



# **Description of XML communication protocol**

	1
1. Read information about device readiness for data processing (DLE).....	5
2. Read information about logical state of the device (ENQ).....	5
3. Starting invoice.....	6
4. Invoice cancellation.....	7
5. Closing the invoice.....	7
5.1. Discount / markup to the whole invoice.....	7
5.2. Additional non-fiscal line.....	8
5.3. Definition of an invoice footer.....	8
6. Invoice examples:.....	8
7. Opening a receipt.....	8
8. Receipt cancellation.....	9
9. Closing receipt.....	9
9.1. Discount / markup to the whole receipt.....	9
9.2. Additional non-fiscal line.....	9
9.3. Definition of receipt footer.....	10
10. Payment in advance.....	10
11. Forms of payment.....	10
12. Adding position to receipt / invoice.....	11
13. Settlement of returnable packages.....	12
14. Giving discounts / markups.....	12
15. Non-fiscal printout – examples.....	13
16. Information about date and version of device.....	13
17. Information from ECR.....	13
18. Information about fiscal memory.....	14
19. Information of SD card.....	15
20. Request about state of current transaction.....	15
21. Request about state of active or the last transaction.....	15
22. Information about drawer status.....	16
23. Information about device – general information.....	17
24. Information about device – detailed for fiscal memory.....	17
25. Information about device – details concerning power condition, accumulator and battery.....	18
26. Information about device – information about printout status.....	18
27. Information about device – information about drawer status.....	19
28. Information about device – information about data base.....	19
29. Information about device – status of accumulator and power supply.....	19
30. Information about communication port.....	20
31. Information about dispatch to the repository.....	20
32. Controlling instructions.....	20
32.1. Sound signal.....	20
32.2. Paper feeding of quantity of lines.....	20
32.3. Opening drawer.....	20
32.4. Showing text on display.....	21
32.5. Display controlling in post-receipt mode.....	21
32.6. Display clearing.....	21
32.7. Access to the printer's menu.....	21
33. Daily report.....	21
34. Periodical report.....	22
35. Shift report.....	22
36. Report of data from protected memory.....	23
37. Events report.....	23
38. Register report in fiscal memory.....	23
39. Report of service reviews.....	23
40. Cashier log-in and log-off.....	23
41. Data readout of currently log-in cashier.....	24
42. Pay-in, pay-out and cash balance in ECR.....	24
43. Mode of errors handling.....	24
44. Control of goods base.....	24
45. USB configuration.....	25
46. Configuration.....	25

47.	Setting date and hour. ....	26
48.	Programming an online header .....	26
49.	Online approval (fiscalisation) .....	27
50.	Programming producer's data .....	27
51.	Service lock. ....	27
52.	Readout of update logs. ....	28
53.	Readout of module logs.....	28
54.	Service review. ....	29
55.	Confirmation of making a service review.....	29
56.	Programming change of currency. ....	29
57.	Programming mode of printing currency conversion.....	30
58.	Setting PTU rates. ....	30
59.	Request of sending back data from fiscal memory. ....	31
59.1.	Type = „report” – Record of daily report. ....	31
59.2.	Type = „reset” – Resetting RAM memory.....	32
59.3.	Type = „tax_rates” – Programming VAT rates.....	32
59.4.	Type = „currency_set” – Record of setting a currency change. ....	32
59.5.	Type = „currency_change” – Record of currency change. ....	32
59.6.	Type = „cardclose” – Record of closing card.....	33
59.7.	Type = „end” – End of data.....	33
60.	Request of sending back logs from port. ....	33
61.	Non-fiscal printout. ....	33
62.	Programming a graphic. ....	34
63.	Programming servers' address for the printers with online system.....	35
64.	Programming an animation. ....	35
65.	Information about amount of printed paper. ....	37
66.	Information about number of drawer openings.....	37
67.	Example with multiple spaces following each other. ....	37
68.	Graphic programming – example. ....	37
	Graphic programming .....	37
69.	Exemplary non-fiscal printouts. ....	38
69.1.	Non-fiscal printout with QR code .....	38
69.2.	Non-fiscal printout with several lines, QR code and bar code.....	39
69.3.	Non-fiscal printout with empty line and underline.....	39
70.	Examples – receipts .....	39
70.1.	Payments.....	39
70.1.1.	Receipt with many items.....	39
70.1.2.	Goods sale.....	40
70.1.3.	Payment with change .....	40
70.1.4.	Payment by card.....	40
70.1.5.	Payment by card with change .....	40
70.1.6.	Supporting forms of payment – many forms of payment.....	41
70.1.7.	Supporting forms of payment – cancelling 3 from 4 payments .....	41
70.2.	Discounts / Markups .....	42
70.2.1.	Sale with quota discount on the position .....	42
70.2.2.	Sale with percentage markup on the position .....	42
70.2.3.	Sale with percentage discount on the position .....	43
70.2.4.	Sale with quota markup on the receipt .....	43
70.2.5.	Discounts on the position and on sub-total and on the receipt .....	43
70.2.6.	Percentage discount on goods in given PTU rate - A .....	43
70.2.7.	Quota markup on goods in given PTU rate – C .....	44
70.3.	Deposits.....	45
70.3.1.	Deposit return on the receipt .....	45
70.3.2.	Reverse (storno) of deposit collected on the receipt.....	45
70.3.3.	Collections and refunds of deposit on the receipt .....	45
70.4.	Reversals (Storno).....	45
70.4.1.	Cancelled receipt. ....	45
70.4.2.	Receipt with storno .....	46
70.4.3.	Receipt with storno after discount to sub-total .....	46
70.5.	Others .....	46
70.5.1.	Receipt with footer's lines - 3 lines sys no. QR code .....	46
70.5.2.	Receipt with additional non-fiscal part.....	46

70.5.3.	In advance payment .....	47
70.5.4.	In advance payment storno (reversal) .....	47
70.5.5.	Settlement of in-advance payment .....	47
70.5.6.	Reversal (Storno) of in advance payment settlement .....	48
71.	Invoice examples .....	48
71.1.	Invoice with 1 copy .....	48
71.2.	Invoice with many positions .....	48
71.3.	Invoice with percentage discount on the invoice .....	49
71.4.	Invoice with quota markup on the position .....	50
71.5.	Invoice with quota discount to the subtotal .....	50
71.6.	Invoice with different forms of payment .....	51
71.7.	Invoice with data and additional footer's lines .....	51
71.8.	Invoice with additional non-fiscal part .....	52
71.9.	Invoice options – Print the amount of pennies in words and do not print the recipient's data and the VAT INVOICE label .....	52
71.10.	Invoice cancellation .....	53

## Warning !

Details of each packet can be optionally completed with checksum (CRC32 standard). General information about CRC32 checksum can be found at:

[http://pl.wikipedia.org/wiki/Cykliczny\\_kod\\_nadmiarowy](http://pl.wikipedia.org/wiki/Cykliczny_kod_nadmiarowy).

Using the checksum makes sense wherever there may appear interference and distortion and transport protocol itself contains no error correction mechanisms. In the case of the fiscal printers checksums make sense for RS232 and USB communication and there is no need to apply them in the case of TCP / IP communication.

If more detailed information are needed, please contact technical support of NOVITUS.

Example of packet without the checksum:

```
<packet>  
<info action="transaction"/>  
</packet>
```

Example of the same packet with the checksum:

```
<packet crc="BB1E3EC8">  
<info action="transaction"/>  
</packet>
```

Communications buffer size is 5000 bytes. In case if the data packet exceeds this value, data must be divided into smaller packets and sent individually.

### 1. Read information about device readiness for data processing (DLE).

In response to the question:

```
<packet>  
<dle...pl/>  
</packet>
```

the printer sends back information in the following form:

```
<packet crc="0153C518">  
<dle_online="yes" no_paper="no" printer_error="no"/>  
</packet>
```

where:

online – when the printer is in online mode - „yes”, otherwise - „no”.

no\_paper – when no paper occurs - „yes”, otherwise - „no”.

printer\_error – if occurs problem with printing mechanism - „yes”, otherwise - „no”.

### 2. Read information about logical state of the device (ENQ).

In response to the question:

```
<packet>  
<enq...pl/>  
</packet>
```

the printer sends back information in the following form:

```
<packet crc="39B3018B">  
<enq_fiscal="yes" last_command_oK="no" intranction_mode="no" last_transaction_ok  
="yes"/>  
</packet>
```

where:

Fiscal – if the printer is in the fiscal mode - „yes”, if in the training mode - „no”.  
 Last\_command\_ok – if the last order was executed correctly - „yes”, otherwise - „no”.  
 Transaction\_mode – if the printer is in transaction mode - „yes”, otherwise - „no”.  
 Last\_transaction\_ok – when the last transaction was finalized correctly - „yes”, otherwise - „no”.

### 3. Starting invoice.

After sending the following packet the printer receives order of starting invoice:

```
<packet>
<invoice action="begin" number="120/2012"nip="" description="both" paymentname="cash"
  payment_date="dd-mm-yyyy"=10-10-2013 recipient="" issuer="" copies="1" margins="no"
  buyer's_name_options+"information"
  Seller_name_options="none" sign_area="yes" paid by cash"
  selldate="dd-mm-yyyy"="10-10-2013">
  <customer>data of recipient 1</customer>
  <option id="1" />
</invoice></packet>
```

where:

invoicetype – invoice type

1. invoice
2. pharmacy

action = "begin" – starting invoice.

number – invoice number (not required).

nip – Nip number of the buyer (not required).

description – parameter ignored

payment\_name – form of payment (not required).

Payment\_date – date of payment (format: day – month – year).

recipient – recipient (not required).

issuer – issuer (not required).

copies – amount of copies – defaulty 0 (not required) 255 – no copy printed.

margins – margins – takes values „yes” or „no” (not required). Option not used.

Signature\_area – space for signature – takes values „yes” or „no” (not required).

Customer\_name\_options – takes the following values:

1. customer\_name\_options = „info” – options of buyer name (for information only).
2. Customer\_name\_options = „all” – name and information block (default value), invoice packet can't contain option **signature\_area**
3. Customer\_name\_options = „none” – no information about buyer, please set **signature\_area="no"**.

Seller\_name\_options – takes the following values:

1. Seller\_name\_options = „info” – options of seller's name (for information only).
2. Seller\_name\_options = „all” – name and information block (default value), invoice packet can't contain option **signature\_area**
3. Seller\_name\_options = „none” – no information about seller, please set **signature\_area="no"**

paidlabel – informative inscription about form of payment (not required).

Sell\_date – date of sale (format: day– month – year, not required).

tag <customer> – additional line of invoice for receiver's data (not required).

tag <option> - id takes the following values (not required),

- 1 – omit the verbal description of amount to be paid,
- 2 – omit block of the gross amounts in tax settlement,
- 3 – bold label „buyer”.
- 4 – bold label „seller”,
- 5 – bold NIP number of buyer,
- 6 – bold NIP number of seller,
- 7 – print "description / symbol" label in the header of invoice before the invoice items,
- 8 – print position number in invoice item,
- 9 – label „to pay” before block of tax settlement,
- 10 – print amount of pennies in verbal form,
- 11 – don't print date of sale if it's the same like date of issuing the invoice,
- 12 – don't print seller's data,
- 13 – don't print descriptions to invoice items (information describing the content of the fields in invoice items),

14 – enabled payments (such as in a receipt),

15 – don't print receiver's data (doesn't include field with NIP (regon, pesel number)),

- 16 – print inscription „Paid by cash”,
- 17 – omit label „seller:”,
- 18 – omit label „ORIGINAL” (it operates only if there is no copy to print),
- 19 – print label „VAT INVOICE” (instead of label „INVOICE”),

Buyer\_label – buyer label:

1. buyer
2. recivier

additional\_info – additional information to the invoice.

#### 4. Invoice cancellation.

After sending the following packet, the printer receives order to cancel the invoice.

```
<packet>
<invoice action="cancel"></invoice>
</packet>
```

#### 5. Closing the invoice.

After sending the following packet, the printer receives order to end the invoice.

```
<packet>
<invoice action="close" buyer="Jan Jan" total="123.32" system_no="123" checkbox_no="1"
cashier="Jan">
</invoice>
</packet>
```

where:

buyer - buyer

total – amount to pay,

system\_no – system number (@ - if number is to be a QR – code, # - if number is to be a bar code),

checkbox\_no – checkbox number,

cashier – cashier name.

Function of closing invoice allows to call additional options, which should be placed in tag „<invoice>” responsible for closing invoice. Additional options are:

##### 5.1. Discount / markup to the whole invoice.

```
<packet>
<discount value="23%" name="Occasional" descid="1" action="discount"></discount>
</packet>
```

where:

value – value of given discount / markup,

name – name of given discount / markup (in case when the field „name” is completed, parameter „descid” is ignored, not required ),

Id description:

- 0: no description,
- 1: „occasional”,
- 2: „occasion”,
- 3: „festive”
- 4: „regular customer”,
- 5: „jubilee”,
- 6: „birthday”,
- 7: „for employee”,
- 8: „promotion”,
- 9: „reward”,
- 10: „sale”,
- 11: „discount”,
- 12: „seasonal”,
- 13: „night”,

14: "personnel"

action – takes the following values (not required):

1. action = „discount“ (default value),
2. action = „markup“.

### 5.2. Additional non-fiscal line.

```
<packet>  
<non-fiscal_line type="line" id="1" value="1234"></non-fiscal_line>  
</packet>
```

where:

type – default value is „line“ (not required),

value – value which is to be printed.

id – identifier of additional line. Detailed description of all identifiers together with characteristic is placed in Novitus communication protocol.

### 5.3. Definition of an invoice footer.

```
<packet>  
<non-fiscal_line type="definition">  
  <line>abc line 1</line>  
  <line>abc line 2</line>  
  <line>abc line 3</line>  
</non-fiscal_line>  
</packet>
```

#### Warning!

There is only one footer change possible between daily reports.

Example:

### 6. Invoice examples:

- 6.1. Invoice with 1 copy
- 6.2. Invoice with many positions
- 6.3. Invoice with percentage discount on invoice
- 6.4. Invoice with amount markup to position
- 6.5. Invoice with amount discount to sub-total
- 6.6. Invoice with different form of payment
- 6.7. Invoice with data and additional footer's lines
- 6.8. Invoice with additional non-fiscal part
- 6.9. Invoice options – **Print quantity of pennies in words** and **Don't print receiver's data** and **VAT INVOICE label**
- 6.10. Invoice cancelling

### 7. Opening a receipt.

After sending the following packet, the printer receives order of opening receipt.

```
<packet>  
<receipt action="begin" mode="online" pharmaceutical="no"></receipt>  
</packet>
```

where:

mode – takes the following values (not required):

1. mode = „online“ – mode of online receipt (default value)
2. mode = „offline“ – mode of offline receipt.



pharmaceutical – takes the following values:

1. pharmaceutical = „yes” – pharmacy receipt.
2. pharmaceutical = „no” – there will be printed an ordinary receipt (default value).

## 8. Receipt cancellation.

After sending the following packet, the printer receives order of receipt cancellation.

```
<packet>  
<receipt action="cancel"></receipt>  
</packet>
```

## 9. Closing receipt.

After sending the following packet, the printer receives order of closing the receipt.

```
<packet>  
<receipt action="close" system_no="123" checkbox_no="1" cashier="Jan" total="123.32"  
charge="10.00" nip="123-456-78-90">  
</receipt>  
</packet>
```

where:

system\_no – receipt number (@ - if it is to be a QR– code, # - if it is to be a barcode),  
checkbox – checkbox number (not required),  
cashier – cashier name (not required),  
total – checks out value of receipt calculated by the printer and program (not required).  
charge - amount to be paid by the customer. Option requested for pharmacy receipt

nip, regon, pesel – allow to enter buyer's data in the receipt footer.

Function of closing receipt allows to call the additional options, which should be placed in tag „<receipt>”, responsible for closing receipt. The additional options are:

### 9.1. Discount / markup to the whole receipt.

```
<packet>  
<discount value="23%" name="Occasional" desc_id="1" action="discount"></discount>  
</packet>
```

where:

value – value of given discount / markup,  
name – name of given discount / markup (in case when the field „name” is completed, parameter „desc\_id” is ignored, not required),  
action – takes the following values (not required):  
1. action = „discount” – (default value),  
2. action = „markup”.

## [Examples](#)

### 9.2. Additional non-fiscal line.

```
<packet>  
<non-fiscal_line type="line" id="1" value="1234"></non-fiscal_line>  
</packet>
```

where:

type – default value is „line” (not required),

value – value which is to be printed.

id – identifier of the additional line. Detailed description of all identifiers together with characteristic is placed in Novitus communication protocol.

### Examples

#### 9.3. Definition of receipt footer.

```
<packet>
  <non-fiscal_line type="definition">
    <line>abc line 1</line>
    <line>abc line 2</line>
    <line>abc line 3</line>
  </non-fiscal_line>
</packet>
```

#### Warning!

There is only one footer change possible between daily reports.

#### Example:

#### 10. Payment in advance.

To send payment in advance you should send a packet:

```
<packet>
<in_advances action="in_advances_payment" description="description" ptu="A" value="2"
surcharge="1"></in_advances>
</packet>
```

where:

action:

- In\_advance
- in\_advance\_settlement
- in\_advance STORNO / reversal
- stornosettlementrcpitemadvances – in\_advance\_STORNO/reversal\_settlement

description – in\_advance description

ptu – ptu/tax\_rate

value – value

surcharge – sum to pay after advance payment deduction

#### Warning!

Order should be sent after beginning or before ending the receipt.

#### 11. Forms of payment.

After sending the following packet, the printer receives order of choosing the forms of payment.

```
<packet>
  <payment type="cash" action="add" value="100.00" mode="payment" rate="" name="">
  </payment>
</packet>
```

where:

type – takes the following values:

1. type = „card”,

2. type = „cheque”,
3. type = „voucher”, (on printout „Token”),
4. type = „credit”,
5. type = „transfer”,
6. type = „customer\_account”,
7. type = „foreign\_currency”,
8. type = „cash”,
9. type = „mobile”
10. type = „token”, (on printout „ Voucher”),

value – value of payment,

rate – exchange rate in the foreign currency (not required, default takes value 1),

mode – takes the following values (not required),

1. mode = „payment” (default value),
2. mode = „rest” – rest used when it's necessary to display rest in the foreign currency,

name – currency name (not required).

### Warning!

If parameter rate is different than 1, then the parameter „value” is interpreted as value of payment in the foreign currency (value in the current currency is calculated as product of parameter „value” and „rate”). To pay in different forms of payment, you should use the payment packet after receipt positions and before its ending.

## 12. Adding position to receipt / invoice.

After sending the following packet, the printer receives order of adding position to the receipt / the invoice.

```
<packet>
  <item name="name" quantity="2" quantity_unit="szt" ptu="A" price="1.23" total="2.46"
    recipe="" charge="" plu="" description="" action="sale">
  </item>
</packet>
```

where:

name – name of added goods,

quantity – quantity of goods,

ptu – VAT rates,

price – price for unit,

total – calculated as product of parameter quantity and price (not required),

recipe – requested only in the pharmacy version,

charge – requested only in the pharmacy version,

plu – goods code (@ - if code is to be a QR– code, # - if code is to be a barcode),

description – goods description (not required),

action – takes the following values (not required):

1. action = „Sale” – (default value),
2. action = „Storno” – reversal.

Function of adding a line to receipt / invoice allows to call an additional option, which is giving a discount / markup to the chosen position. This option is defined in the following way:

```
<packet>
  <discount value="23%" name="Occasional" desc_id="1" action="discount"></discount>
</packet>
```

where:

value – value of given discount / markup,

name – name of given discount / markup (in case when the field „name” is completed, parameter „descid” is ignored, not required ),

action – takes the following values (not required):

1. action = „discount” – (default value),
2. action = „markup”.

### Warning !

Option of giving discount / markup to the chosen position must be placed in tag <item>, responsible for the position, to which discount / markup is to be given.

[Example:](#)

```
<packet>
  <item name="exemplary goodsr" quantity="1" quantityunit="szt" ptu="A" price="1.23"
    total="1.23" recipe="" charge="" plu="" description="" action="sale">
    <discount value="23%" name="Occasional" descid="1" action="discount"></discount>
  </item>
</packet>
```

As a result of sending this data packet to the receipt / invoice, there will be added goods with name „exemplary goods”, which price is 1.23. To the chosen position was given an occasional discount of 23 % of the goods value.

### 13. Settlement of returnable packages.

After sending the following packet, the printer receives order of returnable packages settlement.

```
<packet>
  <container action="sale" price="1.00" type="taken" quantity="1"></container>
</packet>
```

where:

action – takes the following values (not requested):

1. action = „sale” – (default value),
2. action = „storno” – reversal,

price – price for package,

type – takes the following values (not required):

1. type = „taken” – package sale (default value),
2. type = „returned” – package return,

quantity – quantity of packages (default 1, not required).

[Examples](#)

### 14. Giving discounts / markups.

After sending the following packet, the printer receives order of giving discount / markup to the sum of all positions on receipt.

```
<packet>
  <discount value="23%" name="Occasional" desc_id="1" type="subtotal" total="2" ptu="A"
    action="discount"></discount>
</packet>
```

where:

value – value of given discount / markup,

name – name of given discount / markup (in case when the field „name” is completed, parameter „desc\_id” is ignored, not required ),

type – takes the following values (not required):

1. type = „subtotal” – discount to subtotal of receipt (default value),
2. type = „in\_transaction” – discount during transaction. (This type displays information about amount of discount. (The discount is not printed on receipt),

total – sum of values of all receipt positions before giving a discount / markup.

ptu – VAT rates,

action – takes the following values (not required):

1. action = „discount” – (default value),
2. action = „markup”.

**Warning!**

1. If there is set type = „subtotal” then it is possible to add field „total”, thanks to which before giving a discount to the subtotal, there is checked its value and the sum of all positions on receipt and is reported an error in case of difference of these values.
2. If there is set type = „subtotal” there is possible to add field „ptu”. Then will be given a discount to the goods group in the chosen rate.
3. If there is set type = „in\_transaction”, the parameter „action” can take an additional value „none” – no discount / markup, cancelling earlier given discount / markup and all forms of payment.

## Examples

### 15. Non-fiscal printout – examples

#### 15.1. Non-fiscal printout with empty line and underline

#### 15.2. Non-fiscal printout with many lines, QR code and bar code

#### 15.3. Non-fiscal printout with bar code

#### 15.4. Non-fiscal printout with QR code

### 16. Information about date and version of device.

In response to the question:

```
<packet>
  <info action="date" version="" date=""></info>
</packet>
```

the printer sends back values in the following form:

```
<packet>
  <info action="date" version="" date="08-10-2013 11:13"/>
</packet>
```

where:

action – takes the following values:

1. action = „date” – sending back information about date and time,
2. action = „version” – sending back information about version of the device.

### 17. Information from ECR.

In response to the question:

```
<packet>
  <info action="checkout" type="receipt" last_error="?" isfiscal="?" receipt_open="?"
    last_receipt_error="?" reset_count="?" date="?" receipt_count="?" cash="?"
    unique_no="?" last_receipt="?" last_invoice="?" last_printout="?"></info>
</packet>
```

The printer sends back values in the following form:

```
<packet>
  <info action="cash" type="receipt" last_error="0" isfiscal="yes" receipt_open="no"
    last_receipt_error="no" reset_count="0" date="08-10-2013" receipt_count="11"
    cash="444.76" unique_no="ABC12345678" last_receipt="11" last_invoice="1"
    last_printout="47">
    <ptu name="A">12.50</ptu>
    <ptu name="B">0.00</ptu>
    <ptu name="C">0.00</ptu>
    <ptu name="D">0.00</ptu>
    <ptu name="G">0.00</ptu>
  </info>
</packet>
```

where:

type – takes the following values:

1. type = „receipt” – there are sent back receipts totalizers since the last daily report (default value),
2. type = „invoice” – there are sent back invoices totalizers since the last daily report,

last\_error – number of the last error,

isfiscal – is the printer fiscal,

receipt\_open – is the receipt open,

last\_receipt\_error – has an error appeared during the last transaction,

reset\_count – quantity of memory resettings,

date – current date,

receipt\_count – quantity of receipts,

cash – cash,

unique\_no – unique number,

last\_receipt – number of the last receipt,

last\_invoice – number of the last invoice,

last\_printout – number of the last printout,

name – rate.

## 18. Information about fiscal memory.

In response to the question:

```
<packet>
<info action="fiscal_memory" ></info>
</packet>
```

the printer sends back values in the following form:

```
<packet>
<info action="fiscal_memory" fiscal_memory_size="1048576" record_size="464" fiscal="1"
unique_no="ABC12345678" nip="123-456-78-90" max_records_count="2144"
records_count="7" max_reports_count="1830" reports_count="4" reset_max_count="200"
reset_count="0" tax_rates_prg_limit="30" tax_rates_prg="1" currency_change_prg_limit="4"
currency_change_prg="0" fiscal_start_date="dd-mm-yyyy hh:dd:ss" fiscal_stop_date="dd-
mm-yyyy hh:dd:ss" currency_name="PLN">
<ptu name="A">123.23</ptu>
<ptu name="B">123.23</ptu>
<sale>999.23</sale>
</packet>
```

where:

fiscal\_memory\_size – fiscal memory size (in bytes),

record\_size – record size (in bytes),

fiscal – action mode. Takes the following values:

1. fiscal = „0” – non-fiscal,
2. fiscal = „1” – fiscal,
3. fiscal = „2” – fiscal closed,

unique\_no – unique number,

nip – NIP number,

max\_record\_count – maximum quantity of records,

records\_count – records quantity,

max\_reports\_count – maximum quantity of daily reports,

reports\_count – quantity of daily reports,

reset\_max\_count – maximum quantity of RAM resettings,

reset\_count – quantity of RAM resettings,

tax\_rates\_prglimit – maximum quantity of PTU rate changes,

tax\_rates\_prg – quantity of PTU rate changes,

currency\_change\_prglimit – maximum quantity of currency changes,

currency\_change\_prg – quantity of currency changes,

fiscal\_start\_date – fiscalisation date,

fiscal\_stop\_date – date of closing a fiscal mode,

currency\_name – name of current currency.

## 19. Information of SD card.

In response to the question:

```
<packet>
  <info action="sd_card" type="external"></info>
</packet>
```

the printer sends back values in the following form:

```
<packet>
  <info action="sd_card" type="external" state="open" size="2145124352" free="2045"
  reports_count="3" last_report="4" last_access="08-10-2013 00:00"/>
</packet>
```

where:

type – takes the following values (not required):

1. type = „external” – (default value),

state – card state. Takes the following values:

1. state = „open”,
2. state = „close” – closed,
3. state = „unknow” – unknown,
4. state = „removed” – (no card),
5. state = „error” – error on card,
6. state = „process” – state while determining,

size – card size (in bytes),

free – how much free space,

reports\_count – quantity of files with reports,

last\_report – number of the last daily report,

last\_access – date of the last modification on card.

## 20. Request about state of current transaction.

In response to the question:

```
<packet>
  <info action="transaction"></info>
</packet>
```

the printer sends back values in the following form:

```
<packet>
  <info action="transaction" net_total="233" gross_total="343" type="receipt" mode="online">
    <total name="A" tax="23" net="100" gross="123"></total>
  </info>
</packet>
```

where:

net\_total – sum of totalizers net,

gross\_total – sum of totalizers gross,

type – transaction type. Takes the following values:

1. type = „none” – no transactions. No other data are sent back in this case,
2. type = „receipt” – open receipt. The additional parameter: mode = „online” or mode = „offline” is sent back in this mode,
3. type = „container” – packages settlement,
4. type = „invoice” – open invoice,

name – name of rate,

tax – tax value,

net – value of totalizer net

gross – value of totalizer gross.

## 21. Request about state of active or the last transaction.

In response to the question:

```
<packet>
  <info action="last_transaction"></info>
</packet>
```

the printer sends back values in the following form:

```
<packet>
  <info action="last_transaction" total="343" type="receipt" state="open" number="23"
    date="dd-mm-yyyy hh:dd" printout_no="123" checkbox_no="1" cashier="Jan"
    system_no="1234" discount_calc="1" euro_exchange_rate="4.344" rest="30"
    euro_rest="3">

    <item name="name" code="1234" quantity="4" ptu="A" price="2.10" total="8.40"
      total_discount="12%" total_after_discount="7.20">
      <discount value="5" name="Occasional" total="2" type="discount"
        desc_id="1"></discount>
    </item>
    <container code="2345" quantity="1" price="0.40" total="0.40"></container>
    <payment name="Gift" type="card" value="12"></payment>
    <discount value="23%" name="Occasional" total="2" type="discount" ></discount>
  </info>
</packet>
```

where:

type – type. Takes the following values:

1. type = „receipt” – receipt / invoice,
2. type = „container” – packages settlement,

state – status. Takes the following values:

1. state = „open”,
2. state = „close”,
3. state = „cancelled”,

number – receipt number,

date – date and time of receipt / invoice,

printout\_no – printout number,

checkbox\_no – checkout number,

cashier – cashier name,

system\_no – system number,

discount\_calc – method of discount calculating,

euro\_exchange\_rate – exchange rate of the foreign currency,

rest – rest,

euro\_rest – rest in the foreign currency,

name – discount name,

value – discount value (value as amount or percent),

total\_discount – the resultant total percent discount that results from given discounts on position.

total\_after\_discount – value of position after discounts.

total – value,

quantity – quantity of units of the concrete position.

## 22. Information about drawer status.

In response to the question:

```
<packet>
  <info action="drawer"></info>
</packet>
```

the printer sends back values in the following form:

```
<packet>
```



```
<info action="drawer" state="open"></info>
</packet>
```

where:

state – status. Takes the following values:

1. state = „open”,- drawer opened
2. state = „close”, drawer closed
3. state = „unknow” – unknown drawer status,

### 23. Information about device – general information

To collect information you should send a packet:

```
<packet>
<deviceinfo action="general"/>
</packet>
```

Response:

```
<packet>
<device_info      action="general"      device_name="HD      Online"      version="1.00"
module_printer_version="CITIZ" system_name="OES" system_version="1.00" display_count="2"
display_len="64"      ecopy="1"      fisca_lsize="1048576"/>
</packet>
```

where:

device\_name – device name  
version – program version  
module\_printer\_version- version of printing module  
system\_name – system name  
system\_version – system version  
display\_count – displays number  
display\_len – display width  
ecopy- is there a copy (1- there is, 0 – no display)  
fiscalsize – fiscal memory size

### 24. Information about device – detailed for fiscal memory

To collect information you should send packet:

```
<packet>
<device_info action="detail_fiscal_memory_info"/>
</packet>
```

Response:

```
<packet
<device_info      action="detail_fiscal_memory_info"      fiscal_size="1048576"      record_len="940"
mode="0" unique_no="" nip="" event_len="2108" record_count="6" max_daily_report_count="1830"
daily_report_count="0" resets_max_count="200" resets_count="0" max_tax_rates_count="30"
tax_rates_count="0"      currency_change_max_count="20"      currency_change_count="0"
fiscalization_date="2000-00-00      00:00:00"      fiscal_mode_close_date="2000-00-00      00:00:00"
currency_name="PLN"/>
</packet>
```

where:

fiscal\_size – fiscal memory size in bytes  
record\_len – record size in bytes  
mode – operating mode (0 – nonfiscal 1 – fiscal)  
unique\_no – unique number  
nip – nip (tax number)  
event\_len – number of general events

record\_count – records number  
 max\_daily\_report\_count – Maximum number of daily reports  
 daily\_report\_count – number of daily report  
 resets\_max\_count – Mximum number of RAM resettings  
 resets\_count – number of RAM resettings  
 max\_tax\_rates\_count – Maximum number of tax rates changes  
 tax\_rates\_count – number of tax rates  
 currency\_change\_max\_count – Maximum number of currency changes  
 currency\_change\_count – number of currency changes  
 fiscalization\_date – fiscalisation date  
 fiscal\_mode\_close\_date – date of fiscal mode closing  
 currency\_name – name of current currency

## 25. Information about device – details concerning power condition, accumulator and battery

To collect information from device you should send packet:

```

<packet>
<device_info action="battery_info"/>
</packet>
  
```

Response:

```

<packet>
<device_info      action="battery_info"      power_supply="yes"      power_supply_voltage="25"
power_accumulator_voltage="25"
power_battery_voltage="25"/>
</packet>
  
```

where:

power\_supply - connected power supply (yes, no)  
 power\_supply\_voltage – power supply voltage  
 power\_accumulator\_voltage – accumulator voltage  
 power\_battery\_voltage – battery voltage

## 26. Information about device – information about printout status

To collect information you should send packet:

```

<packet>
<device_info action="printout_info"/>
</packet>
  
```

Response:

```

<packet>
<device_info      action="printout_info"      printer_module_version="CITIZ"      printer_len="64"
printer_len_mm="80"      font="normal"      cutter="yes"      paper_count="1"/>
</packet>
  
```

where:

printer\_module\_version – version of printing module  
 printer\_len – printout width in characters (Maximum number of characters in line)  
 printer\_len\_mm – printout width in [mm]  
 font – increased or normal  
 cutter  
 paper\_count – amount of paper ( 0 – below critical level, 1 – above critical level, 2 – not serviced)

## 27. Information about device – information about drawer status

To collect information you should send packet:

```
<packet>  
<deviceinfo action="drawer_state"/>  
</packet>
```

Response:

```
<packet>  
<device_info action="drawer_state" drawer_state="drawer_close"/>  
</packet>
```

where:

drawer\_state – drawer status - drawer\_close – closed drawer, drawer\_open – open drawer

## 28. Information about device – information about data base

To collect information you should send packet:

```
<packet>  
<device_info action="db_article_state"/>  
</packet>
```

Response:

```
<packet>  
<device_info action="db_article_state" db_article_state="0"/>  
</packet>
```

## 29. Information about device – status of accumulator and power supply

To collect information you should send packet:

```
<packet>  
<device_info action="power_info"/>  
</packet>
```

Response:

```
<packet>  
<device_info action="power_info" power_mode="25" power_supply_voltage="25"  
power_accumulator_voltage="25" power_battery_voltage="25" accumulator_state="2"  
battery_state="5" charge_level_accumulator="15"/>  
</packet>
```

where:

power\_mode – power mode - 0 – work on accumulator 1 – work on power supply  
power\_supply\_voltage – power supply voltage [mV]  
power\_accumulator\_voltage – accumulator voltage [mV]  
power\_battery\_voltage – battery voltage to RAM/clock support [mV]  
accumulator\_state – accumulator status - 0 – totally discharged 1 – low level of charging 2 – charged  
and works properly 3 – accumulator charging 4 – accumulator voltage out of range 5- accumulator  
damaged  
battery\_state – Status of battery supporting RAM / clock - 0 – discharged / damaged 1 – should be  
charged 2 - battery status ok  
charge\_level\_accumulator – level of accumulator charging [%]

### 30. Information about communication port

To collect information you should send packet:

```
<packet>  
<info action="port"/>  
</packet>
```

Response:

```
<packet>  
<info action="port" value="PC1"/>  
</packet>
```

### 31. Information about dispatch to the repository.

In response to question:

```
<packet>  
<info action="repository"></info>  
</packet>
```

The printer sends back values in the following format:

```
<packet>  
<info action="repository" date_send="20-06-2018" data_send_correct="20-06-2018"  
jpk_send_count="14" jpk_actual_send_no="1"></info>  
</packet>
```

where:

date\_send – date of dispatch  
date\_send\_correct – date of the last correct dispatch  
jpk\_send\_count – number of jpk to send  
jpk\_actual\_send\_no – number of currently send jpk

### 32. Controlling instructions.

#### 32.1. Sound signal.

```
<packet>  
<control action="beep"></control>  
</packet>
```

#### 32.2. Paper feeding of quantity of lines.

```
<packet>  
<control action="paper_feed" value="6" cutter="yes"></control>  
</packet>
```

where:

value – number of lines, which paper is to move forward.  
cutter – takes value "yes" – if switched on, "no" – switched off

#### 32.3. Opening drawer.

```
<packet>
```

```
<control action="drawer"></control>
</packet>
```

#### 32.4. Showing text on display.

```
<packet>
<control action="display" line_1="test1" line_5="test5" line_8="test8"></control>
</packet>
```

where:

Line1... 8 – text shown on display.

#### 32.5. Display controlling in post-receipt mode.

```
<packet>
<control action="display_external" text="text"></control>
</packet>
```

Where action – takes values:

- display\_external – displaying text on customer display
- no\_display\_external – displaying switch off

#### 32.6. Display clearing.

```
<packet>
<control action="clear_display" ></control>
</packet>
```

Where action takes values:

- "clear\_display" – clears the printer's display.

#### 32.7. Access to the printer's menu.

```
<packet>
<control action="menu_open"></control>
</packet>
```

where action takes values:

- „open\_menu " – access to the printer's menu

### 33. Daily report.

After sending the following packet, the printer receives order of printing a daily report.

```
<packet>
<report type="daily" date="04-06-2004"></report>
</packet>
```

where:

date – date of daily report (not required, when you submit the current date you do not need to confirm with the keyboard ).

### 34. Periodical report.

After sending the following packet, the printer receives order of printing a periodical report.

```
<packet>
<report type="periodical" from="04-06-2004" to="06-06-2006" checkbox_no="1"
cashier="Jan" kind=""></report>
</packet>
```

where:

from, to – range of dates or report numbers,  
checkbox\_no – checkbox number (not required),  
cashier – cashier name (not required),  
kind – takes the following values:

1. kind = „full” – report is full (fiscal document) from range of chosen dates or numbers (default value),
2. kind = „sale\_summary” – sale summary (non-fiscal document) from range of chosen dates or numbers,
3. kind = „monthly\_full” – full monthly report (fiscal memory),
4. kind = „monthly\_summary” – monthly summary of sale (non-fiscal document),
5. kind = „billing\_full” – full billing report from range of chosen dates,
6. kind = „billing\_summary” – summary of billing report from range of chosen dates.

### 35. Shift report.

After sending the following packet, the printer receives order of printing a shift report.

```
<packet>
<report type="cash" checkbox="1" cashier="Jan" begin="" end="" income="10"
expense="10" cash="23.23" payin="40" payout="10" balance="30" receipt_count="12"
canceled_receipt_count="1" storno_count="0">
<payment name="gift card" value="234.12" type="card"></payment>
<container type="out" value="23"></container>
</report>
</packet>
```

where:

checkbox – checkbox number (not required),  
cashier – cashier name (not required),  
begin  
end ,  
income,  
expense,  
cash ,  
payin, - money paid in  
payout, - money paid out  
balance – cash balance in ECR,  
receipt\_count – quantity of receipts,  
canceled\_receipt\_count – quantity of cancelled receipts,  
storno\_count – quantity of reversals  
payment type – type of payment:

1. type = „cash”,
2. type = „card”,
3. type = „cheque”,
4. type = „voucher”,
5. type = „credit”,
6. type = „transfer”,
7. type = „account”,
8. type = „foreign”,

container -> type – takes the following values:

1. type = „out” – taken (sale),
2. type = „in” – returned (package return)

container -> value – amount of deposit taken / returned.

### 36. Report of data from protected memory.

```
<packet>  
<report type="protected_memory" from="04-06-2004" to="06-06-2006" kind="" report>  
</packet>
```

where:

from, to – range of numbers or dates (day-month-year),

kind – takes the following values:

1. kind = „receipt” – receipts,
2. kind = „invoice” – invoices,
3. kind = „daily\_report” – daily reports,
4. kind = „non\_fiscal” – non-fiscal printouts,
5. kind = „all” – all printouts,

### 37. Events report

To print events report you should send:

```
<packet>  
<report type="events" from="0" to="9999"/>  
</packet>
```

### 38. Register report in fiscal memory.

To make report you should send a packet:

```
<packet>  
<report type="events" from="1" to="99999" mode="full" daily_reports="yes">  
</report>  
</packet>
```

where:

events

from – initial number or initial date in format ("YYYY-MM-DD")

to – final number or final date in format ("YYYY-MM-DD")

mode:

- full– detailed report

settlement\_relief="yes" – discounts settlement

tax\_changes="yes" – tax rates changes

daily\_reports="yes" – daily reports

### 39. Report of service reviews.

To print report you should send a packet:

```
<report type="service_review">  
</report>  
</packet>
```

### 40. Cashier log-in and log-off.

Cashier's log-in is done by sending the following packet:

```
<packet>  
<cashier action="login" number="1" name="Jan"></cashier>  
</packet>
```

where:

action – takes the following values:

1. action = „login” – cashier log-in,
2. action = „logoff” – cashier log off,

number – checkbox number (not required),  
name – cashier name (not required).

#### 41. Data readout of currently log-in cashier.

```
<packet>  
<cashier action="state"></cashier>  
</packet>
```

where action takes values:

-„state” - data of currently log-in cashier.

#### 42. Pay-in, pay-out and cash balance in ECR.

Pay-in to ECR's cash drawer is done by sending the following packet:

```
<packet>  
<cash action="payin" value="2.30" checkbox_no="1" cashier="Jan"></cash>  
</packet>
```

where:

action – takes the following values:

1. action = „payin” – pay-in to the ECR cash drawer,
2. action = „payout” – pay-out from ECR cash drawer,
3. action = „read” – returns the current cash balance,

checkbox\_no – checkbox number (not required),  
cashier – cashier name (not required).

#### 43. Mode of errors handling.

After sending the following packet, in the printer would be set to errors handling mode.

```
<packet>  
<error action="set" value=""></error>  
</packet>
```

where:

value – takes the following values:

1. value = „display” – error is shown on display (not required),
2. value = „silent” – silent mode. No commands on the printer.

In order to collect the last error number, you should send to the printer the following packet:

```
<packet>  
<error action="get" value=""></error>  
</packet>
```

#### 44. Control of goods base.

Control of goods base is done by sending the following packet:

```
<packet>  
<dbcheck action="begin" mode="all" plu_name="pluname" ptu="A" checkbox_no="1"  
cashier="Jan"></dbcheck>
```



**</packet>**

where:

action – takes the following values:

1. action = „begin” – starting report,
2. action = „check” – checking / printout of goods,
3. action = „end” – ending report,

mode – takes the following values:

1. mode = „all” – printing all goods (default value),
2. mode = „lock” – printing only blocked goods,

plu\_name – name of goods,

ptu – VAT rate,

checkbox\_no – checkbox number (not required),

cashier – cashier name (not required).

#### 45. USB configuration.

In order to set parameters you should send a packet in the following form:

**<packet>**

```
<usb action="set" usb_modem_pid="12AB" usb_modem_vid="13AB"
usb_modem_type="acmx" usb_modem_protocol="0" usb_modem_interface="1"
usb_ex_pid="14AB" usb_ex_vid="15AB" usb_ex_type="usbx" usb_ex_protocol="1"
usb_ex_interface="0"/>
```

**</packet>**

where:

action – takes the following values:

1. action = „set” – setting configuration,
2. action = „get” – getting configuration

usb\_modem\_type –

acmx

usbx

#### 46. Configuration.

In order to set parameters you should send a packet in the following form:

**<packet>**

```
<config action="set">
<set id="0">200</set>
<set id="1">125</set>
<set id="discount_calculation">0</set>
```

**</config>**

**</packet>**

where:

action – takes the following values:

1. action = „set” – setting parameters,
2. action = „get” – getting / receiving parameters. In response you collect only these parameters, which you asked for.

id = „discount\_calculation” – choice of method of discount calculation.

In order to make a readout of parameters from given range:

**<packet>**

```
<config action="get" from="1" to="200">
```

```
</config>
```

**</packet>**

**Warning:**

All configuration options or possible configuration are placed in description of NOVITUS communication protocol.

#### 47. Setting date and hour.

In order to set the date and hour you should send to the printer packet in the following form:

```
<packet>  
<clock date="dd-mm-yyyy hh:dd"></clock>  
</packet>
```

where:

DD – day,  
MM – month,  
YYYY – year,  
HH – hour,  
DD – minute.

#### 48. Programming an online header

In order to program a header you should send to the printer a packet in the following form:

```
<packet>  
<header_online action="set">  
<line bold="no" inwers="no" align="center" double_height="yes" double_width="yes" index1="1"  
index2="0" index3="0" index4="0" index5="0" index6="0" index7="0" text1="COMP S.A. ul. Jutrzenki  
116" text2="" text3="" text4="" text5="" text6="" text7=""></line>  
<line bold="no" inwers="no" align="center" double_height="yes" double_width="yes" index1="1"  
index2="0" index3="0" index4="0" index5="0" index6="0" index7="0" text1="COMP S.A. ul. Jutrzenki  
116" text2="" text3="" text4="" text5="" text6="" text7=""></line>  
<line bold="no" inwers="no" align="center" double_height="yes" double_width="yes" index1="1"  
index2="0" index3="0" index4="0" index5="0" index6="0" index7="0" text1="COMP S.A. ul. Jutrzenki  
116" text2="" text3="" text4="" text5="" text6="" text7=""></line>  
<line bold="no" inwers="no" align="center" double_height="yes" double_width="yes" index1="1"  
index2="0" index3="0" index4="0" index5="0" index6="0" index7="0" text1="COMP S.A. ul. Jutrzenki  
116" text2="" text3="" text4="" text5="" text6="" text7=""></line>  
<line bold="no" inwers="no" align="center" double_height="yes" double_width="yes" index1="1"  
index2="0" index3="0" index4="0" index5="0" index6="0" index7="0" text1="COMP S.A. ul. Jutrzenki  
116" text2="" text3="" text4="" text5="" text6="" text7=""></line>  
<line bold="no" inwers="no" align="center" double_height="yes" double_width="yes" index1="1"  
index2="0" index3="0" index4="0" index5="0" index6="0" index7="0" text1="COMP S.A. ul. Jutrzenki  
116" text2="" text3="" text4="" text5="" text6="" text7=""></line>  
<line bold="no" inwers="no" align="center" double_height="yes" double_width="yes" index1="1"  
index2="0" index3="0" index4="0" index5="0" index6="0" index7="0" text1="COMP S.A. ul. Jutrzenki  
116" text2="" text3="" text4="" text5="" text6="" text7=""></line>  
</header_online></packet>
```

where:

action – takes the following values:

1. action = „set” – set a header,
2. action = „get” – header readout,

bold – (not requested),

align – takes the following values (not required);

1. align = „centered”,
2. align = „left” – left-aligned (default value),
3. align = „right” – right-aligned,

inwers – inwers (not required);

double\_height – double height (yes/no)

double\_width – double width (yes/no)

index1...index7 – indexes of fields in line corresponding to fields types

- 0- none
- 1- tax payer name
- 2- postal code
- 3- city
- 4- building number
- 5 – place number
- 6 - post office name
- 7 – street name
- 8 – other dates

text1...text7 – content of line, text number corresponds to index (line consists of 7 fields)

#### 49. Online approval (fiscalisation)

To get approve for the printer (get fiscalisation) you should send a packet:

```
<packet>
<fiscalisation action="training_mode" type_device="receipt_printer" nip="1111111111"
office_name="urzad" service_name="service" service_man_name="serwisant"
service_id="serw111" tax_office_code ="1234" reg_number="111" tax_service="2222"
type_numbering="global" />
</packet>
```

where:

action – takes the following values:  
training\_mode – training mode  
fiscalization – approval  
type\_device – takes values:  
receipt\_printer – receipt printer  
ticket\_printer – ticket printer  
pharmace\_printer – pharmace printer  
type\_numbering – takes values:  
global – global receipt numbering  
daily – daily receipt numbering  
office\_name- tax office data  
service\_name – service point data  
service\_man\_name – service man data  
service\_id – service man ID  
reg\_number – registration number  
tax\_office\_code – tax office code  
tax\_service – tax number of service point

#### 50. Programming producer's data

```
<packet>
<producer unique_no="ABC0123456789" producer_no="ABC0123456789"
mac="123" producer_option="0">
</producer>
</packet>
```

#### 51. Service lock.

In order to impose a service lock you should send to the printer a packet in the following form:

```
<packet>
<service action="lock" date="dd-mm-yyyy" description="service lock"
password="1234"></service>
</packet>
```

where:

action – takes the following values:

1. action = „lock” – service lock,

date – date (day – month – year). If the date isn't set, the review is deleted.

description – description (not required),

password – password (not required),

in order to unlock send the following packet:

```
<packet>
  <service action="unlock"></service>
</packet>
```

## 52. Readout of update logs.

To read firmware update logs you should send packet:

```
<packet>
<update_log action="begin"/>
</packet>
```

where:

action – takes values:

begin – start downloading logs

next – downloading the next set of logs

Response:

```
<packet>
<update_log action="data" value=""/>
</packet>
```

where:

action – takes values:

data – next data to download

end – end of logs downloading

## 53. Readout of module logs

To read module logs you should send packet:

```
<packet>
<processlog action="begin" date_time_start="" date_time_stop="" level="" module=""/>
</packet>
```

where:

action – takes values:

begin – starting logs readout

next – downloading the next line of logs

date\_time\_start – date from which starts logs readout, format yyyy-mm-dd hh:mm:ss

date\_time\_stop – final date of logs readout, format yyyy-mm-dd hh:mm:ss

level – logs level, takes values:

DEBUG

LOG

ERROR

CRITICAL

module – module from which you want to read logs

ALL

General

TPM

http

BIOS

SPI

FW

Response:

```
<packet>
  <processlog action="data" date_time_start="" date_time_stop="" level="" module="" value="2018-01-23T08:53:12.470Z LOG TPM OFDCryptoModule.cpp:341, init: 0 1 2&#xA;" />
</packet>
```

where:

action - takes values  
date – there are the next data to read  
end – end of logs read  
value – log data

Warning:

Logs are read line by line, when we specify any of the parameters, only logs matching the selection criteria are displayed, when no criteria is given, everything is sent back until the entire log file is read.

#### 54. Service review.

To set the date of service review you should send to the printer a packet in the following form:

```
<packet>
  <service action="review" date="dd-mm-yyyy" description="periodical review"></service>
</packet>
```

where:

date – date of review (day – month – year),  
description – description (not required),

To get confirmation of making service review you should send to the printer packet in the following format:

```
<packet>
  <service action="review_done" service_id="123456"></service>
</packet>
```

where:

service\_id – service man id number

To get a report of conducted service reviews you should send to the printer packet in the following form:

```
<packet>
  <report type="service_review">
</report>
</packet>
```

#### 55. Confirmation of making a service review.

To get a confirmation of service review you should send packet:

```
<packet>
  <service action="review_done" service_id="123456"></service>
</packet>
```

where:

service\_id – service man id number

#### 56. Programming change of currency.

In order to program changing currency, you should send to the printer packet in the following form:

```

<packet>
  <currency action="change" date="dd-mm-yyyy hh:dd" name="EUR"
    exchange_rate="4.0000001"></currency>
</packet>

```

where:

date – date and time of change (day – month – year hour, minute),  
name – the-three-letter name of currency complied with the convention used by NBP (the National Bank of Poland),  
exchange\_rate – exchange rate (from: 0:00000001 to: 9999:99999999),

### 57. Programming mode of printing currency conversion.

In order to program mode of printing currency conversion, you should send to the printer packet in the following form:

```

<packet>
  <currency action="print" name="EUR" exchange_rate="4.0000001"
    type="defined"></currency>
</packet>

```

where:

name – the-three-letter name of currency that complied with the convention used by NBP (the National Bank of Poland), ,  
exchange\_rate – exchange rate (from: 0:00000001 to: 9999:99999999),  
type – takes the following values:

1. type = „defined” – defined mode. Fields „Name” and „exchange\_rate” are required,
2. type = „none” – without conversion

### 58. Setting PTU rates.

In order to set PTU rates you should send to the printer packet in the following form:

```

<packet>
  <tax_rates action="set" checkbox="1" cashier="Jan" date="dd-mm-yyyy hh:dd">
    <ptu name="A">23%</ptu>
  </tax_rates>
</packet>

```

where:

action – takes the following values:

1. action = „set” – setting rates,
2. action = „get” - read rates,

checkbox – checkbox number (not required),

cashier – cashier name (not required),

date – date and hour of modification (day – month – year hour:minutes),

name – name of PTU rate.

#### Warning!

To enter rate free of tax, you should in tag „ptu” instead of percentage value enter value „free”.

Example:

```

<packet>
  <tax_rates action="set" checkbox="1" cashier="Jan" date="10-10-2013 10:00">
    <ptu name="A">23%</ptu>
    <ptu name="B">free</ptu>
  </tax_rates>
</packet>

```

After completing this packet there would be set two VAT rates. Rate A includes 23% of tax while rate B is free of tax.

### 59. Request of sending back data from fiscal memory.

In order to read data from fiscal memory, you should send to the printer the following packet:

```
<packet>
<fiscal_memory action="get" from="dd-mm-yyyy hh:dd"></fiscal_memory>
</packet>
```

then you should send the same data packet with parameter action="next" in order to read the next data packages.

where:

action – takes the following values:

1. action = „get” – request of sending back data
2. action = „next” – sending back the next record,

from – date and hour or record number.

Sending back records can take the following types:

#### 59.1. Type = „report” – Record of daily report.

```
<packet>
<fiscal_memory type="report" time="dd-mm-yyyy hh:dd" receipt_count="23"
invoice_count="3" canceled_receipt_count="2" number="1" last_receipt="4" last_invoice="2"
last_printout="20" sale="12342.33" tax="20" total_sale="123213.12" total_tax="232.12"
invoice_sale="21.33" invoice_tax="22.12" invoice_total_sale="1213.52"
invoice_total_tax="3231.12" currency_name="PLN" non_taxable="0.00" tickets="0"
normal="0" subsidy="0" reduced="0" abroad="0">

<ptu name="A" rate="23%" sale="2.31" tax="23.12" total_tax="21142.23" invoice_sale="2.20"
invoice_tax="0.23" invoice_total_tax="232.12"></ptu>

</fiscalmemory>
</packet>
```

where:

time – time of record registering,  
receipt\_count – quantity of receipts,  
invoice\_count – quantity of invoices,  
canceled\_receipt\_count – quantity of cancelled receipts  
number – report number,  
last\_receipt – number of last receipt,  
last\_invoice – number of last invoice,  
sale – sum of gross sale (receipts),  
tax – sum of PTU tax (receipts),  
total\_sale – sum of sale increasingly (receipts),  
total\_tax – sum of tax increasingly (receipts),  
invoice\_sale – sum of sale (invoices),  
invoice\_tax – sum of tax (invoices),  
invoice\_total\_sale – sum of sale increasingly (invoices),  
invoice\_total\_tax – sum of tax increasingly (invoices),  
currency\_name – name of currency,  
name = "A".."G" – name of rate,  
rate – tax rate,  
sale – sale in the rate (receipts),  
tax – value of tax in the rate (receipts),  
total\_tax – value of tax increasingly in the rate (receipts),  
invoice\_sale – sale in the rate (invoices),  
invoice\_tax – value of tax in the rate (invoices),

invoice\_total\_tax – value of tax increasingly in the rate (invoices)  
 non\_taxable - value not subject to PTU (accessible since HD E 3.0)  
 tickets – number of tickets (accessible since HD E 3.0)  
 normal – number of normal tickets (accessible since HD E 3.0)  
 subsidy – number of half-price tickets with supplements (accessible since HD E 3.0)  
 reduced – number of half-price tickets without supplements (accessible since HD E 3.0)  
 abroad – number of abroad tickets (accessible since HD E 3.0)

#### 59.2. Type = „reset” – Resetting RAM memory.

```
<packet>
  <fiscal_memory type="reset" time="dd-mm-yyyy hh:dd" number="4"
  reason="internal"></fiscal_memory>
</packet>
```

where:

time – time of record registering,  
 reason – reason of memory resetting (internal or external),

#### 59.3. Type = „tax\_rates” – Programming VAT rates.

```
<packet>
  <fiscal_memory type="tax_rates" time="dd-mm-yyyy hh:dd">
  <ptu name="A">23%</ptu>
  <ptu name="A">Free</ptu>
  </fiscal_memory>
</packet>
```

where:

time – time of record registering.

#### 59.4. Type = „currency\_set” – Record of setting a currency change.

```
<packet>
  <fiscal_memory type="currency_set" time="dd-mm-yyyy hh:dd" name="EUR"
  change_time="dd-mm-yyyy hh:dd" exchange_rate="4.3231"></fiscal_memory>
</packet>
```

where:

time – time of record registering,  
 name – currency name,  
 change\_time – date and time of currency change,  
 exchange\_rate – exchange rate.

#### 59.5. Type = „currency\_change” – Record of currency change.

```
<packet>
  <fiscal_memory type="currency_change" time="dd-mm-yyyy hh:dd" name="EUR"
  change_time="dd-mm-yyyy hh:dd" exchange_rate="4.3231"></fiscal_memory>
</packet>
```

where:

time – time of record registering,  
 name – currency name,  
 change\_time – date and time of currency change,



exchange\_rate – exchange rate.

#### 59.6. Type = „cardclose” – Record of closing card.

```
<packet>
  <fiscal_memory type="card_close" time="dd-mm-yyyy hh:dd" number="23"
  close_time="dd-mm-yyyy hh:dd"></fiscal_memory>
</packet>
```

where:

time – time of record registering,  
number – card number (1..999).

#### 59.7. Type = „end” – End of data.

```
<packet>
  <fiscal_memory type="end"></fiscal_memory>
</packet>
```

#### 60. Request of sending back logs from port.

In response to the question:

```
<packet>
  <log_pl action="read" port="com1"></log_pl>
</packet>
```

the printer sends back information in the following form:

```
<packet>
  <log_pl action="read" port="com1" date="dd-mm-yyyy hh:dd">dane....</log_pl>
  <log_pl action="write" port="com1" date="dd-mm-yyyy hh:dd">dane....</log_pl>
</packet>
```

where:

Port – chosen port „com1”, „com2”, ..., „comN” . If there is no port parameter, there are sent back data from the all ports,

Action – takes the following values:

1. Action = „read” – read data,
2. Action = „write” – sent data,

Date – date and time of retrieving / sending data (day – month – year hour:minutes).

#### 61. Non-fiscal printout.

In order to perform non-fiscal printout you should send to the printer the following data packet:

```
<packet>
  <non-fiscal_printout system_no="123" non-fiscal_header="yes">
    <line type="line" bold="yes" inwers="yes" center="yes" font_id="1" print_on="copy"
    font_attr="big">linia1</line>
    <line type="underline"></line>
    <line type="graphic">graphic number</line>
    <line>linia 3</line>
  </non-fiscal_printout>
</packet>
```

where:

system\_no – system number (not required),

non-fiscal\_header – describes whether the header "Non-fiscal printout" is to be printed. It takes values "yes" - printed, "no" – not printed.

type – takes the following values (parameter not required):

1. type = „line” – text (default value),
2. type = „barcode” – bar code,
3. type = „qrcode” – code 2D,
4. type = „underline” – underline,
5. type = "graphic" – programmed graphic

bold – bolded (reduces by half number of characters falling in the line, not required),

font\_id – font (1 – basic font, 2- second font, parameter not required),

font\_attr – font attributes. It takes the following values (not required):

1. font\_attr = „big” – bigger font (reduces by half number of characters in line. Enlargement contained in font type),
2. font\_attr = „high” – higher font,
3. font\_attr = „bold” – widen font (reduces by half number of characters in line),
4. font\_attr = „inwers” – inversion,

### Warning:

QR code (type = „qrcode”) takes defined characters and limits, which are dependent on device on which they are used. Detailed information concerning operating and QR code are placed in description of Novitus communication protocol.

## 62. Programming a graphic.

In order to program a graphic header you should send to the printer the following data packet:

```
<packet>  
<graphic action="programm" id="0" type="all" >graphic in hexadecimal form</graphic>  
</packet>
```

where:

id – number of programmed graphic (0 – 50, where 0 – graphic header),

type – takes the following values:

1. type = „begin” – start of graphic data packet,
2. type = „next” – the next graphic data packet,
3. type = „end” – end of graphic,
4. type = „all” – the whole graphic in one data packet.

In order to delete graphic or collect its checksum you should send the following data packet:

```
<packet>  
<graphic action="delete" id="0"></graphic>  
</packet>
```

where:

action – takes the following values:

1. action = „delete” – deleting a graphic header,
2. action = „read\_crc” – readout of graphic checksum,

To read a programmed graphic you should send the following data packet:

```
<packet>  
<graphic action="read" id="0" type="begin" ></graphic>  
</packet>
```

where:

action – takes the following values:

1. type = „begin” – start of graphic readout,
2. type = „next” – the next graphic data packet,

3. type = „repeat” – repeat sending back the previous data packet,

In response to the above packet, the printer sends back data in the following form:

```
<packet>
<graphic action="read" id="0" type="data" >graphic in hexadecimal form</graphic>
</packet>
```

where:

type – takes the following values:

1. type = „data” – the next data packet
2. type = „end” – end of graphic.

#### Warning:

To program a graphic you should send data in hexadecimal form. The graphic can be sent in one packet (provided that packet size doesn't exceed memory buffer size) or it can be divided into smaller packets. In that case data in hexadecimal form must have even number of characters.

Examples:

- [Graphic programming](#)
- [Graphic deleting](#)

### 63. Programming servers' address for the printers with online system

To program servers' address you should send a packet:

```
<packet>
<serweraddr action="set" value="" type="1">
</serweraddr>
</packet>
```

where:

action – takes the following values:

- set – program servers' address
- get – downloading servers' address

value – server's address, when downloading address the field is empty

type – server's type 1- server CPD, 2 – server WebApi, when downloading address, it must be submitted for which type

when action takes value – get, response looks like:

```
<packet >
<serweraddr action="get" value="https://esb-te.mf.gov.pl:5062" type="1"/>
</packet>
```

### 64. Programming an animation.

To program an animation you should send to the printer packets in the following form:

```
<packet>
<animation action="programm" name="test.png" size="147682" type="begin"> data
packets in coding base64</animation>
</packet>
```

where:

action = „programm” – programming an animation,

name – animation name with extension

type – takes the following values:

1. type = „begin” – the first data packet,
2. type = „next” – next data packets,

There are also possible other operations concerning an animation. In order to use one of them you should send to the printer the following data packet:

```

<packet>
  <animation action="delete" name="test.png" > </animation>
</packet>

```

where:

action – takes the following values:

1. action = „delete” – deleting chosen animation,
2. action = „readcrc” – readout of chosen animation checksum,
3. action = „setactive” – setting chosen animation as active,
4. action = „hide” – hiding an active animation (attribute 'name' is not required),
5. action = „deleteall” – deleting all animations placed in the printer (attribute 'name' is not required).
6. action = „readlist” – read list of programmed animations (option accessible since version HD E 2.0) ,
7. action = „readactive” – read name of active animation (option accessible since version HD E 2.0) .

#### Warning:

To program an animation you should send data packet prepared in coding base64. You ought to remember that length of a single packet doesn't exceed memory buffer size, otherwise the printer reports an error. The name must be unique and it should be submitted with file extension, and the animation size should be submitted in bytes and it's the physical size of image on your hard drive. The first packet must be sent with attribute type = „begin”, and the next packets with parameter type = „next”. Sending will be completed when size of sent packets would be equal to the size given in animation attributes.

To program a timesheet of animations displaying you should send to the printer the following data packet:

```

<packet>
  <animation action="programmplaylist" id="0" datetimestart="11-11-2014 13:13"
  datetimestop="31-11-2014 23:00:00" name="name.png" day="127" repeat="5" active="1"
  default="0" clocksettingvertical="2" clocksettinghorizontal="0" clockvisible="1"
  disp_date="22"> </animation>
</packet>

```

where:

action – takes the following values:

1. action = „programmplaylist” – programming a timesheet of advertisements displaying.
2. action = „receiveplaylist” – read position of timesheet with given identifier
3. action = „deleteplaylist” – playlist deleting (accessible since HD E 3.0)

id – identifier of timesheet position

date\_time\_start – date and time of beginning animation displaying in form day-month- year hour:minute. The hour indicates an accurate time of starting animation on each day comprised between start date, end date and days of week.

date\_time\_stop – date and time of ending animation displaying in form hour:minute:second day-month- year hour:minute. The hour indicates an accurate time of ending animation displaying on each day comprised between start date, end date and days of week.

name – name of animation (with its extension), for which a timesheet is programmed.

day – the number corresponding to the days of week, on which an advertisement is to be displayed. Each day corresponds to one bit. 0 – means that none of day was selected, 127 –all days of week.

repeat – number of repetitions of animation displaying before switching to the next in line.

active – takes the following values:

1. active = „1” – timesheet with given identifier active
2. active = „0” – timesheet with given identifier non active.

default – takes the following values:

1. default = „1” – default graphic.
2. default = „0” – non default graphic.

clock\_setting\_vertical - placing the clock vertically (2 - up, 3- down, 4 - center),

clock\_setting\_horizontal – placing the clock horizontally (0 - left, 1- right, 4- center)

clock\_visible – takes the following values:

1. clock\_visible = „1” – clock visible
2. clock\_visible = „0” – clock invisible.

disp\_date = minimum time of advertisement displaying





```

<line type="line" font_number="1" >line 1</line>
<line type="line" font_number="1" >line 2</line>
<line type="line" font_number="1" >line 3</line>
<line type="line" font_number="1" >line 4</line>
<line type="line" font_number="1" >line 5</line>
<line type="qr_code" bolded="yes" centered="yes" font_number="1"
font_attribute="big">NOVITUS</line>
</non-fiscal_printout>
</packet>

```

### 69.2. Non-fiscal printout with several lines, QR code and bar code

```

<packet>
<non-fiscal_printout system_number="12345678">
<line type="line" font_number="1" >line 1</line>
<line type="line" font_number="1" >line 2</line>
<line type="line" font_number="1" >line 3</line>
<line type="line" font_number="1" >line 4</line>
<line type="line" font_number="1" >line 5</line>
<line type="qr_code" bolded="yes" centered="yes" font_number="1"
font_attribute="big">NOVITUS</line>
<line type="qr_code" bolded="yes" centered="yes" font_number="1"
font_attribute="big">NOVITUS</line>
</non-fiscal_printout>
</packet>

```

### 69.3. Non-fiscal printout with empty line and underline

```

<packet>
<non-fiscal_printout system_number="123" non-fiscal_printout_header="yes">
<line type="line" bolded="yes" negative="no" centered="yes" font_number="1"
printout_on="copy" font_attribute="big">line1</line>
<line type="bolded"></line>
<line>line 3</line>
<line></line>
<line>line 4</line>
</non-fiscal_printout>
</packet>

```

## 70. Examples – receipts

### 70.1. Payments

#### 70.1.1. Receipt with many items

```

<packet>
<receipt action="begin" mode="online">
</receipt>
<position name="water" quantity="1" unit="pcs" rate="A" price="100.00" recipe="" charge="" plu="#"
description="" action="sale"/>
<position name="juice" quantity="1" unit="pcs" rate="A" price="150.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="sugar" ilosc="1" unit="pcs" rate="A" price="50.00" recipe="" charge="" plu="@ "
description="" action="storno"/>
< position name="bread" quantity="1" unit="pcs" rate="A" price="50.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10%" discount_id="2" action="discount"></discount></position>
<position name="butter" quantity="1" unit="pcs" rate="A" price="50.00" recipe="" charge="" plu="#"
description="" action="sale"/>

```

```

<position name="eggs" quantity="1" unit="pcs" rate="A" price="50.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="beer" quantity="1" unit="pcs" rate="A" price="50.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10.00" name="special" action="markup"/>
<position name="candy" quantity="1" unit="pcs" rate="A" price="50.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="fingersticks" quantity="1" unit="pcs" rate="A" price="50.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="bananas" quantity="1" unit="pcs" rate="A" price="50.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="nuts" quantity="1" unit="pcs" rate="A" price="50.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" system_number="123" cashbox_number="1" cashier="Jan Kowalski"
amount="655.00">
<discount value="39%" discount_id="6" action="discount"/></receipt>
</packet>

```

#### 70.1.2. Goods sale

```

<packet>
<receipt action="begin" mode="online"></receipt>
<position name="bread" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" system_number="" cashbox_number="02" cashier="Adam Adam"
amount="1.00" />
</packet>

```

#### 70.1.3. Payment with change

```

<packet>

<receipt action="begin" mode="online"></receipt>
<position name="bread" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<payment type="cash" action="add" value="10.00"></payment>
<receipt action="close" system_number="" cashbox_number="02" cashier="Adam Adam"
amount="1.00" />
</packet>

```

#### 70.1.4. Payment by card

```

<packet>
<receipt action="begin" mode="online"></receipt>
<position name="bread" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<payment type="card" action="add" value="1.00" name="VISA"></payment>
<receipt action="close" system_number="" cashbox_number="02" cashier="Adam Adam"
amount="1.00" />
</packet>

```

#### 70.1.5. Payment by card with change

```

<packet>
<receipt action="begin" mode="online"></receipt>

```



```

<position name="bread" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<payment type="card" action="add" value="10.00" name="VISA"></payment>
<receipt action="close" system_number="" cashbox_number="02" cashier="Adam Adam"
amount="1.00" />
</packet>

```

#### 70.1.6. Supporting forms of payment – many forms of payment

```

<packet>
<receipt action="begin" mode="online"></receipt>
<position name="coca cola" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pepsi" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="fingersticks" quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="kiwi" quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="kiwi juice" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="water" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="mineral water" quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="sugar" quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pepper" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="chili pepper" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="horseradish" quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="tangerines" quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<payment type="card" action="add" value="20.00" name="card"></payment>
<payment type="cheque" action="add" value="30.00" name="cheque"></payment>
<payment type="credit" action="add" value="50.00" name="credit"></payment>
<payment type="mobil" action="add" value="100.00" name="mobil"></payment>
<receipt action="close" system_number="" 123456" cashbox_number="02" cashier="Adam Adam"
amount="300.00" />
</packet>

```

#### 70.1.7. Supporting forms of payment – cancelling 3 from 4 payments

```

<packet>
<receipt action="begin" mode="online"></receipt>
<position name="coca cola" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pepsi" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="fingersticks" quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="kiwi" quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="kiwi juice" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>

```

```

<position name="water" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="mineral water" quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="sugar" quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pepper" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="chili pepper" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="horseradish " quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="tangerines " quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<payment type="card" action="add" value="20.00" name="card"></payment>
<payment type="cheque" action="add" value="30.00" name="cheque"></payment>
<payment type="credit" action="add" value="50.00" name="credit"></payment>
<payment type="mobil" action="add" value="100.00" name="mobil"></payment>
< payment type="card" action="delete" value="20.00" name="card"></payment>
< payment type="cheque" action="delete" value="30.00" name="cheque"></payment>
< payment type="credit" action="delete" value="50.00" name="credit"></payment>
< receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="300.00" />
</packet>

```

## 70.2. Discounts / Markups

### 70.2.1. Sale with quota discount on the position

```

<packet>
<receipt action="begin" mode="online"></receipt>
<position name="water" quantity="1" unit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10.00" name="special" discount_id="1" action="discount"></discount>
</position>
<position name="soda water" quantity="1" unit="" rate="A" price="5.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="6.00"></receipt>
</packet>

```

### 70.2.2. Sale with percentage markup on the position

```

<packet>
<receipt action="begin" mode="online"></receipt>
<position name="water" quantity="1" unit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10%" name="special" discount_id="1" action="markup"></discount>
</position>
<position name="soda water" quantity="1" unit="" rate="A" price="5.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="17.10"></receipt>
</packet>

```

### 70.2.3. Sale with percentage discount on the position

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="water" quantity="1" unit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/></position>
<position name="soda water" quantity="1" unit="" rate="A" price="5.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="16.00"></receipt>
<discount value="20%" name="special" discount_id="1" action="discount">
</discount>
</receipt>
</packet>
```

### 70.2.4. Sale with quota markup on the receipt

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="water" quantity="1" unit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/></position>
<position name="soda water" quantity="1" unit="" rate="A" price="5.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="16.00">
<discount value="5.00" name="special" discount_id="1" amount="16.00" action="markup">
</discount>
</receipt>
</packet>
```

### 70.2.5. Discounts on the position and on sub-total and on the receipt

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="water" quantity="1" unit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10.00" name="special" discount_id="1" action="discount">
</discount>
</position>
<discount value="10%" name="" discount_id="1" action="discount"></discount>

<position name="water" quantity="1" unit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10%" name="special" discount_id="1" action="discount">
</discount>
</position>
<discount value="1.80" name="special" discount_id="1" action="discount"></discount>
<receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="9.00">
<discount value="10%" name="special" discount_id="1" action="discount"></discount>
</receipt>
</packet>
```

### 70.2.6. Percentage discount on goods in given PTU rate - A

```
<packet>
<receipt action="begin" mode="online"></receipt>
```

```

<position name="coca cola" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pepsi" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="fingersticks" quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="kiwi" quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="kiwi juice" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="water" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="mineral water" quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="sugar" quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pepper" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="chili pepper" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="horseradish" quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="tangerines" quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="20%" name="special" discount_id="1" rate="A" action="discount"></discount>
<payment type="cash" action="add" value="300.00"></payment>
< receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="294.00"></receipt>
</packet>

```

#### 70.2.7. Quota markup on goods in given PTU rate – C

```

<packet>
<receipt action="begin" mode="online"></receipt>
<position name="coca cola" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pepsi" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="fingersticks" quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="kiwi" quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="kiwi juice" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="water" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="mineral water" quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="sugar" quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pepper" quantity="1" unit="" rate="A" price="10.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="chili pepper" quantity="1" unit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="horseradish" quantity="1" unit="" rate="C" price="30.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="tangerines" quantity="1" unit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10.00" name="special" discount_id="1" rate="C" action="markup"></discount>
<payment type="cash" action="add" value="340.00"></payment>
< receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="310.00"></receipt>

```

</packet>

### 70.3. Deposits

#### 70.3.1. Deposit return on the receipt

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="soda water" quantity="1" unit="" rate="A" price="5.00" recipe="" charge="" plu=""
description="" action="sale"/>
<deposit action="sale" price="0.40" type="returned" quantity="2"></deposit>
< receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="5.00"></receipt>
</packet>
```

#### 70.3.2. Reverse (storno) of deposit collected on the receipt

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="soda water" quantity="1" unit="" rate="A" price="5.00" recipe="" charge="" plu=""
description="" action="sale"/>
<deposit action="storno" price="0.35" type="returned" quantity="1"></deposit>
< receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="5.00"></receipt>
</packet>
```

#### 70.3.3. Collections and refunds of deposit on the receipt

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="soda water" quantity="1" unit="" rate="A" price="5.00" recipe="" charge="" plu=""
description="" action="sale"/>
<deposit action="sale" price="0.35" type="taken" quantity="2"></deposit>
<deposit action="sale" price="0.10" type="returned" quantity="8"></deposit>
<deposit action="sale" price="0.40" type="returned" quantity="2"></deposit>
<payment type="cash" action="add" value="5.00"></payment>
< receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="5.00"></receipt>
</packet>
```

### 70.4. Reversals (Storno)

#### 70.4.1. Cancelled receipt.

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="bread" quantity="1" unit="" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="cancel"></receipt>
</packet>
```

#### 70.4.2. Receipt with storno

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="juice" quantity="1" unit="" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="beer" quantity="1" unit="" rate="A" price="2.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="juice" quantity="1" unit="" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="storno"/>
<receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="2.00">
</packet>
```

#### 70.4.3. Receipt with storno after discount to sub-total

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="juice2" quantity="1" unit="" rate="A" price="90.00" recipe="" charge="" plu=""
description="" action="sale"/></position>
<position name="mango" quantity="1" unit="" rate="A" price="50.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10%" name="special" discount_id="1" action="discount"></discount>
<position name="juice2" quantity="1" unit="" rate="A" price="90.00" recipe="" charge="" plu=""
description="" action="storno"></position>
<receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="36.00"></receipt>
</packet>
```

#### 70.5. Others

##### 70.5.1. Receipt with footer's lines - 3 lines sys no. QR code

```
packet>
<receipt action="begin" mode="online"></receipt>
<position name="bread1" quantity="1" unit="" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="1.00">
<non-fiscal_line type="definition">
<line>line 1</line>
<line>line 2</line>
<line>@line3</line>
</non-fiscal_line>
</receipt>
</packet>
```

##### 70.5.2. Receipt with additional non-fiscal part

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="bread2" quantity="1" unit="" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="1.00">
<non-fiscal_line type="definition">
<line>line 1</line>
```

```

<line>line 2</line>
<line>line 3</line>
</non-fiscal_line>
<non-fiscal_line type="line" id="1" value="567"></non-fiscal_line>
< non-fiscal_line type="line" id="2" value ="602"></non-fiscal_line>
<non-fiscal_line type="line" id="3" value="777"></non-fiscal_line>
<non-fiscal_line type="line" id="4" value="Kowalski"></non-fiscal_line>
<non-fiscal_line type="line" id="5" value="ABCD"></non-fiscal_line>
<non-fiscal_line type="line" id="6" value="987"></non-fiscal_line>
<non-fiscal_line type="line" id="7" value="28.09.2020"></non-fiscal_line>
<non-fiscal_line type="line" id="8" value="75757"></non-fiscal_line>
<non-fiscal_line type="line" id="9" value="NCASHIER"></non-fiscal_line>
<non-fiscal_line type="line" id="10" value="100.50"></non-fiscal_line>
<non-fiscal_line type="line" id="11" value="PLN"></non-fiscal_line>
<non-fiscal_line type="line" id="12" value="4.000000"></non-fiscal_line>
<non-fiscal_line type="line" id="13" value="54545"></non-fiscal_line>
<non-fiscal_line type="line" id="14" value="56"></non-fiscal_line>
<non-fiscal_line type="line" id="15" value="worker name"></non-fiscal_line>
<non-fiscal_line type="line" id="16" value="56733"></non-fiscal_line>
<non-fiscal_line type="line" id="17" value="8002"></non-fiscal_line>
<non-fiscal_line type="line" id="18" value="7002"></non-fiscal_line>
<non-fiscal_line type="line" id="19" value="5676"></non-fiscal_line>
</receipt>
</packet>

```

### 70.5.3. In advance payment

```

<packet>
<receipt action="begin" mode="online"></receipt>
<in advance action="in advance" rate="A" description="Television set" value="100"></in advance>
<receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="100.00">
</packet>

```

### 70.5.4. In advance payment storno (reversal)

```

<packet>
<receipt action="begin" mode="online"></receipt>
<in advance action="in advance" rate="A" description="Television set" value="100"></in advance>
<in advance action="in advance_storno" rate="A" description="Television set" value="100"></in
advance>
<position name="bread" quantity="1" unit="" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" system_number="123456" cashbox_number="02" cashier="Adam Adam"
amount="1.00">
</packet>

```

### 70.5.5. Settlement of in-advance payment

```

<packet>
<receipt action="begin" mode="online"></receipt>
<position name="Television set " quantity="1" unit="" rate="A" price="500.00" recipe="" charge=""
plu="" description="" action="sale"/>
<in advance action="in advance_settlement" rate="A" description="Television set" value="100"
surcharge="400"></in advance>
<receipt action="close" system_number="123" cashbox_number="1" cashier="Jan Kowalski"
amount="400.00"/>
</packet>

```

## 70.5.6. Reversal (Storno) of in advance payment settlement

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="Television set " quantity="1" unit="" rate="A" price="500.00" recipe="" charge=""
plu="" description="" action="sale"/>
<in advance action="in advance_settlement" rate="A" description="Television set" value="100"></in
advance>
<in advance action="storno_in advance_settlement" rate="A" description="Television set"
value="100"></in advance>
<receipt action="close" system_number="123" cashbox_number="1" cashier="Jan Kowalski"
amount="500.00"/>
</packet>
```

## 71. Invoice examples

### 71.1. Invoice with 1 copy

```
<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
payment_name="cash"
payment_data="10-10-2013" receiver="" issuer="" copy_number="0" margines="no"
options_buyer_name="information" options_seller_name="no" space_for_signature="yes"
paid="" sale_date="10-10-2013">
<customer>receiver data 1</customer>
<option id="1"/><option id="2"/>
<option id="11"/></invoice>
<position name="fruit juice" quantity="2" unit="pcs" rate="A" price="2.00" recipe="" charge="" plu=""
description="" action="sale"/><invoice actjon="close" buyer="jan jan" amounta="4.00"
system_number="123" cashbox_number="02" cashier="Adam Adam"/>
</packet>
```

### 71.2. Invoice with many positions

```
<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
payment_name="cash"
payment_data="10-10-2013" receiver="" issuer="" copy_number="255" margines="no"
options_buyer_name="information" options_seller_name="no" space_for_signature="yes"
paid="" sale_date="10-10-2013">
<customer>receiver data 1</customer>
<option id="1"/>
<option id="2"/>
<option id="11"/>
</invoice>
<position name="fruit juice" quantity="2" unit="pcs" rate="A" price="2.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="beer no alcoh." quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="soda water" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="cookies" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pencil" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pen" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
```



```

<position name="express tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="black tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="green tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="butter" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="finger sticks" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="apple" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<invoice actjon="close" buyer="jan jan" amount="15.00" system_number="123"
cashbox_number="02"
cashier="Adam Adam"/>
</packet>

```

### 71.3. Invoice with percentage discount on the invoice

```

<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
payment_name="cash"
payment_data="10-10-2013" receiver="" issuer="" copy_number="255" margines="no"
options_buyer_name="information" options_seller_name="no" space_for_signature="yes"
paid="" sale_date="10-10-2013">
<customer>receiver data 1</customer>
</invoice>
<position name="fruit juice" quantity="2" unit="pcs" rate="A" price="2.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="beer no alcoh." quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="soda water" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="cookies" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pencil" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pen" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="express tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="black tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="green tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="butter" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="finger sticks" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="apple" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<invoice actjon="close" buyer="jan jan" amount="15.00" system_number="123"
cashbox_number="02"
cashier="Adam Adam"/>
<discpunt value="10%" name="SPECIAL" discount_id="1" action="discount"></discount>
</invoice>
</packet>

```

#### 71.4. Invoice with quota markup on the position

```
<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
payment_name="cash"
payment_data="10-10-2013" receiver="" issuer="" copy_number="255" margines="no"
options_buyer_name="information" options_seller_name="no" space_for_signature="yes"
paid="" sale_date="10-10-2013">
<customer>receiver data 1</customer>
</invoice>
<position name="fruit juice" quantity="2" unit="pcs" rate="A" price="2.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="beer no alcohol." quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="soda water" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="cookies" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="pencil" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<action="sale"><discount value="10.00" name="SPECIAL" discount_id="1"
action="markup"></discount>
</position>
<position name="pen" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="express tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="black tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="green tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="butter" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="finger sticks" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="apple" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<invoice actjon="close" buyer="jan jan" amount="25.00" system_number="123"
cashbox_number="02"
cashier="Adam Adam"/>
</invoice>
</packet>
```

#### 71.5. Invoice with quota discount to the subtotal

```
<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
payment_name="cash"
payment_data="10-10-2013" receiver="" issuer="" copy_number="255" margines="no"
options_buyer_name="information" options_seller_name="no" space_for_signature="yes"
paid=""
sale_date="10-10-2013">
<customer>receiver data 1</customer>
</invoice>
<position name="fruit juice" quantity="2" unit="pcs" rate="A" price="2.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="beer no alcohol." quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="soda water" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="cookies" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
```

```

<position name="pencil" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/></position>
<discount value="5.00" name="SPECIAL" discount_id="1" action="discount"></discount>
<position name="pen" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="express tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="black tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="green tea" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="butter" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="finger sticks" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="apple" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/></position>
<invoice action="close" buyer="jan jan" amount="4.00" system_number="123" cashbox_number="02"
cashier="Adam Adam"/>
</invoice>
</packet>

```

#### 71.6. Invoice with different forms of payment

```

<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
payment_name="cash"
payment_data="10-10-2013" receiver="" issuer="" copy_number="255" margines="no"
options_buyer_name="information" options_seller_name="no" space_for_signature="yes"
paid=""
sale_date="10-10-2013">
<customer>receiver data 1</customer>
<ld option="14"/>
</invoice>
<position name="Article 1" quantity="1" unit="pcs" rate="A" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<payment type="card" name="VISA" action="add" value="5.00"></payment>
<payment type="cheque" name="cheque" action="add" value="5.00"></payment>
<payment type="credit" name="credit" action="add" value="5.00"></payment>
<payment type="transfer" name="transfer" action="add" value="5.00"></payment>
<invoice action="close" amount="20.00" system_number="123" cashbox_number="1"
cashier="Adam Adam"/>
buyer="jan jan" />
</packet>

```

#### 71.7. Invoice with data and additional footer's lines

```

<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
payment_name="cash"
payment_data="10-10-2013" receiver="" issuer="" copy_number="255" margines="no"
options_buyer_name="information" options_seller_name="no" space_for_signature="yes"
paid=""
sale_date="10-10-2013">
<customer>receiver data 1</customer>
<customer>information 1</customer>
<customer>information 2</customer>
</invoice>
<position name="fruit juice" quantity="2" unit="pcs" rate="A" price="2.00" recipe="" charge="" plu=""
description="" action="sale"/>
<invoice action="close" buyer="jan jan" amount="4.00" system_number="123" cashbox_number="02"

```

```

cashier="Adam Adam"/>
<non-fiscal_line type="definition">
<line>abc line 1</line>
<line>abc line 2</line>
<line>abc line 3</line>
</non-fiscal_line>
</faktura>
</pakiet>

```

### 71.8. Invoice with additional non-fiscal part

```

packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
payment_name="cash"
payment_data="10-10-2013" receiver="" issuer="" copy_number="255" margines="no"
options_buyer_name="information" options_seller_name="no" space_for_signature="yes"
paid=""
sale_date="10-10-2013">
<customer>receiver data 1</customer>
<id option="1"/><id option="2"/>
<id option="11"/></invoice>
<position name="fruit juice" quantity="2" unit="pcs" rate="A" price="2.00" recipe="" charge="" plu=""
description="" action="sale"><invoice actjon="close" buyer="jan jan" amount="4.00"
system_number="123" cashbox_number="02"
cashier="Adam Adam"/>
<non-fiscal_line type="definition">
<line>line 1</line>
<line>line 2</line>
<line>line 3</line>
</non-fiscal_line>
<non-fiscal_line type="line" id="1" value="567"></non-fiscal_line>
< non-fiscal_line type="line" id="2" value="602"></non-fiscal_line>
<non-fiscal_line type="line" id="3" value="777"></non-fiscal_line>
<non-fiscal_line type="line" id="4" value="Kowalski"></non-fiscal_line>
<non-fiscal_line type="line" id="5" value="ABCD"></non-fiscal_line>
<non-fiscal_line type="line" id="6" value="987"></non-fiscal_line>
<non-fiscal_line type="line" id="7" value="28.09.2020"></non-fiscal_line>
<non-fiscal_line type="line" id="8" value="75757"></non-fiscal_line>
<non-fiscal_line type="line" id="9" value="NCASHIER"></non-fiscal_line>
<non-fiscal_line type="line" id="10" value="100.50"></non-fiscal_line>
<non-fiscal_line type="line" id="11" value="PLN"></non-fiscal_line>
<non-fiscal_line type="line" id="12" value="4.000000"></non-fiscal_line>
<non-fiscal_line type="line" id="13" value="54545"></non-fiscal_line>
<non-fiscal_line type="line" id="14" value="56"></non-fiscal_line>
<non-fiscal_line type="line" id="15" value="worker name"></non-fiscal_line>
<non-fiscal_line type="line" id="16" value="56733"></non-fiscal_line>
<non-fiscal_line type="line" id="17" value="8002"></non-fiscal_line>
<non-fiscal_line type="line" id="18" value="7002"></non-fiscal_line>
</invoice>
</packet>

```

### 71.9. Invoice options – Print the amount of pennies in words and do not print the recipient's data and the VAT INVOICE label

```

<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
payment_name="cash"
payment_data="10-10-2013" receiver="" issuer="" copy_number="255" margines="no"
options_buyer_name="information" options_seller_name="no" space_for_signature="yes"
paid=""

```

```

sale_date="05-06-2019">
<customer>receiver data 1</customer>
<id option="10"/>
<id option="15"/>
<id option="19"/>
</invoice>
<position name="fruit juice" quantity="1" unit="pcs" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="small fruit juice" quantity="1" unit="pcs" rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<invoice action="close" buyer="jan jan" amount="12.00" system_number="123"
cashbox_number="02"
cashier="Adam Adam"/>
</invoice>
</packet>

```

#### 71.10. Invoice cancellation

```

<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
payment_name="cash"
payment_data="10-10-2013" receiver="" issuer="" copy_number="0" margines="yes"
options_buyer_name="information" options_seller_name="no" space_for_signature="yes"
paid=""
sale_date="10-10-2013">
<customer>receiver data 1</customer>
<id option="1"/>
<id option="2"/>
<id option="11"/>
</invoice>
<position name="fruit juice" quantity="2" unit="pcs" rate="A" price="2.00" recipe="" charge="" plu=""
description="" action="sale"/>
<invoice action="cancel"/>
</packet>

```