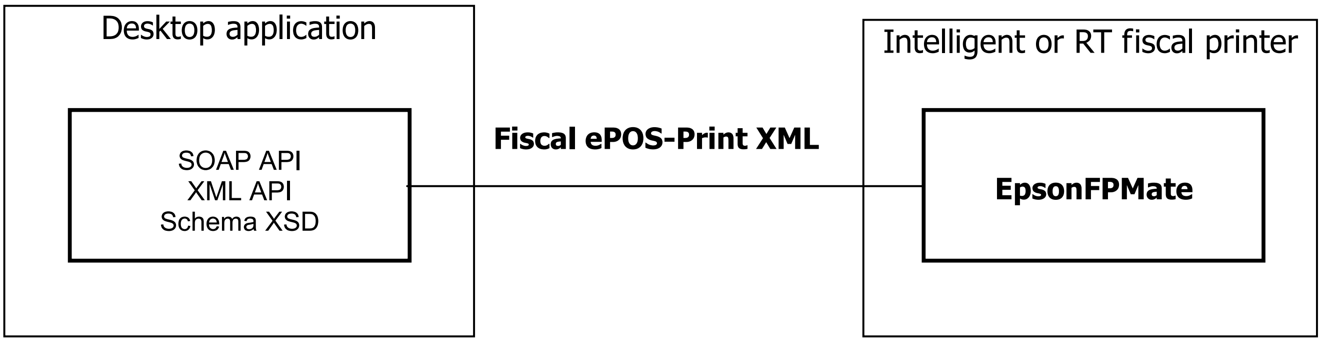
The API is a basic application programming interface that forwards requests to the fiscal ePOS-Print service through the use of client-side JavaScript code.

1.5 Typical Topologies

1.5.1 Web Application


Web server
Can be an Internet web server,
an Intranet web server or
the web server on the Intelligent printer itself
Web browser
WEB page
(HTML, CSS, JavaScript)
Fiscal ePOS-Print API
Fiscal ePOS-Print XML
Intelligent or RT fiscal printer
EpsonFPMate
ion

1.5.2 Desktop Application



2 EpsonFPMate Service

2.1 Features

The EpsonFPMate service is an XML web service incorporated into the Intelligent or RT fiscal printer that uses fiscal ePOS-Print XML for printing, reporting, configuration and reading. When a document request is sent from a host to the FPMate service via SOAP/HTTP, FPMate processes the document and afterwards sends back a response document. Printing is possible from various devices and operating systems that support communication via SOAP/HTTP. The key features and benefits are:

- Installation of drivers and plug-ins is not required
- No PCs or servers are required for printing
- Adds printing functionality to public and private clouds
- Uses TM printer built-in fonts
- Manages intelligent or RT printers
- Manages TM printers via intelligent or RT printers
- Supports Server Direct Print and Remote Request Job functions

2.2 Specifications

2.2.2 The basic end point address

The URL format is as follows:

http://[host]/cgi-bin/fpmate.cgi

When the intelligent or RT fiscal printer usage requires a timeout value, a query string is appended as follows:

http://[host]/cgi-bin/fpmate.cgi?timeout=[timeout]

When the intelligent or RT fiscal printer forwards and manages the request destined for another intelligent or RT printer or a non-fiscal TM printer, a query string is appended as follows:

http://[host]/cgi-bin/fpmate.cgi?devid=[device ID]

The two can be combined in the following manner:

http://[host]/cgi-bin/fpmate.cgi?devid=[device ID]&timeout=[timeout]

[host]
Specifies the IP address or domain name of the intelligent fiscal printer where the FPMate service resides.

[device ID]
Destination printer selection by name. The device ID of the fiscal intelligent, RT or non-fiscal TM printer is specified here. The specific printer that the XML is ultimately destined for is denoted by its DeviceID which is passed in the query string that forms part of the URL. A null or "local_printer" DeviceID indicates the use of the same printer as where the web server resides, i.e. the same as in the host portion of the URL. ePOS-print requests destined for non-fiscal TM printers must contain a DeviceID. Please see the EpsonFPMate Configuration section regarding the configuration and programming of DeviceIDs.

[timeout]
The optional timeout field is used with the authorizeSales EFTPOS payment command and secondary printer management. From version 1.022 onwards, it can also be used with directIO calls. It specifies the timeout period in milliseconds for authorizeSales and directIO elements in an XML file. When this parameter is not specified, the timeout is 30 seconds for authorizeSales and six seconds for directIO. If the directIO timeout attribute is present it overrides the query string timeout setting. When the specified timeout period elapses, the operation is cancelled. However, any XML lines beforehand would have been processed normally.

Examples

http://192.168.192.168/cgi-bin/fpmate.cgi?devid=local_printer&timeout=12000

An example URL for sending a request for onward printing is as follows:

http//192.168.1.1/cgi-bin/fpmate.cgi?devid=cucina

If the device does not exist an error is returned. Bad field names are ignored (e.g. DEVID/devID/DEV££ID/TIMEOUT). Timeout values with non-numeric characters are also ignored.

2.2.3 Request message format

A SOAP message forms part of the HTTP POST request method. A single document request is specified in the SOAP body.

```
[Example]
POST /cgi-bin/fpmate.cgi?devid=local_printer&timeout=10000 HTTP/1.1
Content-Type: text/xml; charset=utf-8
Content-Length: xxxx
If-Modified-Since: Thu, 01 Jan 1970 00:00:00 GMT

<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Body>
            <printerCommand>
                    <queryPrinterStatus  operator="" statusType=""  />
            </printerCommand>
    </s:Body>
</s:Envelope>
```

2.2.4 Response message format

A SOAP message is returned as the response to the HTTP POST method. The SOAP body contains the response document. To permit cross-domain communication, the response contains the HTTP "Access-Control-Allow-Origin: *" header. See *Fiscal Printer Intelligent Features Guide* for browser cross-domain information.

```
[Example]
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: text/xml; charset=utf-8
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
    <s:Body>
            <response success="true" code="" status="xxxxx" />
    </s:Body>
</s:Envelope>
```

3 Fiscal ePOS-Print XML

The fiscal ePOS-Print XML is a printer command system that defines the main fiscal functions in XML.

This XML is intended to facilitate the control of printers, independent of hardware or operating systems, from personal computers and POS terminals, as well as from network devices such as web application servers, smart phones and tablet computers.

The XML structure must contain one of the following root elements as it identifies the type of request. Root elements (also known as root nodes) have empty tags with no attributes or data. Currently none of the XML in the entire library contains any data except for addInfo (additional information in response messages), printGraphicCoupon and setLogo (where base64 binary data is inserted); everything else is passed as tags and attributes.

The last line must contain the root element closure statement. For example:

</printerFiscalReceipt>.

Root Element Tag Name	Description
printerFiscalReceipt	Fiscal receipt or commercial document
printerFiscalReport	Daily closure (X and Z reports)
printerNonFiscal	Non-fiscal receipt or administrative document
printerFiscalDocument	Fiscal document (invoice)
printerTicket	Box office ticket (known as a titolo d'accesso)
printerCommand	Command (e.g. printer status)
printerCommands	Multiple requests and programming
epos-print	Print to non-fiscal TM printer (see related documentation)
fpMateConfiguration	Configure or read secondary devices table
eposDevice	For future use

3.1 XML Hierarchy

There are two types of fiscal ePOS-Print XML documents:

- Request documents
- Response documents

The documents are subdivided into two main categories:

- Multi command: Fiscal receipt, Non-fiscal receipt, Fiscal document, Printer Commands and ePOS-Print to non-fiscal TM printer.
- Single command: Report, Printer Command and FPMate configuration.

A printer receives a message from say a host that contains a document and after having processed the document, sends back a message (to the same host) that contains a response document.

The tables below illustrate the main fiscal root elements and their associated sub-elements. Some of the sub-elements can be used with more than one of these root elements such as directIO and openDrawer. Most of the fiscal receipt sub-elements can also be used with the "direct" type of fiscal document that will be explained later.

Fiscal Receipt or Commercial Document	**Non-fiscal Receipt or Administrative Document**	**Report**
printerFiscalReceipt	**printerNonFiscal**	**printerFiscalReport**
beginFiscalReceipt	beginNonFiscal	printXReport
printRecItem	printBarCode	printZReport
printRecItemVoid	printGraphicCoupon	printXZReport
printRecItemAdjustment	printNormal	
printRecItemAdjustmentVoid	printRecMessage	
printRecMessage	setLogo	
printRecRefund *	endNonFiscal	
printRecRefundVoid *		
printRecSubtotal		
printRecSubtotalAdjustment		
printRecSubtotalAdjustVoid		
printRecTotal		
printGraphicCoupon		
printBarCode		
endFiscalReceipt		

* Refunds are converted to corrections (storni) on RT commercial documents.

Command	Commands	Fiscal Document	Box Office Tickets
printerCommand	**printerCommands**	**printerFiscalDocument**	**printerTicket**
authorizeSales	beginTraining	beginFiscalDocument	enterTicket
beginTraining	endTraining	printFiscalDocumentLine	beginTicket
endTraining	printContentByDate	All the printerFiscalReceipt commands apart from printBarCode and printGraphicCoupon	printTicket
getDate	printContentByNumbers	endFiscalReceipt / endFiscalDocument	printTicketVoid
printContentByDate	printDuplicateReceipt		printBarCode
printContentByNumbers	printRecCash		endTicket
printDuplicateReceipt	printRecVoid		exitTicket
printRecCash	queryPrinterStatus		
printRecVoid	resetPrinter		
queryContentByDate	setDate		
queryContentByNumbers	setLogo		
queryPrinterStatus			
resetPrinter			
setDate			
setLogo			

The table below illustrates the common commands that can be used in multi-command documents or as the single command in a printerCommand document:

openDrawer
directIO
clearText
displayText

The table below illustrates the configuration, ePos-print and Response root elements with their associated sub-elements.

Configuration	ePOS Print (TM printer)	Response
fpMateConfiguration	**epos-print**	**response**
addDevice	text	addInfo
readDevices	feed	
readVersion	Refer to ePOS document for complete command set	
	barcode	
	cut	

3.2 Fiscal Receipt / Commercial Document Structure

Fiscal receipts / commercial documents must contain the following structure:

```
<printerFiscalReceipt>
        <beginFiscalReceipt  operator="" />
                Sale(s)
                Payment(s)
        <endFiscalReceipt  operator="" />
</printerFiscalReceipt>
```

Responses are sent by EpsonFPMate after “endFiscalReceipt” has been processed. Specific fiscal receipt / commercial document responses are described in the chapter entitled **5.4.1 Fiscal Receipt / Commercial Document Response File.**

3.3 Non-fiscal Receipt / Administrative Document Structure

Non-fiscal receipts / administrative documents must contain the following structure:

```
<printerNonFiscal>
        <beginNonFiscal  operator="" />
                Commands such as for printing text , barcodes, logos or coupons
        <endNonFiscal  operator="" />
</printerNonFiscal>
```

Responses are sent by EpsonFPMate after “endNonFiscal” has been processed. Specific non-fiscal receipt / administrative document responses are described in the chapter entitled **5.8.1 Non-fiscal Receipt / Administrative Document Response File**

3.4 Report Structure

This structure is used to print out reports (Z report, X report etc.). Only one of the sub-elements should be present here since EpsonFPMate manages just one report command at a time.

```
<printerFiscalReport>
        <printZReport  operator="" timeout="" />
</printerFiscalReport>
```

Responses arrive after the reports have been printed and if relevant whenever the RT data transmission has been completed. Specific report responses are described in the chapter entitled **5.9.1 Report Response File**

3.5 Command Structure

The key difference here is that the structure must only contain one command. For example:

```
<printerCommand>
        <getDate />
</printerCommand>
```

Responses arrive immediately or after printing. Additional information is not always present since it depends on the command. Please see the particular command.

3.6 Commands Structure

This structure is used if you wish to send multiple command type elements, For example, multiple Direct IO requests:

```
<printerCommand>
        <directIO  command="4015" data="09001" />
        <directIO  command="4015" data="10002" />
</printerCommand>
```

Responses are sent by EpsonFPMate after the last command has been processed and indicates the printer status.

3.7 Fiscal Document Structure

The structure depends on the type of document. Please see the document types table (chapter **5.9.1**

Document Types). The following is required:

```
<printerFiscalDocument>
        <beginFiscalDocument  operator="" documentType="" … />
        -----
        -----
Either  <endFiscalDocument  operator="" operationType="" /> * or
        <endFiscalReceipt  operator="" /> ** or
        <endFiscalDocument  operator="" /> ** or
        Nothing in the case of invoices based on the last fiscal receipt
</printerFiscalDocument>
```

* endFiscalDocument together with operationType is used with free invoices which are supported from firmware version 3.00D and Receipt based free invoices from firmware version 4.001/4.01 (MF printer variants only).

** endFiscalDocument or endFiscalReceipt without operationType are used with direct invoices.

Responses are sent by EpsonFPMate after either "endFiscalDocument" or "endFiscalReceipt" has been processed or immediately in the case of invoices based on the last fiscal receipt. Specific fiscal document responses are described in chapter Error! Reference source not found. Error! Reference source not found.

3.8 Box Office Ticket Structure

This structure is used to print out "Titoli d'Accesso" (box office tickets) in Italy. The structure must be as follows:

```
<printerTicket>
        <enterTicket  randomNum="" controlCode="" />
                <beginTicket  randomNum="" controlCode="" />
                <printTicket  randomNum="" controlCode="" textType="R" font="1" text="TEST" />
                <printTicket  randomNum="" controlCode="" textType="R" font="2" text="TEST" />
                -----
                -----
                <printTicket  randomNum="" controlCode="" textType="S" font="4" text="SEAL" />
                <printBarCode  operator="1" position="0" width="2" height="120" hRIPosition="1"
                hRIFont="A" codeType="CODE128" code="{BAFB7F9A37AD2CD64" />
                <endTicket  randomNum="" controlCode="" />
        <exitTicket  randomNum="" controlCode="" />
</printerTicket>
```

Tickets can also be cancelled:

```
<printerTicket>
        <printTicketVoid  randomNum="" controlCode="" />
</printerTicket>
```

Responses are sent by EpsonFPMate after the last command has been processed. Specific box office ticket responses are described in the chapter entitled **5.10.1**

Ticket Response File.

3.9 Configuration Structure

This structure is used to configure EpsonFPMate so that it can manage secondary devices (the so-called Intelligent or RT Fiscal Printer Hub Function). There is no limit to the number of devices that can be added. It is also used to determine the EpsonFpMate embedded version.

```
<fpMateConfiguration>
    <addDevice  deviceId="Cucina" deviceIp="192.168.1.44" deviceType="non-fiscal"
    deviceModel="TM-T70" deviceRetry="10000" />
</fpMateConfiguration>

<fpMateConfiguration>
    <readVersion />
</fpMateConfiguration>
```

3.10 ePOS-Print Structure (for TM printer)

The structure follows the same standard as for the non-fiscal intelligent printer. Additional information is not presented (see the ePOS-Print_eng.pdf document).

3.11 Response Structure

Responses always contain the following default structure:

```
<response success="true" code="" status="2">
        <addInfo>
                <elementList>lastCommand</elementList>
                <lastCommand>YY</lastCommand>
        </addInfo>
</response>
```

Further information is reflected in additional elements whose names are also appended to the comma delimited elementList data. This varies from command to command.

3.12 Malformed Structures

Failure to adhere to the structure can have unforeseen results. Common errors are:

- <endFiscalReceipt /> missing – The fiscal receipt or commercial document remains open and an "INCOMPLETE FILE" response will be returned to the browser.
- Insufficient or no payment in fiscal receipt or commercial document – endFiscalReceipt will return an error.
- </printerFiscalReceipt> missing – A parser error is immediately returned.
- Bad attribute value – The printer processes and prints up until the sub-element with the erroneous line after which a printer error is returned.

3.13 Printer Anomalies

To counter any printer anomalies, the EpsonFPMate CGI automatically performs a printer reset every time a fiscal/non fiscal/report/document request arrives. This means for example that if a previous fiscal receipt or commercial document was open at the time the request arrives it will be cancelled so that a new one can be printed.

4 Fiscal ePOS-Print XML Details (Configuration)

4.1 FpMateConfiguration

This element is used to determine the EpsonFpMate embedded version or to configure and read secondary devices. The latter is only necessary when the "Hub" function is employed which allows the "master" printer to control secondary fiscal/non-fiscal TM printers.

By virtue of networking, EpsonFpMate can manage as many printers as you like both fiscal and non-fiscal. Non-fiscal TM printers and previous generation fiscal printers are not sent requests directly; an intelligent or RT fiscal printer acts as a proxy. If the fiscal printer where EpsonFPMate resides manages other devices, the following must be configured for each secondary device prior to use:

- Device ID.
- IP Address (the TCP port is fixed at 9100).
- Printer type (fiscal or non-fiscal).
- Printer model.
- Retry.

The information is saved internally in the intelligent or RT fiscal printer. This is not required if the fiscal printer operates "stand-alone". Non-fiscal TM printer requests destined for a fiscal device and vice-versa are refused and an appropriate error is returned. To setup the intelligent or RT fiscal printer to manage secondary printers, an appropriate one-time configuration write request needs to be sent. This overwrites any configuration previously performed so if these details need to be retrieved, a configuration read request must be performed beforehand. The configuration is not retained in the case of a printer restart.

4.1.1 Root Elements

The root elements for configuration XML file types are:

- fpMateConfiguration
- Response

4.1.2 Sub-elements

4.1.2.1 addDevice

When writing the configuration file, first the addDevice sub-element is used for each distinct device and it can be repeated as many times as you like. It has the following attributes:

N	Attribute	Description
1	deviceId	Device identifier – Can be anything you like, but avoid operating system environment variable names such as HOME or PATH.
2	deviceIp	IP address of the device in numerical not DNS format.
3	deviceType	TmPrinter or FpPrinter for example.
4	deviceModel	FP-90 or TM-T70 for example.
5	deviceRetry	Specifies the retry interval towards the secondary printer, in milliseconds.

Responses contain the following structure with no additional information:

```
<response success="true" code="" status="xxxxxx">
</response>
```

4.1.2.2 readDevices

Read requests always have the following structure with no attributes:

```
<fpMateConfiguration>
        <readDevices />
</fpMateConfiguration>
```

Responses are presented as additional information in the following format:

```
    <elementList>deviceCount</elementList>
    <deviceCount>xx
        <deviceNumber1  deviceId="Cucina" deviceIp="192.168.1.44" deviceType="TmPrinter"
        deviceModel="TM-T70" deviceRetry="10000" />
        <deviceNumber2  deviceId="Servizio" deviceIp="192.168.1.45" deviceType="FpPrinter"
        deviceModel="FP-90" deviceRetry="10000" />
        And so on
     </deviceCount>
```

It is also possible to view the stored file called fpMateConfiguration.xml directly from your browser:

- http://xxx.xxx.xxx.xxx/cgi-bin/fpMateConfiguration.xml

4.1.2.3 readVersion

This sub-element is used to determine the EpsonFpMate embedded version. It has the following structure with no attributes:

```
<fpMateConfiguration>
        <readVersion />
</fpMateConfiguration>
```

The response is presented as additional information in the following format:

```
        <elementList>fpmateVersion</elementList>
                <fpmateVersion></fpmateVersion>
```

4.1.3 Adding Devices Example

The following example sets up two secondary printers:

```
<fpMateConfiguration>
    <addDevice  deviceId="Cucina" deviceIp="192.168.1.44" deviceType="TmPrinter"
    deviceModel="TM-T70" deviceRetry="10000" />
    <addDevice  deviceId="Servizio" deviceIp="192.168.1.45" deviceType="FpPrinter"
    deviceModel="FP-90" deviceRetry="10000" />
</fpMateConfiguration>
```

5 Fiscal ePOS-Print XML Details (Printer Documents and Commands)

5.1 Description

In this section, root elements are described. The root element identifies the type of file. This allows for the control of any kind of print out and the use of various commands.

First, the common attributes and sub-elements are explained. Then every root element, its sub-elements and attributes will be described in detail.

5.2 Common Attributes

Many of the attributes passed to fiscal printers are common to many sub-elements. The following table illustrates them:

N	Attribute	Description
1	operator	Operator ID number (range 1 - 12). The operator attribute can be omitted. In this case, Operator 01 is assumed. A +50 offset can used with printRecItem, printRecRefund and printRecItemVoid (regarding printRecItemVoid, the offset cannot be used if voidLastItem="1"). Normally if the quantity = 1, the quantity line with "1x" is not printed. Adding the offset to the operator attribute instructs the fiscal printer to print the line when the quantity is exactly 1. The offset has no effect with direct invoices.
2	description	String whose maximum length varies depending on the printer model. Currently the maximum is: Fiscal receipts and commercial documents – 38 characters Direct invoices – 37 characters When printing invoices based on the last fiscal receipt, any 38 character descriptions in the fiscal receipt are truncated to 37 characters. Descriptions that exceed limits are truncated. Note that other attribute names with a similar meaning can have higher maximum lengths such as the "data" attribute used with the "printNormal" sub-element.
3	department	Department ID number (range 1 - 99).
4	quantity	Epson fiscal printers can compute quantities from 0.001 up to 9999.999. EpsonFPMate automatically rounds down quantities with more than three decimal places. If the quantity exceeds 9999.999, an error is returned. Either a comma or a full stop (period) can represent the decimal point. Thousand separators should not be used. The value must not be zero.
5	unitPrice amount payment	Epson fiscal printers can accept prices from 0.00 up to 9999999.99. EpsonFPMate automatically rounds down amounts with more than two decimal places. If it exceeds 9999999.99, an error is returned. Either a comma or a full stop (period) can represent the decimal point. Thousand separators should not be used. The unitPrice and payment attributes can be zero but the amount attribute cannot be zero.
6	justification	Set the customer display description justification. The maximum number of characters that can appear on the display on one line is 20, but descriptions can be longer. This parameter tells the fiscal printer whether to take the first 20 characters or the last 20 from the relative description text to use on the display. Set the attribute as follows: 1 = First 20 characters 2 = Last 20 characters

N	Attribute	Description
7	font	Fiscal printers support four different font types. Set the attribute as follows: 1 = Normal 2 = Bold 3 = Double height 4 = Bold and double height The font attribute used with barcodes has a different meaning. Please see the ePOS-Print_eng document for an explanation of non-fiscal TM printer fonts.
8	comment	Ignored by the printer. Useful especially for directIO requests.

5.3 Common Sub-elements

The following table illustrates the common fiscal printer sub-elements that can be used irrespective of the fiscal printer root element (request type). However, they cannot be used with ePOS-print non-fiscal TM printer requests.

N	Sub-element	Description
1	openDrawer	Open the cash drawer.
2	directIO	Send PDU strings (allows full use of all the intelligent or RT printer functionality).
3	clearText	Blank the display.
4	displayText	Send messages to the display.

5.3.1 openDrawer

Opens the cash drawer. This sub-element has only one attribute to indicate the operator.

```
<openDrawer  operator="" />
```

5.3.2 directIO

Sends PDU native communication protocol strings to the fiscal printer. directIO gives the added benefit of being able to send any one of the full set of command strings to the printer so you can make full use of all the fiscal printer functionality. Two attributes are passed; command and data. From version 1.022 onwards, the "timeout" attribute can also be passed. This overrides an eventual query string timeout. Please refer to the "Communication Protocol" document for the command reference. The command attribute is a concatenation of H1 and H2 and is always four bytes long. The data attribute contains the remaining fields and its length varies depending on the command. The timeout is expressed in milliseconds. Care must be taken that the protocol is followed to the letter. None of the corrective mechanisms such as rounding and truncating operate here. The following example opens the cash drawer, defines operator 01 and sets a five second timeout.

```
<directIO  command="1050" data="01" timeout="5000" comment="Open cash drawer, operator 1 and five second timeout" />
```

DirectIO responses contain the following additional information:

- lastCommand – Last executed command. Expressed as a number – See Communication Protocol document for reference.
- printerStatus – Five byte status code. See Responses section for full explanation.
- responseCommand – Direct IO command. Should be four bytes.
- responseData – Contains the fiscal protocol PDU string returned by the fiscal printer (from the point after the four byte command up to but not including the checksum).

An example response message:

```
<response success="true" code="" status="x">
  <addInfo>
    <elementList>lastCommand,printerStatus,responseCommand,responseData</elementList>
    <lastCommand>74</lastCommand>
    <printerStatus>20010</printerStatus>
    <responseCommand>1047</responseCommand>
    <responseData>0123071218130001</responseData>
  </addInfo>
</response>
```

5.3.3 clearText

Blanks the display. This sub-element has only one attribute to indicate the operator.

```
<clearText  operator="" />
```

5.3.4 displayText

Sends text messages to the customer display. You cannot insert carriage returns or line feeds so use spaces to pad out line 1 and begin line 2. This sub-element has two attributes; one to indicate the operator and one for the text itself. The maximum number of characters is 40. This reduces to 20 if used with printerTicket files.

```
<displayText  operator="" data="" />
```

5.4 printerFiscalReceipt

Emission of commercial documents (documenti commerciali) on RT fiscal printers is performed in the same manner as for fiscal receipts (scontrini fiscali) on MF fiscal printers in that the same printerFiscalReceipt XML file type is maintained.

These sub-elements can be divided into commands that are used to “manage” fiscal receipts and commercial documents and those that are used “inside” the individual print out which can consist of many records of different types. The former have only the operator attribute whereas the latter have many attributes that will be discussed further in detail.

Management commands:

N	Sub-element	Description
1	beginFiscalReceipt	Open fiscal receipt or commercial document
2	endFiscalReceipt	Close fiscal receipt or commercial document
3	printRecVoid	Cancel fiscal receipt or commercial document

Non-management commands:

N	Sub-element	Description
1	printRecItem	Item sale
2	printRecItemVoid	Cancel sale (Storno in Italy)
3	printRecItemAdjustment	Sale discount or surcharge
4	printRecItemAdjustmentVoid	Cancel sale discount or surcharge
5	printRecMessage	Additional row or description
6	printRecRefund	Goods return (Reso in Italy) *
7	printRecRefundVoid	Cancel goods return
8	printRecSubtotal	Print subtotal
9	printRecSubtotalAdjustment	Subtotal discount or surcharge
10	printRecSubtotalAdjustVoid	Cancel subtotal discount or surcharge
11	printRecTotal	Send payment
12	printGraphicCoupon	Print graphic coupon
13	printBarCode	Print barcode

* With RT printers, refunds are automatically converted to cancellations (storni).

Note: Apart from when issuing a credit note receipt (MF printers only), each fiscal receipt or commercial document file must comply with the following conditions:

- It must be opened.
- It must include at least one sale row.
- The total amount cannot be negative.
- The payment must be completed.
- It must be closed.

Multiple PrintRecTotal lines can be used if for example a customer pays partly by cheque and partly by cash (assuming the total due hasn’t already been reached or exceeded).

Additional header lines must be placed before beginFiscalReceipt as they are preloaded in memory.

5.4.1 Fiscal Receipt / Commercial Document Response File

Fiscal receipt responses contain the following additional information:

- lastCommand – 74 (The printer status command).
- printerStatus – Five byte status code. See Responses section for full explanation.
- fiscalReceiptNumber – Equals the number of the fiscal receipt or commercial document just printed.
- fiscalReceiptAmount – Equals the value of the fiscal receipt or commercial document just printed.
- fiscalReceiptDate – Equals the date of the fiscal receipt or commercial document just printed.
- fiscalReceiptTime – Equals the time of the fiscal receipt or commercial document just printed.
- zRepNumber – Equals the Z report sequence number for the day. *

An example response message:

```
<response success="true" code="" status="x">
  <addInfo>
    <elementList>lastCommand,printerStatus,fiscalReceiptNumber,fiscalReceiptAmount,fiscalReceiptDate,
fiscalReceiptTime,zRepNumber</elementList>
    <lastCommand>74</lastCommand>
    <printerStatus>20010</printerStatus>
    <fiscalReceiptNumber>nnnn</fiscalReceiptNumber>
    <fiscalReceiptAmount>yyyyy,cc</fiscalReceiptAmount>
    <fiscalReceiptDate>gg/mm/aaaa</fiscalReceiptDate>
    <fiscalReceiptTime>oo:mm</fiscalReceiptTime>
    <zRepNumber>zzzz</zRepNumber> *
  </addInfo>
</response>
```

* zRepNumber available from version 1.027.

5.4.2 printRecItem – Sale Items

Prints sale items on a fiscal receipt or commercial document.

```
<printRecItem  operator="" description="" quantity="" unitPrice="" department="" justification=""
/>
```

Adding 50 to the operator number will mean that the quantity line will be printed even if it's 1.

5.4.3 printRecItemVoid – Cancel Sale Items

Cancels the previous (last) sale operation or a previous one. Cancelling a previous (but not last) sale is known in Italy as a Storno. By setting the attribute voidLastItem="1" the cancellation is deemed a correction rather than a Storno. In this case, a printRecItem sub-element must immediately precede this sub-element (it cannot be the first transaction). The absence of the voidLastItem attribute or if the value is "0", deems the operation a Storno whereby specific sale information has to be passed just as in a sale item operation.

A Storno example:

```
<printRecItemVoid  operator="" description="" quantity="" unitPrice="" department=""
justification="" />
```

```
<printRecItemVoid  voidLastItem="0" operator="" description="" quantity="" unitPrice=""
department="" justification="" />
```

Adding 50 to the operator number will mean that the quantity line will be printed even if it’s 1.

A void last item example:

```
<printRecItemVoid  operator="" voidLastItem="1"
```

5.4.4 printRecItemAdjustment

A discount or surcharge (uplift) is applied either with reference to the previous sale operation or by a specific department number. When the last sale option is selected, the department value is ignored and can be null or the said attribute can be omitted. This is because the fiscal printer uses the department number relative to the last sale. When the last sale option is selected, this sub-element cannot be the first transaction in a fiscal receipt, commercial document or direct invoice. Amount must be greater than zero.

```
<printRecItemAdjustment  operator="" adjustmentType="" description="" amount=""
department="" justification="" />
```

Where:

- **adjustmentType** determines discount/uplift operation to perform:
  - 0 = Discount on last sale.
  - 3 = Discount on a department.
  - 5 = Surcharge on last sale.
  - 8 = Surcharge on a department.

Other values are not allowed.

Note: A zero amount will throw a printer error 16.

5.4.5 printRecItemAdjustmentVoid

Cancels the previous (last) discount or surcharge operation. A printRecItemAdjustment sub-element must immediately precede this sub-element.

```
<printRecItemAdjustmentVoid  operator="" />
```

5.4.6 printRecMessage

Applies additional header, additional sale descriptions or trailer lines to the fiscal receipt or commercial document. Also applies additional header and client lines to invoices plus prints or erases the EFT-POS transaction lines and sets customer ID in JSON file. One command is required for each printed row except for the EFT-POS transaction lines whereby this sub-element instructs the fiscal printer to print or erase all EFT-POS transaction lines from memory. Additional header lines must be sent prior to beginFiscalReceipt or prior to the credit note special opening command. Additional invoice header and client lines must be sent prior to beginFiscalDocument. From firmware version 3.008 onwards and with RT models, the command can also be used in direct invoices with messageType 4.

Regarding the additional invoice header lines, the SET 25 "RIGHE INIZIO FATTURA" setting on the fiscal printer must at least coincide with the index attribute setting. The default printer setting is zero which effectively disables the printing of type 5 lines even if the printer accepts the printRecMessage sub-element without returning an error.

```
<printRecMessage  operator="" messageType="" index="" font="" message="" /> or
<printRecMessage  operator="" messageType="4" message="" or
<printRecMessage  operator="" messageType="8" clearEFTPOSBuffer="" />
```

where:

- **messageType** defines the row type to be printed:
  - 1 = Additional header. This type must be placed before the beginFiscalReceipt sub-element.
  - 2 = Additional row (before MF logotype on MF printers and after NUMERO CONFEZIONI on RT printers).
  - 3 = Additional promo (after MF logotype and after NUMERO CASSA on RT printers).
  - 4 = Additional description (in the body of the fiscal receipt or commercial document).
  - 7 = Customer Id. Sets CustomerId field in www/json_files/rec.json file (see Fiscal Printer Intelligent Features Guide for details). The font has no relevance so the attribute can be omitted. Requires firmware version >= 4.01 or RT model.
  - 8 = Print or Erase all EFT-POS transaction lines. Requires firmware version >= 4.012 or RT model.

- **index** indicates the line number:
  - Range 1 - 9 for additional header (type 1).
  - Range 1 - 99 for additional promo and post total descriptions (types 2 and 3).
  - No meaning for additional row, Customer Id and EFT-POS transaction lines (types 4, 7 and 8). The attribute can be omitted.

- **font** attribute can be omitted when messageType is either 4, 7 or 8.

- **message** represents the text to be printed or the customer ID. The maximum lengths are as follows:

  - Message type 4 – 38 (or 37 with invoices).
  - Message type 7 – 46 (although native protocol limit is 64).
  - Message type 8 – Not applicable. Attribute can be omitted.
  - All other message types – 46.

  Further characters are truncated.

- **clearEFTPOSBuffer** attribute is only relevant when messageType is 8:

  - 0 = Print EFT-POS transaction lines.
  - 1 = Cancel EFT-POS transaction lines.

5.4.7 printRecRefund

In the case of RT printers, refunds are converted to corrections (storni). When goods are returned by a customer, a refund is required. The relative line is printed on the fiscal receipt. This is equivalent to "RESO MERCE" in Italy. Note that in Italy a fiscal receipt cannot be closed if the subtotal is negative (excluding payments). In this case a credit note has be raised (MF printers only) – see the **5.5**

Credit Note Fiscal Receipts **section.**

```
<printRecRefund  operator="" description="" quantity="" unitPrice="" department="" justification=""
/>
```

Adding 50 to the operator number will mean that the quantity line will be printed even if it's 1.

5.4.8 printRecRefundVoid

Cancels the previous (last) refund operation. a printRecRefund sub-element must immediately precede this sub-element.

```
<printRecRefundVoid  operator="" />
```

5.4.9 printRecSubtotal

This command allows the real-time subtotal to be printed and/or shown on the display.

<printRecSubtotal operator="" option="" />

where:

- **option** sets the subtotal option:
  - 0 = Print and show on the display.
  - 1 = Only print.
  - 2 = Only show on the display.

  If option 3 (request subtotal amount) is passed, the EpsonFPMate CGI ignores this sub-element and continues to the subsequent one.

5.4.10 printRecSubtotalAdjustment

A discount or surcharge (uplift) is applied on the subtotal. The department attribute is ignored and can be null or the attribute itself can be omitted. Discounts and surcharges are applied automatically and proportionally to each department based on the individual department total for this particular fiscal receipt, commercial document or direct invoice. Amount must be greater than zero.

<printRecSubtotalAdjustment operator="" adjustmentType="" description="" amount="" justification="" />

where:

- **adjustmentType** determines discount/uplift operation to perform:
  - 1 = Discount on subtotal with subtotal printed out.
  - 2 = Discount on subtotal without subtotal printed out.
  - 6 = surcharge on subtotal with subtotal printed out.
  - 7 = surcharge on subtotal without subtotal printed out.

Other values are not allowed.

Note: A zero amount will throw a printer error 16.

5.4.11 printRecSubtotalAdjustVoid

Cancels the previous (last) discount/surcharge on subtotal operation. Fixed wording is printed so only the operator attribute needs to be passed. A printRecSubtotalAdjustment sub-element must immediately precede this sub-element.

<printRecSubtotalAdjustVoid operator="" />

5.4.12 printRecTotal

One or more commands can be sent; more than one means that the payment is composed of several partial payments. In this case, once the total has been reached or exceeded, no more payment commands can be sent.

Note that insufficient payments are reflected in the wording "DIFFERENZA" on the display whereas payments with change show "RESTO". If the payment amount is exactly the same as the total outstanding to pay, the description attribute value will be shown on the display with the appropriate justification. The description on the receipt however will always be the description attribute value except in the following circumstance (MF models only):

- payment="0" paymentType="0" index="0" (as the only payment on the fiscal receipt

In this case the payment and RESTO lines are not printed at all.

Credit notes only require the operator attribute.

```
<printRecTotal  operator="" description="" payment="" paymentType="" index="" justification=""
/>
```

where:

- **paymentType** indicates the payment method:
  - 0 = Cash.
  - 1 = Cheque.
  - 2 = Credit or credit card.
  - 3 = Ticket.

- **index** is used with either credit cards or tickets to select which kind of credit card or ticket has been used. It is also used with cash to increment different cash totalizers inside the fiscal printer. index has no relevance with cheques. The following indexes are available:
  - Cash – 0 - 5.
  - Credit – 0
  - Credit card – 1 - 10.
  - Ticket – 1 - 10.

5.4.13 printGraphicCoupon

One graphic coupon can be printed at the end of the fiscal receipt or commercial document after the additional row, description or barcode. The sub-element must be placed between beginFiscalReceipt and endFiscalReceipt even though the coupon is printed after all the trailer lines, barcode and FOOTER. The binary data must be supplied in the base64 format and is passed as element data after the attributes rather than being a particular attribute value. The sub-element cannot be auto closed with /> and must therefore contain the closing sub-element name prepended with /. Do not use this command to print coupons without base64 data – It is not meant to be used with files uploaded via the upload.cgi service.

<printGraphicCoupon operator="" graphicFormat="" >*base64 data*</printGraphicCoupon>

where:

- **graphicFormat** indicates the original image format (irrespective of the base 64 encoding):
  - B = BMP.
  - R = Raster (file created with the Epson BMP to Raster utility or ESC/POS sequence).

5.4.14 printBarCode

Barcodes or QR codes are printed at the end of the fiscal receipt or commercial document after the additional trailer lines but before the FOOTER. Only one barcode can be printed in a fiscal receipt or commercial document. Different attributes can be used depending on the barcode type:

For standard barcodes:

<printBarCode operator="" position="" width="" height="" hRIPosition="" hRIFont="" codeType="" code="" />

For QR codes:

<printBarCode operator="" qRCodeAlignment="" qRCodeSize ="" qRCodeErrorCorrection="" codeType="" code="" />

where:

- **position** defines the starting position from the left margin (range 0 - 511). This attribute is not used with QR codes and can therefore be omitted.

- **qRCodeAlignment** defines the QR code position as follows:
  - 0 = Left aligned.
  - 1 = Centred.
  - 2 = Right aligned.

- **width** indicates the print dot width of each distinct bar (range 1 - 6). This attribute is not used with QR codes and can therefore be omitted.

- **height** indicates the height of the bar code measured in print dots (range 1 - 255). This attribute is not used with QR codes and can therefore be omitted.

- **qRCodeSize** indicates the QR code dimension (range 1 - 16).

- **hRIPosition** selects one of three ways to print the alphanumeric representation of the barcode or to disable it altogether. This attribute is not used with QR codes and can therefore be omitted. The options are as follows:

  - 0 = Disabled.
  - 1 = Below the barcode.
  - 2 = Above the barcode.
  - 3 = Below and above the barcode.

- **qRCodeDataType** indicates whether the QR code data is alphanumeric or binary. When binary is chosen, the code attribute value is represented by pairs of hexadecimal digits. For example, HELLO = 48454C4C4F. The options are as follows:

  - alphaNumeric / 0.
  - binary / 9.

- **hRIFont** indicates the font to be used for the HRI string. This attribute is not used with QR codes and can therefore be omitted. The options are as follows:

  - A.
  - B.
  - C (not all models).

- **qRCodeErrorCorrection** specifies the level of error correction to employ (range 0 - 3):

  - 0 = Low (smallest dimension).
  - 1 = Medium low.
  - 2 = Medium high.
  - 3 = High (largest dimension).

- **codeType** indicates the barcode or QR code standard. Choose from one of the following:

  - UPC-A / 65 *
  - UPC-E / 66 *
  - EAN13 / 67 *
  - EAN8 / 68 *
  - CODE39 / 69 *
  - ITF / 70 *
  - CODABAR / 71 *
  - CODE93 / 72 *
  - CODE128 / 73 *
  - QRCODE1 / 91 *
  - QRCODE2 / 92 *

* The use of number aliases requires firmware >= 4.00 or an RT model.

- **code** represents the barcode or QR code itself. QR codes up to 256 characters can be printed. If the qRCodeDataType attribute indicates a binary representation, this attribute value is represented by pairs of hexadecimal digits. For example, HELLO = 48454C4C4F.

Important Notes:

- The character set and the string length are defined by the standards related to each bar code system.

- The starting position, the string length and the barcode width must be defined according to the paper roll width. If there is not enough room width wise, the barcode will not be printed at all.

- According to the CODE128 standard, one of the following character sets must be specified:

  - CODE A. To select CODE A, the first two characters of the "code" attribute must be "{A"
  - CODE B. To select CODE B, the first two characters of the "code" attribute must be "{B"
  - CODE C. To select CODE C, the first two characters of the "code" attribute must be "{C"

  When employing CODE128, note that these first two characters are not part of the bar code itself and are therefore not printed. They are present only in order to select the character set.

- The four attributes qRCodeAlignment, qRCodeSize, qRCodeErrorCorrection and qRCodeDataType should only be used if codeType is QRCODE1 or QRCODE2.

- The five attributes position, width, height, hRIPosition and hRIFont should not be used if codeType is QRCODE1 or QRCODE2.

- CODABAR strings must contain the start and stop characters.

Please refer to chapter entitled H1=1; H2=075 – STAMPA BARCODE O QR PARAMETRIZZATO in the Communication Protocol document for complete details.

5.5 Credit Note Fiscal Receipts (MF Printers Only)

MF Fiscal printers can emit negative receipts solely related to goods refund operations. It means for example that customers can be refunded the same amount that was paid when the item was originally bought without having to purchase new items to make the receipt positive. The commands allowed in a Credit Note (Nota di Credito in Italy) are almost the same as those in a normal fiscal receipt. The root element remains printerFiscalReceipt but the following commands are not permitted:

printRecItem
printRecItemVoid

Note: Each credit note fiscal receipt request requires:

- The special open command.
- At least one goods return row.
- Payment (amount is ignored but with firmware < 3.007, the attribute with a value of zero must still be present in printRecTotal).
- endFiscalReceipt.

5.5.1 Credit Note Response File

Credit note responses are identical to fiscal receipt responses.

5.5.2 Opening a Credit Note

To open a Credit Note, a special type 4 "printRecMessage" sub-element must be used that has to begin with the following 16 character case-sensitive wording:

- "PRATICA DI RESO " (note the space after "RESO")

It is important to use capitals and separate the words with blanks. The string can be appended with whatever you like up to a further 16 characters making a total of 32. The wording must be unique for each credit note emitted. This 32 character limit is the same for all models. beginFiscalReceipt is not required with credit notes but can be still used for conformity if desired (placing it after printRecMessage).

The special printRecMessage command has to be the first in the credit note sequence (but after any type 1 "printRecMessage" optional additional header lines).

5.5.3 Closing a Credit Note

A Credit Note is closed by a payment command followed by endFiscalReceipt. The only attribute necessary in the payment command is "operator" since the amount reported on the receipt is the one computed by the fiscal printer itself based on refunds and any discounts etc. Furthermore, the total line uses fixed wording and the payment method is not relevant. Firmware versions < 3.007 still require the payment="0" attribute to be present (zero amount). The description attribute is necessary (can be null) with fpmate versions < 1.029.

5.6 Commercial Refund Document (Documento di Reso) (RT Printers Only)

Epson RT fiscal printers can print out commercial refund documents related to goods refund operations. It means for example that customers can be refunded the same amount that was paid when the item was originally bought without having to purchase new items to make the commercial document positive. The commands allowed depend on the SET 14/58 flag. In Ndc mode, the same commands as for credit notes are allowed and the same rules are followed. In non-Ndc mode, the same commands as for commercial documents are allowed and the same rules are followed. The root element remains printerFiscalReceipt. If the original document exists on the EJ, individual totals for each of the 20 tax groups present in the refund document cannot exceed the same totals in the original document. To check whether a particular document can be used as the reference document, a query document status request can be performed.

5.6.1 Commercial Refund Document Response File

Commercial refund document responses are identical to commercial document responses.

5.6.2 Opening a Commercial Refund Document

To open a commercial refund document, a special type 4 "printRecMessage" sub-element must be used that has to conform to one of two possible formats:

- RESO MERCE N.zzzz-nnnn del dd-mm-yyyy
- REFUND zzzz nnnn ddmmyyyy ssssssssss

where:

- **zzzz** equals the four digit Z report number
- **nnnn** equals the four digit document number
- **dd-mm-yyy** equals the date
- **sssssssssss** equals the eleven character fiscal serial number of the printer

It is important to use capitals and separate the fields with blanks. beginFiscalReceipt is not required with commercial refund documents but can be still used for conformity if desired (placing it after printRecMessage). If the fiscal serial number is not the same as the printer that receives the request, an EJ search is not performed. Currently there is no fiscal printer intercommunication function. The special printRecMessage command has to be the first in the commercial refund document sequence (but after any type 1 "printRecMessage" optional additional header lines).

5.6.3 Closing a Commercial Refund Document

In Ndc mode (flag SET 14/58), the same rules as for credit notes are followed except that partial payments are accepted. In non-Ndc mode, the same rules as for commercial documents are followed.

5.7 Commercial Void Document (Documento di Annullo) (RT Printers Only)

Epson RT fiscal printers can print out commercial void documents. It means for example that an erroneously printed commercial document can be completely voided. With automatic printing, only printRecMessage is used otherwise the commands allowed depend on the SET 14/59 flag. In Ndc mode, the same commands as for credit notes are allowed and the same rules are followed. In non-Ndc mode, the same commands as for commercial documents are allowed and the same rules are followed. With automatic printing, two approaches are possible:

- printerCommand root element with a single Direct IO command or
- printerFiscalReceipt root element with printRecMessage followed by the original transactions in case the document is not found.

If the original document exists on the EJ, the opening command will automatically print and close the new void document. Therefore, before proceeding a query document status request can be performed to ascertain whether a particular document can be used as the reference document.

5.7.1 Commercial Void Document Response File

Automatic printRecMessage responses and non-automatic responses are identical to commercial document responses.

Automatic Direct IO responses contain the following additional information:

- lastCommand – 74 (The printer status command).
- printerStatus – Five byte status code. See Responses section for full explanation.
- responseCommand – 1078
- responseData - Either matches the operator number used in the request or adds an offset of 50

An example response message:

```
<response success="true" code="" status="x">
  <addInfo>
    <elementList>lastCommand,printerStatus,responseCommand,responseData</elementList>
    <lastCommand>74</lastCommand>
    <printerStatus>20010</printerStatus>
    <responseCommand>1078</responseCommand>
    <responseData>51</responseData>
  </addInfo>
</response>
```

5.7.2 Automatically Printing a Commercial Void Document (Direct IO)

Once it has been established that a document exists and can be voided, a single opening command is required. Make a Direct IO call with native command 1078 in a printerCommand file (see example in this document). The operator reply number (responseData) should be verified to ascertain whether the automatic generation was successful. If the operator number has an offset of 50 when compared to the operator number used on the request, it means that the automatic print-out was successful. If it is the same number, it means that a commercial void document was merely opened since the reference document was not found. In this condition, a resetPrinter command should be sent to void the open document.

5.7.3 Automatically Printing a Commercial Void Document (printRecMessage)

The format is the same as for non-automatic printing. If the document is found and a commercial void document is automatically printed, XML lines after printRecMessage are not processed.

5.7.4 Opening a Commercial Void Document

Once it has been established that a document does not exist or cannot be automatically voided, a single opening command is required. To open a commercial void document, a special type 4 "printRecMessage" sub-element must be used that has to conform to one of two possible formats:

- ANNULLAMENTO N.zzzz-nnnn del dd-mm-yyyy
- VOID zzzz nnnn ddmmyyyy sssssssssss

where:

- **zzzz** equals the four digit Z report number
- **nnnn** equals the four digit document number
- **dd-mm-yyyy** or **ddmmyyyy** equals the date
- **sssssssssss** equals the eleven character fiscal serial number of the printer

It is important to use capitals and separate the fields with blanks. beginFiscalReceipt is not required with commercial void documents but can be still used for conformity if desired (placing it after printRecMessage). If the fiscal serial number is not the same as the printer that receives the request, an EJ search is not performed. Currently there is no fiscal printer intercommunication function.

The special printRecMessage command has to be the first in the commercial void document sequence (but after any type 1 "printRecMessage" optional additional header lines).

5.7.5 Closing a Commercial Void Document

In Ndc mode (flag SET 14/59), the same rules as for credit notes are followed except that partial payments are accepted. In non-Ndc mode, the same rules as for commercial documents are followed.

5.8 printerNonFiscal

Epson fiscal printers can also emit non-fiscal receipts and administrative documents. The relative sub-elements can be divided into commands that are used to "manage" them and those that are used "inside" the individual receipt or document which consists of only a few commands when compared to a fiscal receipt or commercial document. Management sub-elements only have the operator attribute whereas the others have many attributes that will be discussed further in detail. printerNonFiscal cannot be used with non-fiscal TM printers.

Management commands:

N	Sub-element	Description
1	beginNonFiscal	Open non-fiscal receipt or administrative document
2	endNonFiscal	Close non-fiscal receipt or administrative document

Non-management commands:

N	Sub-element	Description
1	printBarCode	Print barcode
2	printGraphicCoupon	Print graphic coupon
3	printNormal	Print row
4	printRecMessage	Prints or erases EFT-POS response lines
5	setLogo	Uploads logos to NVRAM and programs SET15 HEADER and FOOTER indexes

5.8.1 Non-fiscal Receipt / Administrative Document Response File

Responses only contain the two basic additional information entries (lastCommand and printerStatus). An example response message:

```
<response success="true" code="" status="x">
        <addInfo>
                <elementList>lastCommand,printerStatus</elementList>
                <lastCommand>74</lastCommand>
                <printerStatus>00100</printerStatus>
        </addInfo>
</response>
```

5.8.2 printBarCode

The printing of barcodes and QR codes in non-fiscal receipts and administrative documents uses the same commands and rules as described in the printerFiscalReceipt section (printBarCode). However, unlike a fiscal receipt or commercial document, there is no limit to the number of barcodes or QR codes that can be printed. Barcodes and QR codes are printed immediately.

5.8.3 printGraphicCoupon

The printing of graphic coupons in non-fiscal receipts and administrative documents uses the same commands and rules as described in the printerFiscalReceipt section (printGraphicCoupon). The only difference is that while in fiscal receipt or commercial document mode the graphic coupon can only be printed once at the end of the print-out, in non-fiscal receipt / administrative document mode it can be printed everywhere, and more than once. The printing of several different graphic images is therefore allowed. The sub-element must be used between beginNonFiscal and endNonFiscal.

5.8.4 printNormal

Prints a row. All four fonts are supported. The format is as follows:

```
<printNormal  operator="" font="" data="" />
```

where:

- **data** contains the text to be printed (max 46 characters).

5.8.5 printRecMessage

When used in a non-fiscal receipt or administrative document, the **messageType** attribute has to be set to 8 (Print or Erase all EFT-POS transaction lines). The only attribute necessary apart from the operator is **clearEFTPOSBuffer.** All other attributes can be omitted. See printRecMessage.

5.8.6 setLogo

setLogo in a non-fiscal receipt or administrative document can perform one of three functions:

1. Logo upload to NVRAM or
2. Index programming (SET 15 / 9, 10, 19 and 20) or
3. Both the above

When uploading logos to NVRAM, the binary data must be supplied in the base64 format. It is passed as element data after the attributes rather than being a particular attribute value. In this case, the sub-element cannot be auto closed with /> and must therefore contain the closing sub-element name prepended with /. Graphic logo uploading to NVRAM must be done via a non-fiscal receipt or administrative document. setLogo can also be used in a printerCommand or printerCommands file but in this case only the indexes can be programmed. Do not use this command to upload logos to NVRAM without base64 data – It is not meant to be used with files uploaded via the upload.cgi service.

```
<setLogo  operator="" location="" index="" option="0" graphicFormat="" >base64 data</setLogo>
<setLogo  operator="" index="1" option="1" graphicFormat="B" >base64 data</setLogo>
<setLogo  operator="" option="1" graphicFormat="R" >base64 data</setLogo>
<setLogo  location="" index="" option="2" />
```

where:

- **location** defines where the logo will be printed. To program two different logos as HEADER and FOOTER would require two distinct setLogo sub-elements. They could of course reside in a single XML file. The options are as follows:

    - 0 = All positions (HEADER and FOOTER for receipts plus HEADER and FOOTER for invoices)
    - 1 = Receipt HEADER only
    - 2 = Receipt FOOTER only
    - 3 = Invoice HEADER only
    - 4 = Invoice FOOTER only

- **index** specifies the number that will be programmed in the positions specified by the location attribute. Values range from 0 to 9. Zero deactivates logo printing. It is also required if the file format is BMP but in this case must not be zero. Raster files created by the BMP to Raster utility on the other hand already contain the index in the .bin file header (based on the key code pair values specified during the conversion).

- **option** specifies the one of three modes as follows:

    - 0 = Upload and SET 15 programming.
    - 1 = Upload only.
      In this case, the location attribute is not necessary and the index attribute is only necessary if the graphic format is BMP.
    - 2 = SET 15 programming only.
      In this case since base64 data is not supplied, the setLogo sub-element can be auto-closed with /> and the graphicFormat attribute can be omitted. The operator is also not needed.

- **graphicFormat** specifies one of two formats (before base 64 encoding) as follows:

    - B = BMP
    - R = Raster (File created with Epson BMP to Raster utility).

5.9 printerFiscalReport

EpsonFPMate can instruct the printer to perform a daily fiscal closure (Z report), a daily financial report (X report) or both. Only the operator attribute is necessary. From version 1.030, a timeout attribute can be used (except printXReport). This is necessary with RT printers to allow time for data transmission to the tax authority server. Only one report command is used per request file. Common commands such as displayText can however be used but must be placed before the report command. See examples at the end of this document.

Sub-element	Description
printXReport	Daily financial report (X Report)
printZReport	Daily fiscal closure (Z Report) *
printXZReport	X Report followed immediately by Z report *

* In an RT printer, data transmission to the tax authority server automatically ensues.

5.9.1 Report Response File

Report responses contain the following additional information:

- lastCommand – The last command that EpsonFPMate generated internally. Expressed as a number – See Communication Protocol document for reference.
- printerStatus – Five byte status code. See Responses section for full explanation.
- zRepNumber – Equals the Z report sequence number (the closure just printed).
- dailyAmount – Equals the cumulative daily amount.

An example response message:

```
<response success="true" code="" status="x">
        <addInfo>
                <elementList>lastCommand,printerStatus,zRepNumber,dailyAmount</elementList>
                <lastCommand>74</lastCommand>
                <printerStatus>20110</printerStatus>
                <zRepNumber >11</ zRepNumber>
                <dailyAmount >332,33</ dailyAmount>
        </addInfo>
</response>
```

5.9.2 Report Request Types

The following three report types are supported:

N	Sub-element	Description	XML
1	printXReport	Financial report only	<printXReport operator="" />
2	printZReport	Fiscal daily closure only	<printZReport operator="" timeout="" />
3	printXZReport	Financial report and fiscal daily closure (in that order)	<printXZReport operator="" timeout="" />

From version 1.030 onwards, the "timeout" attribute can also be passed. The value is expressed in milliseconds.

5.10 printerCommand

EpsonFpMate embedded has the capability to extract administrative/diagnostic information from a fiscal printer as well as perform resets. Only one command can be used per request file. If you wish to send more than one, please use printerCommands. Common commands such as displayText but not directIO can however be used but must be placed before the main command. See examples at the end of this document. The following table lists the commands:

N	Sub-element	Description
1	authorizeSales	Activates EFT-POS electronic payment.
2	beginTraining	Activates training mode. If fiscal receipts or commercial documents have been issued following a fiscal closure, this command is inhibited (a closure must be requested beforehand either by executing a printXZReport or a printZReport command).
3	endTraining	Deactivates training mode.
4	getDate	Retrieves the printer date and time.
5	printContentByDate	Prints Electronic Journal data.
6	printContentByNumbers	
7	printDuplicateReceipt	Prints last fiscal receipt or commercial document.
8	printRecCash	For performing CASH IN or CASH OUT operations.
9	printRecVoid	Cancels any open fiscal receipt, commercial document, credit note or direct invoice
10	queryContentByDate	Reads Electronic Journal data.
11	queryContentByNumbers	
12	queryPrinterStatus	Retrieves printer basic or RT status.
13	resetPrinter	Resets fiscal printer. Cancels any open receipts, commercial documents or invoices. Closes any open non-fiscal receipts or administrative documents.
14	setDate	Sets the date and time.
15	setLogo	Programs SET 15 HEADER and FOOTER indexes for receipts, documents and invoices.

5.10.1 Command Response File

Command responses incorporate the basic additional information entry (lastCommand) with command specific elements which are described in each separate command.

A typical response example:

```
<response success="true" code="" status="x">
        <addInfo>
                <elementList>lastCommand,printerStatus</elementList>
                <lastCommand>74</lastCommand>
                <cpuRel>4.018</cpuRel>
                <mfRel>04.3</mfRel>
                <mfStatus>0</mfStatus>
                <fpStatus>00110</fpStatus>
        </addInfo>
</response>
```

5.10.2 authorizeSales

Activates EFT-POS payment authorisation. The only attributes necessary are operator and amount.

```
<authorizeSales  operator="" amount="" />
```

The following additional information is provided in the response:

```
<elementList>lastCommand,printerStatus,paymentResult,lineCount</elementList>
<lastCommand>278</lastCommand>
<printerStatus>00110</printerStatus>
<paymentResult><Max 122 byte string from POS></paymentResult> *
<lineCount>31
        <lineNumber1>  INGENICO ITALIA SPA                    </lineNumber1>
        <lineNumber2>   DIVISIONE MONETICA                    </lineNumber2>
                ...
                ...
        <lineNumber31>Nessun addebito eseguito                </lineNumber31>
</lineCount>
```

* The paymentResult element was introduced from version 1.022.

5.10.3 beginTraining

No attributes are required. Entering training mode is only possible if no fiscal receipts or commercial documents have been issued following a daily fiscal closure.

```
<beginTraining />
```

5.10.4 endTraining

No attributes are required.

```
<endTraining />
```

5.10.5 getDate

No attributes are required. Retrieves the intelligent or RT fiscal printer current date and time.

```
<getDate />
```

The following additional information is provided in the response:

```
<elementList>day,month,year,hour,minute</elementList>
<day></day>
<month></month>
<year></year>
<hour></hour>
<minute></minute>
```

5.10.6 printContentByDate

Used to reprint fiscal receipts, commercial documents, invoices, ticket box office slips, credit notes and fiscal closures that have been stored in the EJ or MPD.

```
<printContentByDate  operator="" dataType="" fromDay="" fromMonth="" fromYear="" toDay="" toMonth="" toYear="" />
```

where:

- **dataType** indicates the type of data to collect:
  - 0 = All.
  - 1 = Fiscal receipts, commercial documents commercial void documents, commercial refund documents, credit notes and fiscal closures.
  - 2 = Invoices.
  - 3 = Box office tickets (Titoli di Accesso).
  - 4 = Credit notes.
  - 5 = Fiscal closures.
- **fromDay** indicates the start day expressed in one or two digits.
- **fromMonth** indicates the start month expressed in one or two digits.
- **fromYear** indicates the start year expressed in either two or four digits.
- **toDay, toMonth and toYear** together indicate the end date in the same format as the three "from" attributes.

5.10.7 printContentByNumbers

```
<printContentByNumbers  operator="" dataType="" day="" month="" year="" fromNumber="" toNumber="" />
```

where:

- **dataType** has the same options as printContentByDate
- **day** indicates the day expressed in one or two digits.
- **month** indicates the month expressed in one or two digits.
- **year** indicates the year expressed in either two or four digits.
- **fromNumber** indicates the first number of the fiscal receipt to print for example. Values range from 0001 to 9999.
- **toNumber** indicates the last number of the fiscal receipt to print for example. Values range from 0001 to 9999.

5.10.8 printDuplicateReceipt

The last fiscal receipt or commercial document (including refund or void variants) is reprinted. This command cannot print fiscal receipts or commercial documents issued prior to a fiscal daily closure. Since originals cannot by law be reprinted in the same format, copies are in fact non-fiscal receipts or administrative documents that contain the original lines as content read from the EJ or MPD.

```
<printDuplicateReceipt  operator="" />
```

5.10.9 printRecCash

This command is used when cash or cheques are paid in or out of the cash drawer.

```
<printRecCash  operator="" direction="" form="" amount="" />
```

where:

- **direction** is either "in" or "out".
- **form** is either "cash" or "cheque" / "check".

5.10.10 printRecVoid

The current fiscal receipt, credit note or direct invoice is cancelled. endFiscalReceipt or endFiscalDocument is still required.

```
<printRecVoid  operator="" />
```

5.10.11 queryContentByDate

Used to collect and return EJ or MPD data in an array.

```
<queryContentByDate  operator="" dataType="" fromDay="" fromMonth="" fromYear="" toDay=""
toMonth="" toYear="" />
```

Where the attributes follow the same rules as for printContentByDate.

The content is contained in the additional information portion in the following format:

```
<elementList>lastCommand,printerStatus,lineCount</elementList>
<lastCommand>74</lastCommand>
<printerStatus>00110</printerStatus>
<lineCount>
        <lineNumber1>xxxxxxxxxxxxxx</lineNumber1>
        <lineNumber2>xxxxxxxxxxxxxx</lineNumber2>
        And so on
</lineCount>
```

A maximum of 200 lines can be read at a time. From version 1.019, no fixed line limit has been set. EpsonFpMate continually monitors the free memory available whilst the query is in progress. If no more free memory is available, the query is terminated. In this case, the last line wording is set to "OUT OF MEMORY" so that the user can be made aware.

5.10.12 queryContentByNumbers

Used to collect and return the Electronic Journal data in an array.

```
<queryContentByNumbers  operator="" dataType="" day="" month="" year="" fromNumber=""
toNumber="" />
```

Where the attributes follow the same rules as for printContentByNumbers.

The content is contained in the additional information portion in the same format as queryContentByDate. The same limits apply.

5.10.13 queryPrinterStatus

Retrieves the intelligent or RT printer status. With MF models, only one type of status can be requested. With RT models, two types. The operator attribute is necessary plus a statusType indication attribute.

```
<queryPrinterStatus  operator="" statusType="" />
```

where:

- **statusType** is either "0" for basic status or "1" for RT status.

The following additional information is provided in the basic status response:

```
<elementList>lastCommand,cpuRel,mfRel,mfStatus,fpStatus</elementList>
<lastCommand>74</lastCommand>
<cpuRel></cpuRel>
<mfRel></mfRel>
<mfStatus></mfStatus>
<fpStatus></fpStatus>
```

where:

- **cpuRel** indicates the fiscal board firmware version.
- **mfRel** indicates the fiscal memory / MPR firmware version.
- **mfStatus** indicates the fiscal memory / MPR status (OK, FULL etc.).
- **fpStatus** indicates the printer status (same as printerStatus five byte code). See Responses section for full explanation.

The following additional information is provided in the RT status response:

```
<elementList>
lastCommand,rtType,rtMainStatus,rtSubStatus,rtDailyOpen,rtNoWorkingPeriod,rtFileToSend,
rtOldFileToSend,rtFileRejected
</elementList>
<lastCommand>138</lastCommand>
<rtType></rtType>
<rtMainStatus></rtMainStatus>
<rtSubStatus></rtSubStatus>
<rtDailyOpen></rtDailyOpen>
<rtNoWorkingPeriod></rtNoWorkingPeriod
<rtFileToSend></rtFileToSend>
<rtOldFileToSend rtOldFileToSend>
<rtFileRejected></rtFileRejected>
```

where:

- **rtType** indicates the RT device type *
- **rtMainStatus** indicates the RT main status *
- **rtSubStatus** indicates the RT sub status *
- **rtDailyOpen** = indicates the day open logical condition (0=closed and 1=open)
- **rtNoWorkingPeriod** indicates whether or not a Z report must be performed (0=no and 1=yes)
- **rtFileToSend** indicates the number of files due to be sent to the tax authority
- **rtOldFileToSend** indicates the number of files due to be sent to the tax authority but still waiting on the printer after a configurable number of days (SET 15/25)
- **rtFileRejected** indicates the number of files rejected by the tax authority

* See Communication Protocol document.

5.10.14 resetPrinter

The operator attribute is the only one required. This command will execute the following functions:

- Return the printer to the STATO REGISTRAZIONE state independent of the current state.
- Close any open non-fiscal receipts.
- Close any open administrative documents.
- Cancel any open fiscal receipts.
- Cancel any open commercial documents.
- Cancel any open credit notes.
- Cancel any open direct invoices printing two copies.
- Cancel any open free invoices printing one copy.
- Cancel any open box office tickets plus return the printer to the "SCONTRINO FISCALE" mode.
- Unblock the keyboard that was previously blocked with the 1055 native protocol command.
- Clean the keyboard buffer.

```
<resetPrinter  operator="" />
```

5.10.15 setDate

Programs the intelligent or RT fiscal printer date and time. It is only possible if no fiscal receipts or commercial documents have been issued following a daily fiscal closure.

```
<setDate  day="" month="" year="" hour="" minute="" />
```

where:

- **day** is either one or two digits. Leading zero is optional.
- **month** is either one or two digits. Leading zero is optional.
- **year** must be four digits. From version 1.015 onwards, two digits are also accepted.
- **hour** is represented in the 24 hour format and is either one or two digits. Leading zero is optional.
- **minute** is either one or two digits. Leading zero is optional.

  It is not possible to set the number of seconds.

5.10.16 setLogo

This command is used to perform SET 15 logo programming in order to specify where graphic logos are printed. When used in a printerCommand file, setLogo cannot be used to upload logos into NVRAM. Please refer to the setLogo section in the printerNonFiscal chapter for an explanation of the attributes (setLogo). Please note that only location and index are necessary and since base64 data is not supplied, the setLogo sub-element can be auto-closed with />. The operator attribute is also not required.

```
<setLogo  location="" index="" />
```

5.11 printerCommands (from version 1.021)

This file type permits multiple elements. The command set is the same as for printerCommand except for the following elements as they generate specific responses:

- queryContentByDate
- queryContentByNumbers
- getDate
- authorizeSales

The response is the same as the queryPrinterStatus response.

5.12 printerFiscalDocument

Epson intelligent and RT fiscal printers can also emit four types of invoices (fatture). These sub-elements can be divided into commands that are used to "manage" invoices and those that are used "inside" them. printerFiscalDocument cannot be used with non-fiscal TM printers. When thermal paper is used, two invoice copies are automatically printed. Cancelled free invoices however only print one copy. Free invoices and invoices based on the last fiscal receipt are supported from firmware version 3.00D. Receipt based free invoices are supported from firmware version 4.001/4.01. RT models only support direct invoices.

Management commands:

N	Sub-element	Description
1	beginFiscalDocument	Open direct, free or receipt based free invoice or Request invoice based on last fiscal receipt

2	endFiscalDocument * ** or
	endFiscalReceipt **
		Close invoice

* endFiscalDocument together with operationType is used with free invoices and receipt based free invoices.

** endFiscalDocument or endFiscalReceipt without operationType are used with direct invoices.

Neither endFiscalDocument nor endFiscalReceipt are required with invoices based on the last fiscal receipt.

Non-management commands:

Commands used with direct invoices:

N	Sub-element	Description
1	printRecItem	Item sale
2	printRecItemVoid	Cancel sale (Storno in Italy)
3	printRecItemAdjustment	Sale discount
4	printRecItemAdjustmentVoid	Cancel sale discount
5	printRecMessage *	Additional header line, client line or additional description line in the body of the direct invoice
6	printRecRefund	Goods return (Reso in Italy) RT printers automatically convert refunds to corrections (storni)
7	printRecRefundVoid	Cancel goods return Not applicable with RT printers
8	printRecSubtotal	Subtotal cannot be printed on invoices. This command can only be used to display the subtotal (option = 2). Since it would normally be overwritten immediately, it is not usually used.
9	printRecSubtotalAdjustment	Subtotal discount
10	printRecSubtotalAdjustVoid	Cancel subtotal discount
11	printRecTotal	Send payment

* Printing of additional description lines in the body of the direct invoice requires firmware >= 3.008.

Commands used with invoices based on the last fiscal receipt:

N	Sub-element	Description
1	printRecMessage	Print additional header line or client line

Commands used with free invoices and receipt based free invoices:

N	Sub-element	Description
1	printFiscalDocumentLine	Print a text row

5.9.1 Fiscal Document Response File

Fiscal document responses only contain the two basic additional information entries (lastCommand and printerStatus). An example response message:

```
<response  success="true" code="" status="x">
        <addInfo>
                <elementList>lastCommand,printerStatus</elementList>
                <lastCommand>74</lastCommand>
                <printerStatus>00110</printerStatus>
        </addInfo>
</response>
```

5.9.1 Document Types

The following table illustrates the different document types and requests. Examples of each can be found in the examples section. lastReceiptInvoice and freeInvoice types require firmware >= 3.00D. The receiptBasedFreeInvoice type requires firmware >= 4.001/4.01. RT models only support direct invoices.

N	Sub type	documentType Attribute Identifier	Description
1	Based on the last fiscal receipt	lastReceiptInvoice	MF models only. Apart from the additional header and/or client lines, only beginFiscalDocument is sent. endFiscalDocument or endFiscalReceipt are unnecessary since the whole print out is automated.
2	Direct	directInvoice	Any additional header and/or client lines must be sent first. Then after having opened a direct invoice, printerFiscalReceipt sub-elements are employed such as printRecItem, printRecTotal etc. rather than printFiscalDocumentLine for example. The following printerFiscalReceipt commands are not allowed: • printRecMessage messageType 2 and 4 * • printGraphicCoupon • printBarCode ** After payment, either endFiscalReceipt or endFiscalDocument is sent. * From MF firmware version 3.008 onwards and with RT models, printRecMessage with messageType 4 can be used. From MF firmware version 4.108 onwards and with RT models, messageType 2 can be used. ** From MF firmware version 4.017 onwards and with RT models, printBarCode can be used.
3	Free	freeInvoice	MF models only. printerFiscalReceipt commands such as printRecItem and printRecMessage cannot be sent. Apart from the total line *, all other lines are printed using printFiscalDocumentLine. The pre-programmed fiscal retail header is not printed automatically. * On the FP-H6000, use printFiscalDocumentLine to print any total lines since they are not printed automatically by the printer in JavaPOS mode.

N	Sub type	documentType Attribute Identifier	Description
4	Receipt based free invoice	receiptBasedFreeInvoice	MF models only. Same behaviour as for freeInvoice print-outs but see note below.

The freeInvoice and receiptBasedFreeInvoice identifiers emit invoices in exactly the same manner but increment a different fiscal daily closure totalizer as follows:

- freeInvoice – "TOTALE FATTURE (DIRETTE)" is incremented.
- receiptBasedFreeInvoice – "TOTALE FATTURE (DA SCONTRINO)" is incremented.

5.9.1 Direct Invoice Sub-Elements

All printerFiscalReceipt sub-elements can be used in direct invoices except for printGraphicCoupon and printBarCode *. The description limits are reduced from 38 to 37 characters. printRecTotal descriptions are not printed but are sent to the display so the character limit is reduced to 20 characters. printRecMessage is described below.

* From MF firmware version 4.017 onwards and with RT models, printBarCode can be used.

5.9.2 beginFiscalDocument

The attributes vary according to the type of requested document. Attributes meant for a different type are simply ignored. The following XML line shows all the attributes for illustrative purposes:

```
<beginFiscalDocument operator="" documentAmount="" documentType="" documentNumber="" />
```

where:

- **documentAmount** is the total amount. MF models only. It only has relevance with the following types:

  - Free invoices (supported from firmware version 3.00D).
  - Receipt based free invoices (supported from firmware version 4.001/4.01).

  From firmware 4.001 onwards, the invoice value can be zero.

- **documentType** can be one of the four types illustrated in the above table. RT models only support the directInvoice type. Set the attribute based on the names shown in the appropriately named column of the above table.

- **documentNumber** starts from 0 * and runs up to 99999 or from 1 to 30000. Depending on the printer configuration and documentType, requests can be refused if numbers do not run in sequence. It only has to be specified (from 0 to 99999) with the following two types:

  - Invoices based on the last fiscal receipt (supported from firmware version 3.00D).
  - Direct invoices.

  * To use zero requires MF firmware >= 3.008 or an RT model – Use this value if you wish the fiscal printer to manage the invoice numbering. The printer automatically increments the invoice number after having printed one of the two abovementioned invoice types. The documentNumber attribute can also be omitted – the zero value will be assumed. With MF firmware versions < 3.008, you must supply the documentNumber starting from 1.

  Optionally specifying this attribute can be useful when printing Free invoices or Receipt based free invoices on thermal paper since it then becomes possible to recall and reprint the free invoice from the Electronic Journal based on this number. In this case however, the range runs from 1 to 30000.

5.9.3 printRecMessage

This sub-element is only used with the following document types:

- Invoices based on the last fiscal receipt.
- Direct invoices.

```
<printRecMessage  operator="" messageType="" index="" message="" /> or
<printRecMessage  operator="" messageType="4" message="" />
```

Where:

- **messageType** defines the row type to be printed:

  - 2 = Trailer lines (after TOTALE EURO line). Font always normal. *
  - 4 = Additional description (in the body of the Direct invoice).
  - 5 = Additional invoice header lines.
  - 6 = Invoice client lines.

* Type 2 lines require MF firmware version 4.018 or later or an RT model.

- **index** indicates the line number:

  - Range 1 - 99 for trailer lines (type 2).
  - Range 1 - 20 for invoice additional header lines (type 5).
  - Range 1 - 5 for invoice client lines (type 6).
  - No meaning for additional row type 4. The attribute can be omitted.

- **font** attribute is not necessary since the font is always normal when the messageType is either 2, 4, 5 or 6.

- **message** represents the text to be printed. The maximum lengths are as follows:

  - Message type 4 – 37.
  - Message types 2, 5 and 6 – 46.

  Further characters are truncated.

5.9.4 printFiscalDocumentLine

MF models only. This sub-element is supported from firmware version 3.00D. An unlimited number of text rows can be printed still respecting the 1000 line general intelligent fiscal printer limit. The only requirement is that they are placed after beginFiscalDocument and prior to endFiscalDocument. All four fonts are supported. This sub-element is only used with the following document types:

- Free invoices.
- Receipt based free invoices.

```
<printFiscalDocumentLine  operator="" font="" documentLine="" />
```

where:

- **documentLine** represents the text to be printed. The maximum length is as follows:

  - 46

  Further characters will be truncated.

5.9.5 printBarCode

Requires MF firmware >= 4.017 or an RT model. Just as for fiscal receipts and commercial documents, only one barcode or QR code can be printed per invoice. The same code is printed on both copies at the foot of the print out after an eventual graphic footer.

5.9.6 endFiscalDocument

This sub-element is supported from MF firmware version 3.00D onwards and with RT models. It is used to either close or cancel the following document types:

- Direct invoices (close only).
- Free invoices (close or cancel). MF models only.
- Receipt based free invoices (close or cancel). MF models only.

```
<endFiscalDocument  operator="" operationType="" />
```

where:

- **operationType** is either:
  - 0 = Close free document.
  - 1 = Cancel free document.

  operationType is not used with direct invoices.

5.9.7 Important Prerequisites

- Previous set-up of the fiscal printer is required.

- The option to print an invoice based on a fiscal receipt is only possible if the request immediately follows a "printerFiscalReceipt" request. The invoice reports all the data present in the previous receipt apart from barcodes and payment descriptions.

- With invoices based on a fiscal receipt and direct invoices, the invoice numbers must be **greater** than the last invoice number printed. But it doesn't have to be previous + 1, it could be previous + 5 for example. With MF firmware >= 3.008 or an RT model, the value zero can be used so that it is managed by the printer.

- With invoices based on a fiscal receipt and direct invoices, any additional headers and/or client lines must be sent before beginFiscalDocument.

- With free invoices and Receipt based free invoices, the retail header is not automatically printed. The retail header lines can be printed using printFiscalDocumentLine. They can be retrieved for use via a directIO request sequence.

5.10 printerTicket

Epson intelligent and RT fiscal printers can also emit box office tickets. The printer must be in shared mode whereby it can print both "titoli" and all the other types of print-out. Shared mode is deactivated by default. If you wish to print "titoli", please contact Epson since to enable shared mode requires printer intervention including the breaking of the fiscal seal. Box office tickets can also contain multiple barcodes plus it is also possible to use common commands such as openDrawer and displayText. Note that when printing box office tickets, text can only be displayed on the top line of the display.

The relative sub-elements can be divided into commands that are used to "manage" box office tickets and those that are used "inside" the individual ticket which consists of only a few commands when compared to a fiscal receipt or a commercial document. The operator attribute is not used. printerTicket cannot be used with non-fiscal TM printers.

Management commands:

N	Sub-element	Description
1	enterTicket	Switch to box office ticket mode
2	beginTicket	Open a new ticket
3	endTicket	Close ticket
4	exitTicket	Switch out of box office ticket mode to fiscal receipt or commercial document mode
5	printTicketVoid	Cancel ticket (automatically switches out of box office ticket mode to fiscal receipt or commercial document mode)

Non-management commands:

N	Sub-element	Description
1	printTicket	Print one of two types of text line
2	printBarCode	Print barcode or QR code

5.10.1 Ticket Response File

Responses contain the following additional information:

- lastCommand
- cpuRel
- mfRel
- mfStatus
- fpStatus

A basic response example:

```
<response success="true" code="" status="x">
        <addInfo>
                <elementList>lastCommand,printerStatus</elementList>
                <lastCommand>74</lastCommand>
                <cpuRel>4.018</cpuRel>
                <mfRel>04.3</mfRel>
                <mfStatus>0</mfStatus>
                <fpStatus>00110</fpStatus>
        </addInfo>
</response>
```

5.10.2 enterTicket

Switches into box office ticket mode.

```
<enterTicket  randomNum="" controlCode="" />
```

Where:

- **randomNum** equals a random number from 01 to 99
- **controlCode** equals the four digit control code. After having signed a non-disclosure agreement (NDA), Epson can supply the instructions on how to calculate this code.

5.10.3 beginTicket

Opens a new box office ticket.

```
<beginTicket  randomNum="" controlCode="" />
```

The attributes follow the same rules as for enterTicket.

5.10.4 printTicket

Prints one of two different types of text line in box office tickets. The two types are “normal” or “seal”. Normal lines are printed and saved in the Electronic Journal. Seal lines of which there must be one per box office ticket are only printed. The printer returns an error if a second seal line sub-element is received. All four fonts are supported.

```
<printTicket  randomNum="" controlCode="" textType="R" font="1" text="" /> or
<printTicket  randomNum="" controlCode="" textType="S" font="1" text="" />
```

Where:

- **randomNum** and **controlCode** follow the same rules as for enterTicket.
- **textType** equals one of two possible values:
  - R = normal text line.
  - S = Seal text line of which there can only be one per box office ticket.
- **Text** represents the text to be printed (up to 46 characters, further characters are truncated).

5.10.5 printTicketVoid

Cancels an open box office ticket. The printer remains in box office ticket mode.

```
<printTicketVoid  randomNum="" controlCode="" />
```

The attributes follow the same rules as for enterTicket.

5.10.6 printBarCode

The printing of barcodes and QR codes in box office tickets uses the same commands and rules as described in the printerFiscalReceipt section (printBarCode) . However, unlike a fiscal receipt or commercial document, there is no limit to the number of barcodes that can be printed in a box office ticket. Barcodes are printed immediately. The operator, randomNum and controlCode attributes are not required.

5.10.7 endTicket

Closes a box office ticket.

```
<endTicket  randomNum="" controlCode="" />
```

The attributes follow the same rules as for enterTicket.

5.10.8 exitTicket

Switches out of box office ticket mode to fiscal receipt / commercial document mode.

```
<exitTicket  randomNum="" controlCode="" />
```

The attributes follow the same rules as for enterTicket.

5.11 ePOS-Print

Please see ePOS documentation since the XML format is almost the same. The main difference is the name of the web service CGI file where the request is forwarded to:

- ePOS TM: service.cgi
- ePOS fiscal EpsonFPMate: fpmate.cgi

5.12 Aliases for Backward Compatibility

In order to aid the migration from the Windows version of EpsonFPMate to the web service variant, the sub-element and attribute naming conventions can be maintained. Minor differences are explained. Note that directIO has a completely new format since the Active X / OCX software layer is not part of the fpmate.cgi code. The following tables illustrate the relative aliases:

Sub-elements:

N	Web Service Name	Windows EpsonFpMate Alias
1	printRecVoid	printerVoidReceipt
2	printRecItemVoid	printRecVoidItem
3	printFiscalDocument	beginFiscalDocument (type 1)
4	printXZReport	printZTotReport
5	queryContentByDate	queryEjContent
6	queryContentByNumbers	queryEjContentByNumbers

Attributes:

N	Web Service Name	Windows EpsonFpMate Alias
1	operator	Ope
2	message	Text (printRecMessage sub-element)
3	description	Text (Sale related sub-elements) PaymentDescription
4	code	Text (printBarCode sub-element)
5	data	Text (printNormal and displayText sub-elements)
6	documentLine	Text (printFiscalDocumentLine sub-element)
7	quantity	Qty
8	unitPrice	UnitCost
9	department	Dep
10	justification	Just
11	adjustmentType	Type
12	messageType	Type
13	paymentType	Type
14	mode	Type
15	option	Type
16	amount	Amount
17	index	Index Num
18	font	Font

19	hRIFont	Font
		In addition the following value aliases exist:
		A = FontA
		B = FontB
		C = FontC

N	Web Service Name	Windows EpsonFpMate Alias
20	Position	Pos
21	width	Width
22	height	Height
23	hRIPosition	Hri In addition the following value aliases exist: 0 = DISABLED 1 = BELOW 2 = ABOVE 3 = TWICE
24	codeType	tipo
25	documentType	Document (1 = Invoice)
26	documentNumber	Number

6 Details of Fiscal ePOS-Print XML (Responses and Errors)

6.1 Description

EpsonFPMate responses contain a minimum of three parameters and depending on the command may contain additional information. The first part of the response contains a <response element tag. The following three attributes can be found in this tag:

N	Attribute	Description
1	success	true or false.
2	code	String description. Usually empty; only filled during error conditions.
3	status	Numeric identifier that follows the Epson ASB status codes as much as possible.

If additional information is to be provided, the <addInfo> sub-element tag will be present. It has no attributes or data. It has as its first sub-element an <elementList> tag which contains as data a comma delimited list of the remaining element tag names. elementList has no attributes. These remaining elements contain data and generally have no attributes. Below is a successful fiscal receipt or commercial document response example (sent after having processed endFiscalReceipt):

```
<response success="true" code="" status="2">
  <addInfo>
    <elementList>lastCommand,printerStatus,fiscalReceiptNumber,fiscalReceiptAmount,fiscalReceiptDate,
fiscalReceiptTime</elementList>
    <lastCommand>74</lastCommand>
    <printerStatus>00010</printerStatus>
    <fiscalReceiptNumber>2</fiscalReceiptNumber>
    <fiscalReceiptAmount>1,00</fiscalReceiptAmount>
    <fiscalReceiptDate>26/02/2018</fiscalReceiptDate>
    <fiscalReceiptTime>12:13</fiscalReceiptTime>
  </addInfo>
</response>
```

The library presents the responses in an onreceive function containing three arrays and the XML:

N	Array name	Description
1	res	Contains success, code and status
2	tag_list_names	Contains the additional info parameter list
3	add_info	Contains the additional info parameters in a hash array. For example use add_info.printerStatus to determine printer status.
4	res_add	Contains all the XML

Please see the html/JavaScript example files for reference.

6.2 Success, Code and Status Cross Reference Table

Success	Code	Status	Comments
True	Always null	2 or TM ABS bitmap	Successful elaboration.
False	NO_DATA	Always = 0	Empty HTTP POST SOAP request.
	NO_RAM		Printer out of RAM memory.
	PARSER_ERROR		Malformed XML file.
	LAN_ERROR		Fpmate embedded fiscal firmware intercommunication timeout during initial communication phase (before XML line processing).
	LAN_TIME_OUT		Fpmate embedded fiscal firmware intercommunication timeout during XML line processing.
	FP_NO_ANSWER		Firmware fiscal did not respond to Fpmate embedded command within timeout.
	TM_NO_ANSWER/OFF_LINE		TM "Connection reset by peer" error or timeout.
	CONFIGURATION_FILE_ERROR		Empty or non-existent secondary devices XML configuration file.
	INCOMPLETE FILE		XML file does not contain the minimum elements necessary (missing endFiscalReceipt for example).
	Non valid XML command		Unknown or misplaced sub-element (endNonFiscal in an invoice for example).
False	EPTR_REC_EMPTY	Always 3	Problem due to cover open, printer offline or no paper roll loaded.
False	PRINTER ERROR	Native fiscal protocol error code (except codes 2 and 3)	Problem due to fiscal printer at the native communication level. For example discount amount = 0. For a complete list of possible errors, please refer to the "Communication Protocol" document and the "A.PDU Errors" chapter.
False	EFT_POS_ERROR	Always = 38	Electronic payment error.
False	FP_NO_ANSWER_NETWORK	Always = 0	JavaScript library ontimeout event or HTTP error. For example: HTTP 404 – No page found or network error.
False	incomplete XML commands	Always = 0	Deprecated.

Success	Code	Status	Comments
False	TM_OFF_LINE	0x80000001	TM printer out of paper for example.
	GENERIC		For future use

6.3 printerStatus and fpStatus

The fiscalprint.js JavaScript library contains a commented out decodeFpStatus function that could be used to decode the five byte printerStatus / fpStatus codes into a human readable format.

The "Communication Protocol" document also contains a table that indicates the meaning of each possible value. It can be found under the section entitled "H1=1; H2=074 – GET PRINTER STATUS".

6.4 Parsing Errors

EpsonFPMate performs XML validation checks and will respond immediately a parser error is detected. Below is an example response:

```
<response success="false" code="PARSER ERROR" status="0">
</response>
```

6.5 Command Errors

EpsonFPMate performs command validation checks and will respond immediately a command error is detected. A typical error could be an illegal quantity or amount. Below is an example response when quantity is zero:

```
<response success="false" code="PRINTER ERROR" status="21">
        <addInfo>
                <elementList>lastCommand,printerStatus</elementList>
                <lastCommand>80</lastCommand>
                <printerStatus>00100</printerStatus>
        </addInfo>
</response>
```

If a sub-element name is misspelt or a sub-element relevant in one type of file is placed in a different type of file where it is not supported, the non valid XML command response is returned (without additional information and status 0):

```
<response success="false" code="non valid XML command" status="0">
</response>
```

6.6 Incomplete or empty files

EpsonFPMate returns an "INCOMPLETE FILE" response message if endFiscalReceipt is missing for example:

```
<response success="false" code=" INCOMPLETE FILE" status="0">
</response>
```

6.7 Secondary Printer Errors

In addition to the three errors above, if the printer where fpmate.cgi resides is unable to communicate with the secondary printer, an appropriate error is returned. Below is an example response:

```
<response success="false" code="LAN ERROR" status="0">
</response>
```

6.8 Fpmate Timeout

The fiscal firmware did not respond to the Fpmate embedded command within the timeout period. Could be due to a long print-out such as bulk reprinting.

```
<response success="false" code="FP_NO_ANSWER" status="0">
</response>
```

This error can be simulated by activating the XON-XOFF protocol on the fiscal printer (SET 14/35).

6.9 XMLHTTP Request Timeouts

If no response arrives from the fpmate.cgi, an onerror function is called. Each browser responds in its own way with its own timeout setting and xhr.status value. The table below illustrates some major browsers behaviour.

N	Browser	Timeout	xhr.status
1	Internet Explorer 8	40 seconds	12029
2	Mozilla Firefox	15 seconds	0
3	iPad Safari	75 seconds	0

Software developers can still use their own timers if necessary setting the timeout argument in the send method. Note that the XMLHTTP request protocol foresees an "ontimeout" function but not all browsers currently support it. Below is an example response:

```
<response success="false" code="FP_NO_ANSWER_NETWORK" status="0">
</response>
```

This error can be simulated by disabling the Ethernet port on the fiscal printer for example (SET 19 CONNESSIONE set to NON CONNESSO).

6.10 Responses when Printer Offline

If the cover is open or no paper is in the printer, EpsonFPMate returns an appropriate error. Below is an example response:

```
<response success="false" code="EPTR_REC_EMPTY" status="3">
        <addInfo>
                <lastCommand>88</lastCommand>
                <printerStatus>30110</printerStatus>
        </addInfo>
</response>
```

6.11 Printer Power Supply Interruption

Power supply interruption whilst there are no EpsonFPMate requests for the printer	**Power supply interruption during EpsonFPMate communication exchanges**
No effect. After power is restored to the printer the system is ready as before.	The EpsonFPMate CGI executes on the fly using static memory and cannot recover from a power failure. Regardless of whether the EpsonFPMate CGI is used or not, if local printing was in progress, the fiscal printer upon restoration performs the following: • Cuts the paper. • Prints a power interruption message. • Prints again up to the same point. This behaviour is not suitable when using this library. It is only suitable when employing the previous generation of Epson drivers or the POS keyboard. Most probably, no response will arrive at the browser and the XMLHTTP request will timeout. Since in most cases the EpsonFPMate CGI automatically performs a printer reset when a request arrives, it is only necessary to send the same request again after the timeout has elapsed. The original print-out will be cancelled and will not be added to the daily totals.

7 Fiscal-ePOS-Print API

The JavaScript library (filename fiscalprint.js) provides an interface for communicating with intelligent fiscal printers. The library has the following key components:

Component	Code	Description
Constructors	epson.fiscalPrint	Initializes an ePOS-Print XML Builder object
Methods	send	Sends a message adding SOAP header
Properties	onreceive	Response message receipt event
	onerror	Communication error event

7.1 Practical Usage

- The first step is to insert the script line at the head of the HTML page:

```
<script  type="text/javascript" src="fiscalprint.js"></script>
```

- Let’s create a queryPrinterStatus request. Firstly, we place the request in a variable:

```
var data_to_send = '<printerCommand>\n' +
          '\t<queryPrinterStatus  operator="1" />\n' +
          '</printerCommand>\n';
```

- Instantiate the fiscal request:

```
var epos = new epson.fiscalPrint();
```

- Place the onreceive code first:

```
epos.onreceive = function (res, tag_list_names, add_info, res_add) {
```

Place any code here to manage the responses.

```
        } // end onReceive
```

- Place the onerror code after:

```
        epos.onerror = function () {
```

Place any code here to manage errors. XMLHTTP request timeouts fire this event as well as HTTP errors. Printer errors on the other hand use onreceive.

```
        } // end onError
```

- Now we can send the request:

```
epos.send(window.location.protocol + "//" + window.location.hostname + "/cgi-bin/fpmate.cgi",
data_to_send, timeout);
```

or with query string:

```
epos.send(window.location.protocol + "//" + window.location.hostname + "/cgi-bin/fpmate.cgi"
 + "?devid=cucina&timeout=10000", data_to_send, timeout);
```

8 UTF-8 Encoding and Character Escaping

The JavaScript and/or HTML files should be UTF-8 encoded since each request will have the following first line automatically inserted by the library:

```
<?xml version="1.0" encoding="utf-8" ?>
```

Italian Epson printers including fiscal variants use OEM codepage-437. The UTF-8 and code page standard ASCII characters are the same. However, non-standard characters such as grave accent “è” are represented by different codes. EpsonFPMate automatically converts the following non-standard characters from UTF-8 to OEM CP-437:

Ä ä à â
Ç ç
É è é ê ë
ì ï î
Ö ò ô ö
Ü ù ü û
ÿ
£ *
°
€ *

* The pound sign in certain cases automatically displays or prints the Euro symbol.

** The Euro symbol can only be used with displayText on MF fiscal printers as it is not printable. With RT fiscal printers, the Euro symbol can be both printed and displayed. In all cases the fiscal printer SET 15/16 value must be set to the default of 156.

The following table describes the different scenarios:

Request	**MF**	**RT**
Euro symbol in displayText	Displays €	Displays €
Pound symbol in displayText	Displays €	Displays €
Euro symbol in printRecItem etc.	Displays £ Prints £	Displays € Prints €
Pound symbol in printRecItem etc.	Displays £ Prints £	Displays € Prints €
Euro symbol in queryContent…	Version <=1.026 returns £ Version >=1.028 returns €	Returns €
Pound symbol in queryContent…	Version <=1.026 returns £ Version >=1.028 returns €	Returns €

The use of other non-standard characters could have undesirable effects.

In addition, the following characters have to be represented with the appropriate HTML codes:

Character	HTML Escape Code
<	&lt;
>	&gt;
&	&amp;
"	&quot;
'	&apos;

9 Automatic Programming

Before processing certain types of XML file, EpsonFpMate reads and sets some fiscal printer parameters.

9.1 printerFiscalReceipt, printerNonFiscal, printerFiscalReport and printerFiscalDocument File Types

The following sequence is used:

1. Sends reset command (closes any open document)
2. Reads SET14/29 JAVAPOS-UPOS flag
3. If JAVAPOS-UPOS mode is deactivated (value zero), it activates it
4. Programs SET 14/11 flag to 1 (paper low warning on request)
5. Programs SET 14/27 flag to 0 (ACK off)
6. Processes XML file
7. if the original JAVAPOS-UPOS mode was deactivated, EpsonFpMate restores the deactivated setting (Flag SET14/29 set to zero)

9.2 printerCommand File Type

1. Programs SET 14/11 flag to 1 (paper low warning on request)
2. Programs SET 14/27 flag to 0 (ACK off)
3. Processes XML file

9.3 printerTicket

No preprograming is performed.

10 XML Examples

10.1 Fiscal Receipt or Commercial Document

```
<printerFiscalReceipt>
    <displayText operator="1" data="Message on         customer display" />
    <printRecMessage operator="1" messageType="1" index="1" font="1" message="First Additional Header Row Type
    1" />
    <printRecMessage operator="1" messageType="1" index="2" font="1" message="Second Additional Header Row
    Type 1" />
    <beginFiscalReceipt operator="1" />
    <printRecMessage operator="1" messageType="4" message="First Additional Row Type 4" />
    <printRecItem operator="1" description="PANINO" quantity="1" unitPrice="6,00" department="15" justification="1"
    />
    <printRecMessage operator="1" messageType="4" message="Second Additional Row Type 4" />
    <printRecItem operator="1" description="Selling Item 2 VAT 20%" quantity="1,234" unitPrice="10,00"
    department="15" justification="1" />
    <printRecItemVoid operator="1" description="Void selling Item 2" quantity="1,234" unitPrice="10,00"
    department="15" justification="1" />
    <printRecItem operator="1" description="Selling Item 3 VAT 20%" quantity="2,50" unitPrice="100,17"
    department="3" justification="1" />
    <printRecMessage operator="1" messageType="4" message="Third Additional Row Type 4" />
    <printRecMessage operator="1" messageType="7" message="123456789" comment="Customer Id" />
    <printRecRefund operator="1" description="Refund (goods return)" quantity="10" unitPrice="5,00"
    department="15" justification="1" />
    <printRecItem operator="1" description="Selling Item 4 VAT 20%" quantity="12,13" unitPrice="216,17"
    department="3" justification="2" />
    <printRecItemAdjustment operator="1" description="Discount applied to the product" adjustmentType="0"
    amount="123,45" justification="2" />
    <printRecMessage operator="1" message="First Additional Row Type 2" messageType="2" index="1" font="1" />
    <printRecMessage operator="1" message="Second Additional Row Type 2" messageType="2" index="2" font="1" />
    <printRecMessage operator="1" message="First Additional Row Type 3" messageType="3" index="1" font="1" />
    <printRecMessage operator="1" message="Second Additional Row Type 3" messageType="3" index="2" font="1" />
    <printRecItem operator="1" description="Selling Item 5 20%" quantity="12,13" unitPrice="216,17" department="3"
    justification="2" />
    <printRecSubtotalAdjustment operator="1" description="Discount applied to the subtotal" adjustmentType="1"
    amount="300,12" justification="2" />
    <printRecSubtotal operator="1" option="1" />
    <printBarCode operator="1" position="10" width="2" height="66" hRIPosition="3" hRIFont="C" codeType="CODE39"
    code="0123456789" />
    <printRecTotal operator="1" description="Payment in cash" payment="6000,00" paymentType="0" index="0"
    justification="2" />
    <displayText operator="1" data="Message on         customer display" />
    <endFiscalReceipt operator="1" />
  </printerFiscalReceipt>
```

10.2 Credit Note Fiscal Receipt (MF models only)

```
<printerFiscalReceipt>
      <printRecMessage  operator="2" message="PRATICA DI RESO 1/2018" messageType="4" />
      <printRecRefund  operator="2" description="Refund 1 (goods return)" quantity="10,25" unitPrice="5,00"
      department="1" justification="1" />
      <printRecRefund  operator="2" description="Refund 2 (goods return)" quantity="13,5" unitPrice="6,00"
      department="2" justification="1" />
      <printRecRefund  operator="2" description="Refund 3 (goods return)" quantity="16" unitPrice="7,00"
      department="3" justification="1" />
      <printRecItemAdjustment  operator="2" description="Discount applied to the product" adjustmentType="0"
      amount="1,23" justification="2" />
      <printRecMessage  operator="2" message="First Additional Row Type 2" messageType="2" index="1" font="1" />
      <printRecMessage  operator="2" message="Second Additional Row Type 2" messageType="2" index="2" font="1" />
      <printRecMessage  operator="2" message="First Additional Row Type 3" messageType="3" index="1" font="1" />
      <printRecMessage  operator="2" message="Second Additional Row Type 3" messageType="3" index="2" font="1" />
      <printRecSubtotal  operator="2" option="1" />
      <printBarCode  operator="2" position="10" width="2" height="66" hRIPosition="1" hRIFont="A" codeType="CODE93"
      code="0123456789ABCDEFG" />
      <printRecTotal  operator="2" />
      <displayText  operator="2" data="Arrivederci e grazie" />
      <endFiscalReceipt  operator="2" />
</printerFiscalReceipt>
```

10.3 Z Report

```
<printerFiscalReport>
      <displayText  operator="1" data="Message on         customer display" />
      <printZReport  operator="1"  timeout="30" />
</printerFiscalReport>
```

10.4 Non Fiscal Receipt or Administrative Document with Bar Codes

```
<printerNonFiscal>
      <displayText  operator="1" data="Message on         customer display" />
      <beginNonFiscal  operator="1" />
      <printNormal  operator="1" font="1" data="Non Fiscal Receipt Row N. 1 Font 1" />
      <printNormal  operator="1" font="2" data="Non Fiscal Receipt Row N. 2 Font 2" />
      <printNormal  operator="1" font="3" data="Non Fiscal Receipt Row N. 3 Font 3" />
      <printBarCode  operator="1" position="10" width="2" height="66" hRIPosition="2" hRIFont="A" codeType="CODE39"
      code="0123456789" />
      <printNormal  operator="1" font="4" data="Non Fiscal Receipt Row N. 4 Font 4" />
      <displayText  operator="1" data=" Message on        customer display " />
      <printNormal  operator="1" font="1" data="Non Fiscal Receipt Row N. 5 Font 1" />
      <printNormal  operator="1" font="2" data="Non Fiscal Receipt Row N. 6 Font 2" />
      <printNormal  operator="1" font="3" data="Non Fiscal Receipt Row N. 7 Font 3" />
      <printBarCode  operator="1" position="10" width="2" height="66" hRIPosition="1" hRIFont="B" codeType="CODE128"
      code="{A0123456789" />
      <printNormal  operator="1" font="4" data="Non Fiscal Receipt Row N. 8 Font 4" />
      <displayText  operator="1" data=" Message on         customer display" />
      <endNonFiscal  operator="1" />
  </printerNonFiscal>
```

10.5 Direct IO Command

```
<printerCommand>
      <directIO  command="4005" data="011750" comment="Set VAT group 1 to 17.5%" />
</printerCommand>
```

10.6 Fiscal Receipt or Commercial Document with Graphic Coupon

```
<printerFiscalReceipt>
    <printRecMessage  operator="1" message="First Additional Header Row Type 1" messageType="1" index="1" font="1" />
    <beginFiscalReceipt  operator="1" />
    <printRecItem  operator="1" description="Selling Item 1 VAT 10%" quantity="1" unitPrice="100,00" department="15" justification="1" />
    <printRecMessage  operator="1" message="Second Additional Row Type 4" messageType="4" index="1" />
    <printRecItem  operator="1" description="Selling Item 2 VAT 20%" quantity="1,234" unitPrice="100,00" department="15" justification="1" />
    <printRecMessage  operator="1" message="First Additional Row Type 2" messageType="2" index="1" font="1" />
    <printRecMessage  operator="1" message="First Additional Row Type 3" messageType="3" index="1" font="1" />
    <printRecItem  operator="3" description="Selling Item 5 20%" quantity="12,13" unitPrice="216,17" department="3" justification="2" />
    <printRecSubtotalAdjustment  operator="1" description="Discount applied to the subtotal" adjustmentType="0" amount="300,12" department="3" justification="2" />
    <printRecSubtotal  operator="3" option="1" />
    <printBarCode  operator="1" position="10" width="2" height="66" hRIPosition="3" hRIFont="A" codeType="CODE39" code="01234567ABCDEF" />
    <printRecTotal  operator="1" description="Payment in cash" payment="3000,00" paymentType="0" index="0" justification="2" />
    <printGraphicCoupon  operator="1" graphicFormat="R" >kjhaskjsfdkjfsksadkjh/=</printGraphicCoupon>
    <displayText  operator="4" data="Customer Display    Printed Fisc Receipt" />
    <endFiscalReceipt  operator="1" />
</printerFiscalReceipt>
```

10.7 Non Fiscal Receipt or Administrative Document with Bar Codes

```
<printerNonFiscal>
    <beginNonFiscal  operator="1" />
    <printNormal  operator="1" font="1" data="Non Fiscal Receipt Row N. 1 Font 1" />
    <printNormal  operator="1" font="1" data="" comment="Add blank line (whitespace)" />
    <printNormal  operator="1" font="1" data="Barcode System CODE39" />
    <printBarCode  operator="1" position="10" width="2" height="66" hRI="2" hRIFont="A" codeType="CODE39" code="0123456789ABCD" />
    <printNormal  operator="1" font="1" data="" comment="Add blank line (whitespace)" />
    <printNormal  operator="1" font="2" data="Non Fiscal Receipt Row N. 2 Font 2" />
    <printNormal  operator="1" font="3" data="Non Fiscal Receipt Row N. 3 Font 3" />
    <printNormal  operator="1" font="1" data="" comment="Add blank line (whitespace)" />
    <printNormal  operator="1" font="1" data="Barcode System CODE93" />
    <printBarCode  operator="1" position="10" width="2" height="66" hRIPosition="1" hRIFont="A" codeType="CODE93" code="0123456789ABCD" />
    <printNormal  operator="1" font="1" data="" comment="Add blank line (whitespace)" />
    <printNormal  operator="1" font="4" data="Non Fiscal Receipt Row N. 4 Font 4" />
    <printNormal  operator="1" font="1" data="Non Fiscal Receipt Row N. 5 Font 1" />
    <printNormal  operator="1" font="2" data="Non Fiscal Receipt Row N. 6 Font 2" />
    <printNormal  operator="1" font="1" data="" comment="Add blank line (whitespace)" />
    <printNormal  operator="1" font="1" data="Barcode System CODE128 CODE A" />
    <printBarCode  operator="1" position="10" width="2" height="66" hRIPosition="0" hRIFont="A" codeType="CODE128" code="{A0123456789ABCD" />
    <printNormal  operator="1" font="1" data="" comment="Add blank line (whitespace)" />
    <printNormal  operator="1" font="3" data="Non Fiscal Receipt Row N. 7 Font 3" />
    <printNormal  operator="1" font="1" data="" comment="Add blank line (whitespace)" />
    <printNormal  operator="1" font="1" data="Barcode System CODE128 CODE B" />
    <printBarCode  operator="1" position="10" width="2" height="66" hRIPosition="1" hRIFont="A" codeType="CODE128" code="{B0123456789ABCD" />
    <printNormal  operator="1" font="1" data="" comment="Add blank line (whitespace)" />
    <printNormal  operator="1" font="4" data="Non Fiscal Receipt Row N. 8 Font 4" />
    <displayText  operator="4" data="Customer Display Printed Non Fisc Doc" />
    <printNormal  operator="1" font="1" data="" />
    <printNormal  operator="1" font="1" data="Barcode System CODE128 CODE C" />
    <printBarCode  operator="1" position="10" width="2" height="66" hRIPosition="3" hRIFont="A" codeType="CODE128" code="{C0123456789ABCD" />
    <displayText  operator="4" data="Customer Display    Printed Non Fisc Doc" />
    <endNonFiscal  operator="1" />
</printerNonFiscal>
```

10.8 Non Fiscal Receipt or Administrative Document with Graphic Coupon

```
<printerNonFiscal>
      <beginNonFiscal  operator="1" />
      <printNormal  operator="1" font="1" data="COUPON PRINTING ..." />
      <printGraphicCoupon  operator="1" graphicFormat="B" >askjdhasdjhkjasd/=</printGraphicCoupon>
      <printNormal  operator="1" font="1" data=" COUPON PRINTED" />
      <endNonFiscal  operator="1" />
</printerNonFiscal>
```

10.9 Free Fiscal Document (Free Invoice not SLIP) (MF models only)

Note that free invoices are supported from firmware version 3.00D.

```
<printerFiscalDocument>
  <beginFiscalDocument  operator="1" documentAmount="1234,56" documentType="freeInvoice" />
  <printFiscalDocumentLine  operator="1" font="1" documentLine="Invoice Text row 1" />
  <printFiscalDocumentLine  operator="1" font="1" documentLine="Invoice Text row 2" />
  <printFiscalDocumentLine  operator="1" font="1" documentLine="Invoice Text row 3" />
  <printFiscalDocumentLine  operator="1" font="1" documentLine="Invoice Text row 4" />
  <printFiscalDocumentLine  operator="1" font="1" documentLine="Invoice Text row 5" />
  <printFiscalDocumentLine  operator="1" font="1" documentLine="Invoice Text row 6" />
  <printFiscalDocumentLine  operator="1" font="1" documentLine="Invoice Text row 7" />
  <endFiscalDocument  operator="1" operationType="0" />
</printerFiscalDocument>
```

10.10 Fiscal Document Based on a Fiscal Receipt (MF models only)

Note that last receipt invoices are supported from firmware version 3.00D.

```
<printerFiscalDocument>
      <printRecMessage  operator="1" message="First Add Header Row Type 5" messageType="5" index="1" />
      <printRecMessage  operator="1" message="First Add Header Row Type 5" messageType="5" index="2" />
      <printRecMessage  operator="1" message="First Client Row Type 6" messageType="6" index="1" />
      <printRecMessage  operator="1" message="Second Client Row Type 6" messageType="6" index="2" />
      <beginFiscalDocument  operator="1" documentType="lastReceiptInvoice" documentNumber="0" />
      <displayText  operator="1" data="Customer Display    Printed Last Rec Fis" />
</printerFiscalDocument>
```

10.11 Direct Fiscal Document

```
<printerFiscalDocument>
      <printRecMessage  operator="1" message="First Add Header Row Type 5" messageType="5" index="1" font="1" />
      <printRecMessage  operator="1" message="First Add Header Row Type 5" messageType="5" index="2" font="1" />
      <printRecMessage  operator="1" message="First Client Row Type 6" messageType="6" index="1" font="1" />
      <printRecMessage  operator="1" message="Second Client Row Type 6" messageType="6" index="2" font="1" />
      <beginFiscalDocument  operator="1" documentType="directInvoice" documentNumber="0" />
      <printRecItem  operator="1" description="Selling Item 1 20%" quantity="22,11" unitPrice="1,333" department="3"
      justification="2" />
      <printRecTotal  operator="1" description="Payment by cheque" payment="300,00" paymentType="1"
      justification="2" />
      <displayText  operator="1" data="Customer Display    Printed Dir Fisc" />
      <endFiscalDocument  operator="1" />
</printerFiscalDocument>
```

10.12 printContentByDate

```
<printerCommand>
   <printContentByDate  operator="1" dataType="0" fromDay="25" fromMonth="2" fromYear="2018" toDay="25"
      toMonth="2" toYear="2018" />
</printerCommand>
```

10.13 printContentByNumbers

```
<printerCommand>
   <printContentByNumbers operator="1" dataType="0" day="25" month="7" year="13" fromNumber="9" toNumber="9" />
</printerCommand>
```

10.14 MF Printer Status or RT Basic Status Query

```
<printerCommand>
   <queryPrinterStatus operator="1" statusType="0" />
</printerCommand>
```

10.15 RT Printer RT Status Query

```
<printerCommand>
   <queryPrinterStatus operator="1" statusType="1" />
</printerCommand>
```

10.16 Display Text Message

```
<printerCommand>
   <displayText operator="1" data="00 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A" />
</printerCommand>
```

10.17 Non-fiscal ePOS-print

```
<epos-print>
   <text dw="0" dh="0" description="Coupondes" />
   <feed type="1" />
   <feed type="1" />
   <cut type="feed" />
</epos-print>
```

10.18 setLogo in Non-fiscal Receipt or Administrative Document

Logo uploaded, programmed in all four positions then set differently for receipt footer

```
<printerNonFiscal>
      <beginNonFiscal operator="1" />
      <printNormal operator="1" font="1" data="LOGO UPLOAD AND PROGRAMMING ..." />
      <setLogo operator="1" location="0" index="1" option="0" graphicFormat="B" >123klkjaskdj/=</setLogo>
      <setLogo location="2" index="2" option="2" comment="Set receipt footer to existing index 2" />
      <printNormal operator="1" font="1" data="LOG UPLOAD AND PROGRAMMING COMPLETE" />
      <endNonFiscal operator="1" />
</printerNonFiscal>
```

10.19 setLogo with printerCommand

```
<printerCommand>
      <setLogo location="3" index="3" comment="operator, option and graphicFormat attributes not necessary" />
 </printerCommand>
```

10.20 RT Query Document Status

```
<printerCommand>
      <directIO  command="9205" data="299SEA0000061202180888003300000000102" timeout="6000" />
 </printerCommand>
```

10.21 RT Commercial Refund Document

```
<printerFiscalReceipt>
      <printRecMessage  operator="1" message="RESO MERCE N.0279-0010 del 08-03-2018" messageType="4"
      <beginFiscalReceipt  operator="1" />
      <printRecItem  operator="1" description="TV" quantity="1" unitPrice="600,00" department="15" justification="1" />
      <printRecTotal  operator="1" description="Payment refunded" payment="6000,00" paymentType="0" index="0"
      justification="2" />
      <endFiscalReceipt  operator="1" />
        </printerFiscalReceipt>
```

10.22 RT Commercial Void Document (Automatic Print-out)

```
<printerCommand>
      <directIO  command="1078" data="0140001ANNULLAMENTO N.0279-0010 del 08-03-2018        "  timeout="10000" />
</printerCommand>
```

10.23 RT Commercial Void Document (Manual Print-out)

```
<printerCommand>
      <printRecMessage  operator="1" message="ANNULLAMENTO N.0279-0010 del 08-03-2018" messageType="4"  />
      <beginFiscalReceipt  operator="1" />
      <printRecItem  operator="1" description="DVD" quantity="1" unitPrice="12" department="15" justification="1" />
      <printRecTotal  operator="1" description="Payment cancelled" payment="6000,00" paymentType="0" index="0"
      justification="2" />
      <endFiscalReceipt  operator="1" />
</printerCommand>
```