NOVITUS

Description of
XML communication protocol

Version: 1.09.ENG (2023)

Warning !

It is forbidden to use characters with codes 127 (0x7F) and 34 (0x22) in the packet field values.

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.

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>
```

Now the example of the same packet with 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.

How to calculate checksum.

The XML packet checksum is calculated from the data between the **<packet>** and **</packet>** tags.
If there are, for example, line breaks and others, they should also be taken into account when calculating crc.

Example of calculating checksum in language C#:

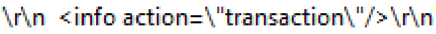
Input data

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

First, remove tags <packet> and </packet> and we are left with:

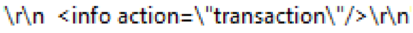
```
  <info action=”transaction”/>
```

And in character preview:



```
\r\n  <info action=\"transaction\"/>\r\n
```

Then we convert this data into a byte array (according to the encoding of windows 1250).
At the end, CRC32 is already counted and converted into hexadecimal.

String *write* includes: 

\r\n <info action=\"transaction\"/>\r\n

```
string hash = string.Empty;
C32 crc = new C32();

byte[] wys = Encoding.GetEncoding(1250).GetBytes(write);
foreach (byte b in crc.ComputeHash(wys, 0, wys.Length))
{
   hash += b.ToString("X2").ToLower();
}

For the above data crc is: bb1e3ec8
So packet with crc looks like:

<packet crc="bb1e3ec8">
  <info action="transaction"/>
</packet>
```

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

In response to the question:

```
<packet>
 <dle/>
</packet>
```

the printer sends back information in the following form:

```
<packet>
 <dle online="yes" papererror="no" printererror="no"/>
</packet>
```

where:

online – when the printer is in online mode - „yes”, otherwise - „no”.
papererror – when no paper occurs - „yes”, otherwise - „no”.
printererror – if occurs problem with printing mechanism - „yes”, otherwise - „no”.

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

In response to the question:

```
<packet>
 <enq/>
</packet>
```

the printer sends back information in the following form:

```
<packet>
<enq fiscal="yes" lastcommanderror="no" intransaction="no" lasttransactioncorrect
="yes"/>
</packet>
```

where:

Fiscal – if the printer is in the fiscal mode - „yes”, if in the training mode - „no”.
Lastcommanderror – if the last order was executed correctly - „yes” , otherwise - „no”.
Intransaction – if the printer is in transaction mode - „yes”, otherwise - „no”.
Lasttransactcorrect – when the last transaction was finalized correctly - „yes”, otherwise - „no”.

3. Read extended information about printer – DLE2 status.

```
<packet>
        <dle2/>
</packet>
```

Response:

```
<packet>
        <dle online="yes" papererror ="no" printererror="no” lasterror ="0"   printout_count ="125"
         receiptcount ="41" pending_report ="no" waiting_for_user_action ="no" in_menu="no"
         head_open="no" intransaction ="no" additional_lines_mode ="no" is_compressing_data
         ="no" is_printing ="no" paper_near_end ="no"  nonfiscal_printout_opened ="no"
         sending_ereceipt_inprogress ="no" lock_mode="nie" generating_periodical_report ="no"  />
</packet>
```

Where:

online – "yes" when the printer is online, otherwise "no".
papererror – "yes" if there is no paper, otherwise "no".
printererror – "yes" if a printing mechanism issue occurs, otherwise "no".
lasterror – Code of the last error.
printout_count – Total number of printouts on the printer.
receiptcount – Total number of receipts printed.
pending_report – "yes" if a report printout is in progress, otherwise "no".
waiting_for_user_action – "yes" if a user action is required on the printer, otherwise "no".
in_menu – "yes" if the printer is in menu mode, otherwise "no".
head_open – "yes" if the printer cover is open, otherwise "no".
intransaction – "yes" if a receipt/invoice transaction is open, otherwise "no".
additional_lines_mode – "yes" if in additional lines mode, otherwise "no".
is_compressing_data – "yes" if data compression is active, otherwise "no".
is_printing – "yes" if the printer is currently printing, otherwise "no".
paper_near_end – "yes" if the paper is running low, otherwise "no".
nonfiscal_printout_opened – "yes" if a non-fiscal printout is open, otherwise "no".
sending_ereceipt_inprogress – "yes" if an e-receipt is being sent, otherwise "no".
lock_mode – "yes" if the printer is in lock mode, otherwise "no".
generating_periodical_report – "yes" if a periodic report is being generated, otherwise "no".

4. Starting invoice.

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

```
<packet>
 <invoice       invoicetype=”invoice”action="begin"       number="120/2012"nip="1234567890"
      description="both" paymentname="cash" paymentdate="18-06-2019" recipient="" issuer=""
      copies="255" margins="yes" signarea="yes" customernameoptions-”info”
sellernameoptions="none" paidlabel=” paid by cash" selldate=”18-06-2019”
buyerlabel=”buyer” additionalinfo=”additional information” >
<customer>data of recipient 1</customer>
<option id=”1”/>
<option id=”2”/>
<option id=”11”/>
</invoice>
</packet>
```

where:

invoicetype – invoice type

1. invoice

action = “begin” – staring invoice.
number – invoice number (not required).
nip – Nip number of the buyer (not required)
        (size 0..56)

Nip – buyer’s NIP number (not required, in HD II online empty place should be replaced with space)
description – takes the following values (not required):

1. description = „both” – original and copy (default value).
2. description = “original” – only original.

paymentname – form of payment (not required)
(size 0..20)
paymentdate – date of payment (format: day – month – year)
(size 0..16)
recipient – recipient (not required)
(size 0..26)
issuer – issuer (not required)
(size 0..26)
copies – amount of copies – defaulty 0 (not required) 255 – no copy printed.
margins – margins – takes values „yes” or „no” (not required).
signarea – space for signature – takes values „yes” or „no” (not required).
customernameoptions – takes the following values:

1. customernameoptions = „info” – options of buyer name (for information only).
2. customernameoptions = „all” – name and information block (default value), invoice packet can’t include option **signarea**
3. customernameoptions = „none” – no information about buyer, please set **signarea=”no”.**

sellernameoptions – takes the following values:

1. sellernameoptions = „info” – options of seller’s name (for information only).
2. sellernameoptions = „all” – name and information block (default value), invoice packet can’t include option **signarea**
3. sellernameoptions = „none” – no information about seller, please set **signarea=”no”**

paidlabel – informative inscription about form of payment (not required).
selldate – date of sale (format: day– month – year, not required).
(size 0..16)

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”),
buyerlabel – buyer label:

1. buyer
2. recivier

additionalinfo – additional information to the invoice.
(size 0..512)

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" total="123.32" systemno="123" checkout="1" cashier="Jan"
buyer=”Ala” >
  </invoice>
</packet>
```

where:

buyer – buyer name
(size 0..40)
total – amount to pay,
systemno – system number (@ - if number is to be a QR – code, # - if number is to be a bar code),
(size 0..30)
checkout – checkbox number,
(size 0..8)
cashier – cashier name.
(size 0..31)

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),
(size 0..20)

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>
  <additional type="line" id="1" value="1234"></additional>
</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 invoice footer.

```
<packet>
 <additional type="definition">
                <line>abc line 1</line>
                <line>abc line 2</line>
                <line>abc line 3</line>
   </additional>
</packet>
```

Number of characters per line depends on printer type:

- 64 characters in HD Online, wide paper, normal font,
- 44 characters in HD Online, wide paper, enlarged font
- 44 characters in HD Online, narrow paper,
- 42 characters in Bono Online, Deon Online,

Warning!

There is only one footer change possible between daily reports.

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"   systemno="123"   checkout="1"   cashier="Jan"   total="123.32"
 charge="10.00" nip="123-456-78-90" >
  </receipt>
</packet>
```

where:

systemno – receipt number (@ - if it is to be a QR– code, # - if it is to be a barcode),
(size 0..30)
checkout – checkbox number (not required),
(size 0..8)
cashier – cashier name (not required),
(size 0..31)
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 – allows to enter buyer’s data in the receipt footer.
(size 0..16)

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="Occassional" 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),
(size 0..20)
action – takes the following values (not required):

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

Examples

9.2. Additional non-fiscal line.

```
<packet>
  <additional type="line" id="1" value="1234"></additional>
</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>
 <additional type="definition">
                <line>abc line 1</line>
                <line>abc line 2</line>
                <line>abc line 3</line>
    </additional>
</packet>
```

Warning!

There is only one footer change possible between daily reports.

Number of characters per line depends on printer type:

- 64 characters in HD Online, wide paper, normal font,
- 44 characters in HD Online, wide paper, enlarged font
- 44 characters in HD Online, narrow paper,
- 42 characters in Bono Online, Deon Online,

10. Payment in advance.

To send payment in advance you should send a packet:

```
<packet>
<rcpitemadvances    action="settlementrcpitemadvances”    description="description"    ptu="A"
value="2"
surcharge="1”>rcpitemadvances>
</packet>
```

where:
action:

- rcpitemadvances – in advance payment
- settlementrcpitemadvances – in advance payment settlement
- stornorcpitemadvances - in advance payment STORNO / reversal
- stornosettlementrcpitemadvances – in advance_payment STORNO/reversal_settlement

description – in_advance description
(size 0..60)
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="1.23" rate="4.32345678" mode=”rest”
  name="EUR">
   </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 = „account”,
7. type = „foreign”,
8. type = „cash”,
9. type = „mobile”
10. type = „token”, (on printout „ Voucher”),

action – takes values:

1. action=”add” – add payment,
2. action=”clear” – clear payment,

value – value of payment,
rate – exchange rate in the foreign currency (not required, default takes value 1),
(size 1..8)
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).
(size 0..15)

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" quantityunit="pcs" rate="A" price="1.23" total="2.46"
  recipe="" charge="" plu="" description="" action="sale">
  </item>
</packet>
```

where:

name – name of added goods,
(size 0..60)
quantity – quantity of goods,
(size 0..16)
rate – VAT rates,
(size 1)
price – price for unit,
(size 1..11)
total – calculated as product of parameter quantity and price (not required),
recipe – requested only in the pharmacy version,
(size 1..14)
charge – requested only in the pharmacy version,
(size 1..11)
plu – goods code ((@ - if code is to be a QR– code, # - if code is to be a barcode),
(size 1..31)
description – goods description (not required),
(size 0..118)
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" 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),
(size 0..20)
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:

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="out" 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,
(size 1..11)
type – takes the following values (not required):

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

quantity – quantity of packages (default 1, not required).
(size 1..11)

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" descid="1" type="subtotal" total="2" rate="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 „descid” is ignored, not required),
(size 0..20)
type – takes the following values (not required):

1. type = „subtotal” – discount to subtotal of receipt (default value),
2. type = „intransaction” – 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.
rate – VAT rates,
(size 1).

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 = „intransaction”, 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" lasterror="?" isfiscal="?" receiptopen="?"
         lastreceipterror="?" resetcount="?" date="?" receiptcount="?" cash="?" uniqueno="?"
         lastreceipt="?" lastinvoice="?" lastprintout="?"></info>
</packet>
```

The printer sends back values in the following form:

```
<packet>
    <info action="checkout" type="receipt" lasterror="0" isfiscal="yes" receiptopen="no"
         lastreceipterror="no" resetcount="0" date="08-10-2013" receiptcount="11" cash="444.76"
         uniqueno="ABC12345678" lastreceipt="11" lastinvoice="1" lastprintout="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,

lasterror – number of the last error,
isfiscal – is the printer fiscal,
receiptopen – is the receipt open,
lastreceipterror – has an error appeared during the last transaction,
resetcount – quantity of memory resettings,
date – current date,
receiptcount – quantity of receipts,
cash – cash,
uniqueno – unique number,
lastreceipt – number of the last receipt,
lastinvoice – number of the last invoice,
lastprintout – number of the last printout,
name – rate.

18. Information about fiscal memory.

In response to the question:

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

the printer sends back values in the following form:

```
<packet>
   <info action="fiscalmemory" fiscalmemorysize="1048576" recordsize="464" fiscal="1"
  uniqueno="ABC12345678" nip="123-456-78-90" maxrecordscount="2144" recordscount="7"
  maxreportscount="1830" reportscount="4" resetmaxcount="200" resetcount="0"
  taxratesprg_limit="30" taxratesprg="1" currencychangeprglimit="4" currencychangeprg="0"
  fiscalstartdate="dd-mm-yyyy hh:dd:ss" fiscalstopdate="dd-mm-yyyy hh:dd:ss"
  currencyname="PLN">
<ptu name="A">123.23</ptu>
<ptu name="B">123.23</ptu>
</packet>
```

where:

fiscalmemorysize – fiscal memory size (in bytes),
recordsize – 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,

uniqueno – unique number,
nip – NIP number,
maxrecordcount – maximum quantity of records,
recordscount – records quantity,
maxreports_count – maximum quantity of daily reports,
reportscount – quantity of daily reports,
resetmaxcount – maximum quantity of RAM resettings,
resetcount – quantity of RAM resettings,
taxratesprglimit – maximum quantity of PTU rate changes,
taxratesprg – quantity of PTU rate changes,
currencychangeprglimit – maximum quantity of currency changes,
currencychangeprg – quantity of currency changes,
fiscalstartdate – fiscalisation date,
fiscalstopdate – date of closing a fiscal mode,
currencyname – name of current currency.

19. Information of SD card.

In response to the question:

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

the printer sends back values in the following form:

```
 <packet>
  <info action="sdcard" type="external" state="unknow" />
 </packet>
```

where:

type – takes the following values (not required):
1. type = „external” – (default value),

state – card state. Takes the following values:
1. state = „unknow” – unknown,

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" nettotal="233" grosstotal="343" type="receipt" mode="online">
     <total name="A" tax="23" net="100" gross="123"></total>
  </info>
</packet>
```

where:

nettotal – sum of totalizers net,
grosstotal – 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="lasttransaction"></info>
</packet>
```

the printer sends back values in the following form:

```
<packet>
```

```
    <info action="lasttransaction" total="343" type="receipt" state="open" number="23"
      date="dd-mm-yyyy hh:dd" printoutno="123" checkout="1" cashier="Jan"
  systemno="1234" discountcalc="1" euroexchangerate="4.344" rest="30" eurorest="3">

      <item name="name" code="1234" quantity="4" rate="A" price="2.10" total="8.40"
        totaldiscount="12%" totalafterdiscount="7.20">
          <discount value="5" name="Occasional" total="2" type="discount"
          descid="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,
printoutno – printout number,
checkout_no – checkout number,
cashier – cashier name,
systemno – system number,
discountcalc – method of discount calculating,
euroexchangerate – exchange rate of the foreign currency,
rest – rest,
eurorest – rest in the foreign currency,
name – discount name,
value – discount value (value as amount or percent),
totaldiscount – the resultant total percent discount that results from given discounts on position.
totalafterdiscount – 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. Request of sending back data of last daily report.

In order to read data of last daily report you should send to the printer the following packet:

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

response:

```
<packet crc="E5B09460">
<info action="last_daily_report" uniqueno="ZAX2201013069" date="2023-05-09 12:14"
reportscount="1" grossA="0.00" grossB="0.00" grossC="0.00" grossD="0.00" grossE="0.00"
grossF="0.00" grossG="0.00" grosssum="0.00" receiptcount="0" printout_count="2"
canceledreceiptcount="0" grossAINV="0.00" grossBINV="0.00" grossCINV="0.00"
grossDINV="0.00" grossEINV="0.00" grossFINV="0.00" grossGINV="0.00"
grosssum_invoice="0.00" invoicecount="0" aborted_invoice_count="0" chargeA="0.00"
chargeB="0.00" chargeC="0.00" chargeD="0.00" chargeE="0.00" chargeF="0.00"
chargeG="0.00" medicamentpricing_count="0" cancelled_medicamentpricing_count="0"
lastreceipt="0" lastprintout="3" lastinvoice="0" nonfiscal_printout_count="2"
non_taxable_sale="0.00" ptuA="23.00" ptuB="8.00" ptuC="5.00" ptuD="0.00" ptuE="100.00"
ptuF="101.00" ptuG="101.00" currency="PLN" first_sale_date="2000-00-00 00:00"
last_sale_date="2000-00-00 00:00" />
</packet>
```

where:

action – takes value „last_daily_report”
uniqueno – unique number
date – date and time of report
reportscount – number of reports (report’s number)
grossA… grossG – gross sales in rate A… G for receipts,
grosssum – sum of gross sales for receipts,
receiptcount – number of receipts
printout_count – number of printouts
canceledreceiptcount – number of cancelled receipts
grossAINV… grossGINV – sales in rate A…G for invoices,
grosssum_invoice – sum of gross sales for invoices,
invoicecount – number of invoices,
aborted_invoice_count – number of cancelled invoices,
chargeA… chargeG – chargé in rate A…G for pharmacy version .
medicamentpricing_count – number of drug valuations,
cancelled_medicamentpricing_count – number of cancelled drug valuations,
lastreceipt – number of last receipt,
lastprintout – number of last printout,
lastinvoice – number of last invoice
nonfiscal_printout_count – number of non-fiscal printouts
non_taxable_sale – tax-free sales
ptuA… ptuG – vat rates A…G
currency – currency
first_sale_date – date of first sales
last_sale_date – date of last sales.

24. Information about device – general information

To collect information you should send a packet:

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

Response:

```
<packet>
<deviceinfo action="general" devicename="HD Online" version="1.00" moduleprinterversion="CITIZ"
systemname="OES" systemversion="1.00" displaycount="2" displaylen="64" ecopy="1"
fiscalsize="1048576"/>
</packet>
```

where:

devicename – device name

version – program version

moduleprinterversion- version of printing module

systemname – system name

systemversion – system version

displaycount – displays number

displaylen – display width

ecopy- is there a copy (1- there is, 0 – there isn't)

fiscalsize – fiscal memory size

25. Information about device – detailed for fiscal memory

To collect information, you should send packet:

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

Response:

```
<packet>
 <deviceinfo action="detailfiscalmemoryinfo" fiscalsize="1048576" recordlen="940" mode="0"
uniqueno="" nip="" eventlen="2108" recordcount="6" maxdailyreportcount="1830" dailyreportcount="0"
resetsmaxcount="200" resetscount="0" maxtaxratescount="30" taxratescount="0"
currency_change_max_count="20" currency_change_count="0" fiscalizationdate="2000-00-00
00:00:00" fiscalmodeclosedate="2000-00-00 00:00:00" currencyname="PLN"/>
</packet>
```

where:

fiscalsize – fiscal memory size in bytes

recordlen – record size in bytes

mode - operating mode (0 – non fiscal 1 – fiscal)

uniqueno – unique number

nip – nip (tax number)

eventlen – number of general events

recordcount – records number

maxdailyreportcount – Maximum number of daily reports

dailyreportcount – number of daily report

resetsmaxcount – Mximum number of RAM resettings

resetscount – number of RAM resettings

maxtaxrates_count – Maximum number of tax rates changes

taxratescount – number of tax rates

currency_change_max_count – Maximum number of currency changes

currency_change_count – number of currency changes

fiscalizationdate – fiscalisation date

fiscalmodeclosedate – date of fiscal mode closing

currencyname – name of current currency

26. Information about device – detailed for fiscal memory – extended version.

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

Response:

```
<packet>
<deviceinfo
action ="fiscal_memory_details"
fiscalsize="65011712"
recordlen="8208"
mode="0"
uniqueno="ZAX2201013069"
nip=""
eventlen="36"
maxeventlen="25484"
maxrecordcount="1390115"
recordcount="193"
maxdailyreportcount="2195"
dailyreportcount="13"
resetsmaxcount="200"
resetscount="4"
maxtaxratescount="30"
taxratescount="13"
currency_change_max_count="4"
currency_change_count="0"
fiscalizationdate="2000-00-00 00:00:00"
fiscalmodeclosedate="2000-00-00 00:00:00"
currencyname="PLN"
reduction_rates_changes_count ="16"
reduction_groups_changes_count ="0"
service_audit_count="0"
events_data_size="35132"
max_events_data_size="47304620"
last_report_date="2023-07-13 12:25:25"
last_record_date="2023-07-18 11:19:49"/>
</packet>
```

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

To collect information from device you should send packet:

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

Response:

```
 <packet>
 <deviceinfo action="batteryinfo" powersupply="yes" powersupplyvoltage="25"
poweraccumulatorvoltage="25" powerbatteryvoltage="25"/>
</packet>
```

where:

powersupply - connected power supply (yes, no)
powersupplyvoltage – power supply voltage
poweraccumulatorvoltage – accumulator voltage
powerbatteryvoltage – battery voltage

28. Information about device – information about printout status

To collect information you should send packet:

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

Response:

```
<packet>
<deviceinfo  action="printoutinfo"  printermoduleversion="CITIZ"  printerlen="64"  printerlenmm="80"
font="normal"                                   cutter="yes"                                   papercount="1"/>
</packet>
```

where:

printermoduleversion – version of printing module
printerlen – printout width in characters (Maximum number of characters in line)
printerlenmm – printout width in [mm]
font – increased or normal
cutter
papercount – amount of paper (0 – below critical level, 1 – above critical level, 2 – not operated)

29. Information about device – information about drawer status

To collect information you should send packet:

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

Response:

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

where:

drawer_state – drawer status - drawer_close – closed drawer, drawer_open – open drawer

30. Information about device – information about data base

To collect information you should send packet:

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

Response:

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

31. Information about device – status of accumulator and power supply

To collect information you should send packet:

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

Response:

```
<packet>
<deviceinfo action="powerinfo" powermode="25" powersupplyvoltage="25"
poweraccumulatorvoltage="25" powerbatteryvoltage="25" accumulatorstate="2" batterystate="5"
chargelevelaccumulator="15"/>
</packet>
```

where:

powermode – power mode - 0 – work on accumulator 1 – work on power supply
powersupplyvoltage – power supply voltage [mV]
poweraccumulatorvoltage – accumulator voltage [mV]
powerbatteryvoltage – battery voltage to RAM/clock support [mV]
accumulatorstate – 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

batterystate – Status of battery supporting RAM / clock - 0 – discharged / damaged 1 – should be charged 2 - battery status ok
chargelevelaccumulator – level of accumulator charging [%]

32. Information about device – fiscalisation (approval) parameters

To collect information about fiscalisation parameters you should send packet:

```
<packet>
<deviceinfo action="fiscal_parameters" mode="?" fiscalstartdate="?" nip="?" regnumber="?"
protectedmemory="?" typedevice="?" typenumbering="?" typeownership="?"></deviceinfo>
</packet>
```

Response:

```
<packet crc="25D76C64">
<deviceinfo action="fiscal_parameters" mode="fiscal" fiscalstartdate="2020-12-30 13:01:40" nip="522-
000-16-94" regnumber="2020/000010760" protectedmemory="ZBN1801017832.0001"
typedevice="general" typenumbering="daily" typeownership="leasing" />
</packet>
```

where:

mode – can take the following values:
- fiscal
- nonfiscal
- readonly

fiscalstartdate – fiscalisation (approval) date
nip – NIP
regnumber – registration number
protectedmemory
typedevice – can take the following values:
- general
- taxi
- pharmacy
- ticket
- gastronomic
- machines
- duty_free

typenumbering – can take the following values:
- daily
- global

typeownership – can take the following values:
- own

- leasing
- leasing_leasing
- rent
- other

33. Information about device – device operation time.

```
<packet>
  <info action="uptime”></info>
</packet>
```

Response:

```
<packet crc="5BCDC695">
<info action="uptime" cycles="487620031" seconds="1572" datetimestart="2021-09-20 11:14:24" />
</packet>
```

where :
cycles – number of cycles since application launching
seconds – number of seconds since application launching
datetimestart – date and time of application starting

34. Information about communication port

To collect information you should send packet:

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

Response:

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

35. 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="100 " 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 – numer of jpk to send
jpk_actual_send_no – number of currently send jpk

36. Interface data.

```
<packet>
<interface_info action = "get" type="0" interface_index ="1" interface_name = "aaa">
</interface_info>
</packet>
```

Response:

```
<packet crc="D5FC0D9D">
<interface_info action="get" type="0" interface_index="1" interface_name="wlan0" result="1" />
</packet>
```

Where:

action – takes values:
get – receiving information about interface
type – takes values:
0 – sending back information based on interface index
1 – sending back based on interface name
2 – sending back information about connecting a network cable

interface_index – takes values :
0 – eth0
1 – wlan0
2 – ppp0
Index is ignored when type takes value „1” or „2”

interface_name – interface name (ignored when type = 0 or 2)

result – takes values:
0 – no interface in system
1 – interface is raised

37. Controlling instructions.

37.1. Sound signal.

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

37.2. Paper feeding of quantity of lines.

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

where:

value – number of lines, which paper is to move forward. (0..20).
cutter – takes value "yes" – if switched on, "no" – switched off

37.3. Opening drawer.

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

37.4. Showing text on display.

```
<packet>
  <control action="display" line1="test1" line5="test5" line8="test8"></control>
</packet>
```

where:

Line1... 8 – text shown on display.

Number of characters:

- 22 characters in HD
- 16 characters in Deon Online,
- 15 characters in Bono Online,

37.5. 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

Number of characters:

- 22 characters in HD
- 16 characters in Deon Online,
- 15 characters in Bono Online,

37.6. Display clearing.

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

Where action takes values:

- ”clear_display” – clears the printer’s display.

37.7. Controlling the cutter and paper feed in open document.

To set cutter and paper feed during printing, send packet to the printer in the following form:

```
<packet>
  <paper action="set" paper_feed="1" cutter="0">
</ paper >
</packet>
```

where:

action – takes values:

1. action = „set”,

paper_feed – takes values:

-1 – unchanged value,
0 – no paper feed,
1 – paper feed after finishing printout

cutter – takes values:

-1 – unchanged value,
0 – no cutting,
1 – total cutting,
2 – partial cutting

Warning ! Packet should be sent after starting printout and before its finishing.

37.8. 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

38. 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 (Format: **dd-mm-yyyy**), not required, if you submit the current date you do not need to confirm with the keyboard).

39. 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" checkout="1" cashier="Jan"
   kind=""></report>
</packet>
```

where:

from, to – range of dates or report numbers (Format: **dd-mm-yyyy**),
checkout – checkbox number (not required),
size (0…8)
cashier – cashier name (not required),
size (0…31)
kind – takes the following values:

1. kind = „full" – report is full (fiscal document) from range of chosen dates or numbers (default value),
2. kind = „salesummary" – sale summary (non-fiscal document) from range of chosen dates or numbers,
3. kind = „monthlyfull" – full monthly report (fiscal memory),
4. kind = „monthlysummary" – monthly summary of sale (non-fiscal document),
5. kind = „billingfull" – full billing report from range of chosen dates,
6. kind = „billingsummary" – summary of billing report from range of chosen dates.

40. Shift report.

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

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

where:

checkout – checkbox number (not required),
size (0…8)
cashier – cashier name (not required),
size (0…31)
begin, size (0…15)
end , size (0…15)
income,
expense,
cash ,
payin, - money paid in
payout, - money paid out
balance – cash balance in ECR,
receiptcount – quantity of receipts,
size (0…5)
canceledreceiptcount – quantity of cancelled receipts,
size (0…5)
stornocount – quantity of reversals
size (0…5),
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”,
9. type = „other”
10. type = ”mobile”,
11. type = „bond”

container -> type – takes the following values:

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

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

41. Report of data from protected memory.

```
<packet>
  <report type="protectedmemory" from="04-06-2004" to="06-06-2006" kind="receipt"
format=”standard”>
</report>
</packet>
```

where:

from, to – range of numbers or dates (day-month-year),
format – takes the following values:
„nonfiscalprintout” – standard printout format. It’s also default value
„standard” – printed as standard printout
kind – takes the following values:

1. kind = „receipt” – receipts,
2. kind = „invoice” – invoices,
3. kind = „dailyreport” – daily reports,
4. kind = „nonfiscal” – non-fiscal printouts,

5. kind = „all” – all printouts,

42. Events report

To print events report you should send:

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

43. 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 numer 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

44. Report of service reviews.

To print report you should send a packet:

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

45. 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),
size (0…8)
name – cashier name (not required).
size (0…31)

46. Data readout of currently log-in cashier.

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

where action takes values:
-„state" - data of currently log-in cashier.

47. Pay-in, pay-out. and cash balance in ECR.

To program general NoviApi settings, you should send to the printer packet in the following form:

```
<packet>
  <cash action="payin" type="1" value="200.00" checkout="1" payer_name ="Jan Kowalski"
   number="123" signarea ="1">
</cash></ packet >
```

To pay-in or pay out cash from ECR's cash drawer you should send the following 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,

type – takes values:

0 – cash,
1 – card,
2 – cheque,
3 – bond,
4 – other,
5 – credit,
6 – customer account,
7 – currency (NOVITUS protocol), Voucher („NOVITUS accordable" protocol)
8 – transfer
9 – Mobile
10 – Voucher
99 – payment in currency € (EUR)

value – pay in value,
checkout – checkbox number (not required),
size (0…8)
payer_name – name of who pay in
number – number of receipt,
signarea – takes values:
0 – without space for signature,
1 – space for signature

48. Cash state in ECR.

To print cash state in ECR you should send to the printer packet in the following form:

```
<packet>
  <cash action="get" type="1" >
</cash>
</ packet >
```

where:

action – takes values:

1. action = „get",

type – takes values:
0 – non-resetting,
1 – resetting

49. 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>
```

50. Control of goods base.

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

```
 <packet>
  <dbcheck action="begin" mode="all" pluname="pluname" ptu="A" checkout="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,

pluname – name of goods,
size (0…40)
ptu – VAT rate,
size 1
checkout – checkbox number (not required),
size (0…8)
cashier – cashier name (not required)
size (0…31).

51. 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. „set” – setting configuration,
2. „get” – getting configuration
3. “clear” – resetting configuration

usb_modem_pid – Text in hex, eg. 12AB
size (4),
usb_modem_vid – Text in hex, eg. 12AB

size (4),
usb_modem_protocol - Text in hex, eg. 12AB
size (4),
usb_modem_interface – Text in hex, eg. 12AB
size (4),
usb_ex_pid - Text in hex, eg. 12AB
size (4),
usb_ex_vid – Text in hex, eg. 12AB
size (4),
usb_ex_type – przyjmuje wartości:
„acmx”
„usbx”
usb_ex_protocol – Tekst w hex, np. 12AB
size (4),
usb_ex_interface - Tekst w hex, np. 12AB
size (4),
usb_modem_type – takes values:
„acmx”
„usbx”

52. 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="discountcalculation">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 = „discountcalculation” – choosing 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.

53. Network interfaces readout.

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

Action:
- begin, initializes reading interfaces, sends back the first network interface
- next, sends back another interface. Successive interfaces can be received until the fields are not empty

Response:

```
    <packet crc="4FAA7FFE">
    <net_interface action="begin" name="eth0" mac="AA:BB:CC:DD:EE:FF"
    ip_addr="192.168.6.169" mask="255.255.255.0" gateway="192.168.111.212" is_wireless="1"
    active="1" />
</packet>
```

54. 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.

55. Reading date and time from device.

```
<packet>
        <clock action="get" ></clock>
</packet>
```

where:

action – takes the following values:

1. **action** = „**get**” – clock readout,

Response format:

```
<packet crc="39CEFD8E">
        <clock action="get" date=" dd-mm-yyyy hh:dd" />
</packet>
```

where:

dd – day,
mm – month,
yyyy – year,
hh – hour,
dd – minute.

56. 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" lines_count=”10”>
<line bold="no"  invers="no" align="center" index1="1" index2="0" index3="0" index4="0" index5="0"
index6="0" index7="0" text1="COMP S.A. Jutrzenki 116, 02-230 Warszawa" text2="" text3="" text4=""
text5="" text6="" text7=""></line>
<line bold="no"  invers="no" align="center" index1="0" index2="0" index3="0" index4="0" index5="0"
index6="0" index7="0" text1="" text2="" text3="" text4="" text5="" text6="" text7=""></line>
<line bold="no"  invers="no" align="center" index1="1" index2="0" index3="0" index4="0" index5="0"
index6="0" index7="0" text1="Oddział Nowy Sącz" text2="" text3="" text4="" text5="" text6=""
text7=""></line>
<line bold="no"  invers="no" align="center" index1="1" index2="0" index3="0" index4="0" index5="0"
index6="0" index7="0" text1="NOVITUS - Centrum Technologii Sprzedaży " text2="" text3="" text4=""
text5="" text6="" text7=""></line>
```

```
<line bold="no"  invers="no" align="center" index1="0" index2="0" index3="0" index4="0" index5="0"
index6="0" index7="0" text1="" text2="" text3="" text4="" text5="" text6="" text7=""></line>
<line bold="no"  invers="no" align="center" index1="7" index2="4" index3="0" index4="0" index5="0"
index6="0" index7="0" text1="Nawojowska" text2="118" text3="" text4="" text5="" text6=""
text7=""></line>
<line bold="no"  invers="no" align="center" index1="2" index2="3" index3="0" index4="0" index5="0"
index6="0" index7="0" text1="33-300" text2="Nowy Sącz" text3="" text4="" text5="" text6=""
text7=""></line>
<line bold="no"  invers="no" align="center" index1="0" index2="0" index3="0" index4="0" index5="0"
index6="0" index7="0" text1="" text2="" text3="" text4="" text5="" text6="" text7=""></line>
<line bold="no"  invers="no" align="center" index1="2" index2="3" index3="0" index4="0" index5="0"
index6="0" index7="0" text1="33-300" text2="Nowy Sącz" text3="" text4="" text5="" text6=""
text7=""></line>
<line bold="no"  invers="no" align="center" index1="0" index2="0" index3="0" index4="0" index5="0"
index6="0" index7="0" text1="" text2="" text3="" text4="" text5="" text6="" text7=""></line>
</header_online></packet>
```

where:

action – takes the following values:

2. action = „set" – set a header,
3. action = „get" – header readout,

lines_count – parameter is not required, (if missing, the number of lines is 8), it takes value 10, other values are ignored

bold – (not required):,
- „yes"
- „no"

invers (not required)
- „yes"
- „no"

align – takes the following values (not required);
1. align = „centered",
2. align = „left" – left-aligned (default value),
3. align = „right" – right-aligned,

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)

57. Online approval (fiscalisation)

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

```
<packet>
<fiscalisation action="trainingmode" typedevice="receiptprinter" nip="1111111111"
officename="office" servicename="service" servicemanname="serviceman" service_id="serw111"
taxofficecode ="1234" regnumber="111" tax_service="2222" typenumbering="global" /></packet>
```

where:

action – takes the following values:
trainingmode – training mode
fiscalization – approval
typedevice – takes values:
receiptprinter – receipt printer
ticketprinter – ticket printer
pharmaceprinter – pharmace printer

typenumbering – takes values:
global – global receipt numbering
daily_num – daily receipt numbering
officename- tax office data
size (0…120)
servicename – service point data
size (0…120)
servicemanname – service man data
size (0…40)
service_id – service man ID
size (0…40)
regnumer – registration number
size (0…15)
taxofficecode – tax office code
size (0…10)
tax_service – tax number of service point
size (0…13)

58. Update controlling.

```
<packet>
<update_software action="get" state="?" lasterror="?"  error_type="?" check_date="?"
version_number="?" sha256="?" />
</packet>
```

Format odpowiedzi:

```
    <packet>
<update_software action="get" state="" lasterror=""  error_type="" check_date=""
version_number="" sha256="" />
    </packet>
```

gdzie:

action - przyjmuje następujące wartości:
"get" - powoduje odebranie statusu aktualizacji programu
state - status aktualizacji
version_number - numer nowej wersji programu (bez kropki, czyli 123 to wersja 1.23).
check_date - data ostatniego poprawnego sprawdzenia.
lasterror - ostatni błąd.
error_type - rodzaj ostatniego błędu.
sha256 - suma kontrolna nowej wersji programu, w hex.

59. Update controlling.

```
<packet>
<update_software action="get" state="?" lasterror="?"  error_type="?" check_date="?"
version_number="?" sha256="?" />
</packet>
```

Response format:

```
<packet>
<update_software action="get" state="" lasterror=""  error_type="" check_date=""
version_number="" sha256="" />
</packet>
```

where:

action – takes the following values:
"get" - results in receiving the program update status
state – update status
version_number - number of new program version (without dot, so 123 is the version 1.23).
check_date – date of the last correct checking
lasterror – the last error
error_type – type of the last error
sha256 – checksum of new program version, in hex.

60. 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,
2. action = „unlock" – Removal of the service lock (password required if the lock was performed with a password)

date – date (day – month – year). If the date isn't set, the review is deleted.
description – description (not required)
size (0…30),
password – password (not required),
size (0…30),

In order to unlock, send the following packet:

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

61. Readout of update logs.

To read firmware update logs you should send packet:

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

where:

action – takes values:
begin – start downloading logs
next – downloading the next set of logs

Response:

```
<packet>
  <updatelog action="data" value=”text”/>
</packet>
```

where:
action – takes values:
data – next data to download
end – end of logs downloading

62. Readout of module logs

To read module logs you should send packet:

```
<packet>
<processlog action="begin" datetimestart="2018-01-01 23:59:00" datetimestop="2018-01-01 23:59:00"
level="LOG" mode=”SPI"/>
</packet>
```

where:

action – takes values:
begin – starting logs readout
next – downloading the next line of logs
datetimestart – date from which starts logs readout, format yyyy-mm-dd hh:mm:ss
datetimestop – final date of logs readout, format yyyy-mm-dd hh:mm:ss
level – logs level, takes values:
DEBUG
LOG
ERROR
CRITICAL
mode – module from which you want to read logs
ALL
General
TPM
http
BIOS
SPI
FW

Response:

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

where:
action - takes values
data – 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.

63. 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
size (0…15),

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>
```

64. 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
size (0…15)

65. 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"
  exchangerate="4.0001"></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),
size (3),
exchangerate – exchange rate (from: 0:00000001 to: 9999:99999999),

66. 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" exchangerate="4.0001" 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),
size (3),
exchangerate – exchange rate (from: 0:00000001 to: 9999:99999999),
type – takes the following values:

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

67. Setting PTU rates.

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

```
<packet>
  <taxrates action="set" checkout="1" cashier="Jan" confirmation=”1” date="YYYY-MM-DD">
  <ptu name="A">23%</ptu>
  </taxrates>
</packet>
```

where:

action – takes the following values:

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

checkout – checkbox number (not required),
size (0…8),
cashier – cashier name (not required),
size (0…31),
confirmation – takes values:
0 – programming rates with confirmation on printer,
1 – programming rates without confirmation on printer,
size (1),
date – date of rates modification, requested while changing rates without confirmation (in format YYYY-MM-DD),
name – name of PTU rate.
size (1),

Warning!

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

Example:

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

68. Automatic PTU rates change.

To set automatic PTU rates change you should send to the printer packet in the following form:

```
      <packet>
        <taxrates action="set" checkout="1" cashier="Jan" confirmation=”1” date="dd-mm-yyyy
hh:dd">
        <ptu name="A">23%</ptu>
        </taxrates>
      </packet>
```

where:

action – takes the following values:

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

checkout – checkbox number (not required),
size (0…8),
cashier – cashier name (not required),
size (0…31),
confirmation – takes values:
0 – programming rates with confirmation on printer,
1 – programming rates without confirmation on printer,
size (1),
date – date and hour of automatic rates modification (day-month-year hour:minutes),
name – name of PTU rate.
size (1),
Warning!

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

69. 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>
  <fiscalmemory action="get" from="dd-mm-yyyy hh:dd"></fiscalmemory>
</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 (format dd-mm-yyyy hh:dd),

Sending back records can take the following types:

69.1. Type = „report” – Record of daily report.

```
<packet>
  <fiscalmemory type="report" time="dd-mm-yyyy hh:dd" receiptcount="23" invoicecount="3"
   canceledreceiptcount="2" number="1" lastreceipt="4" lastinvoice="2" lastprintout="20"
   sale="12342.33" tax="20" totalsale="123213.12" totaltax="232.12" invoicesale="21.33"
   invoicetax="22.12" invoicetotalsale="1213.52" invoicetotaltax="3231.12" currencyname="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" totaltax="21142.23" invoicesale="2.20"
   invoicetax="0.23" invoicetotaltax="232.12"></ptu>

  </fiscalmemory>
</packet>
```

where:

time – time of record registering,
receiptcount – quantity of receipts,
invoicecount – quantity of invoices,
canceledreceiptcount – quantity of cancelled receipts
number – report number,
lastreceipt – number of last receipt,
lastinvoice – number of last invoice,
sale – sum of gross sale (receipts),
tax – sum of PTU tax (receipts),
totalsale – sum of sale increasingly (receipts),
totaltax – sum of tax increasingly (receipts),
invoicesale – sum of sale (invoices),
invoicetax – sum of tax (invoices),
invoicetotalsale – sum of sale increasingly (invoices),
invoicetotaltax – sum of tax increasingly (invoices),
currencyname – 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),
totaltax – value of tax increasingly in the rate (receipts),
invoicesale – sale in the rate (invoices),
invoicetax – value of tax in the rate (invoices),
invoicetotaltax – value of tax increasingly in the rate (invoices)
non_taxable - value not subject to PTU
tickets – number of tickets
normal – number of normal tickets

subsidy – number of half-price tickets with supplements
reduced – number of half-price tickets without supplements
abroad – number of abroad tickets

69.2. Type = „reset” – Resetting RAM memory.

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

where:

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

69.3. Type = „taxrates” – Programming VAT rates.

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

where:

time – time of record registering.

69.4. Type = „currencyset” – Record of setting a currency change.

```
<packet>
    <fiscalmemory type="currencyset" time="dd-mm-yyyy hh:dd" name="EUR"
  changetime="dd-mm-yyyy hh:dd" exchangerate="4.3231"></fiscalmemory>
</packet>
```

where:

time – time of record registering,
name – currency name,
changetime – date and time of currency change,
exchangerate – exchange rate.

69.5. Type = „currencychange” – Record of currency change.

```
<packet>
   <fiscalmemory type="currencychange" time="dd-mm-yyyy hh:dd" name="EUR"
  changetime="dd-mm-yyyy hh:dd" exchangerate="4.3231"></fiscalmemory>
</packet>
```

where:

time – time of record registering,
name – currency name,
changetime – date and time of currency change,
exchangerate – exchange rate.

69.6. Type = „cardclose” – Record of closing card.

```
<packet>
  <fiscalmemory type="cardclose" time="dd-mm-yyyy hh:dd" number="23" closetime="dd-
  mm-yyyy hh:dd"></fiscalmemory>
</packet>
```

where:

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

69.7. Type = „end” – End of data.

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

70. Request of sending back logs from port.

In response to the question:

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

the printer sends back information in the following form:

```
<packet>
  <log action="read" port="com1" date="dd-mm-yyyy hh:dd">dane....</log>
  <log action="write" port="com1" date="dd-mm-yyyy hh:dd">dane....</log>
</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 receiving / sending data (day – month – year hour:minutes).

71. Non-fiscal printout.

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

```
    <packet>
      <nonfiscalprintout systemno="123" nonfiscalheader="yes"
    edocument_send_mode =”0” left_margin =”yes” no_header =”yes” without_nonfiscal_mragins
=”yes”
    transaction_id =”1234567890” cutter=”0” paper_feed=”1” >
        <line type="line" bold="yes" invers="yes" center="yes" fontid="1" print_on=”copy”
          fontattr="big">line1</line>
        <line type="underline"></line>
        <line type="graphic">graphic number</line>
        <line>line 3</line>
      </nonfiscalprintout>
    </packet>
```

where:

print_on – takes values:
copy,
systemno – system number (not required),
edocument_send_mode– sending mode of e-document (optional argument):
0 – if there is no confirmation of e-print receiving by the server, print the printout and do not try to send the e-print,
1 – if there is no confirmation of e-print receiving by the server, keep trying to send e-print until successful,
2 - if there is no confirmation of e-print receiving by the server, print the printout and keep trying to send e-print until successful,
left_margin – left margin,
no_header – no printout header,
transaction_id – transaction id (text), (optional argument)
nonfiscalheader – describes whether the header "Non-fiscal printout" is to be printed. It takes values "yes" - printed, "no" – not printed.
without_nonfiscal_mragins – no margins „Non-fiscal",
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 – bold (reduces by half number of characters in line, not required),
invers – negative, takes values:
-yes
-no
center – centering, takes values:
-yes
-no
fontid – font (0 – basic font, 1- second font, 2 – third font (small), parameter not required),
fontattr – font attributes. It takes the following values (not required):

1. font_attribute = „big" – bigger font (reduces by half number of characters in line. Enlargement contained in font type),
2. fontattr = „high" – higher font,
3. fontattr = „bold" – widen font (reduces by half number of characters in line),
4. fontattr = „invers" – inversion,
5. fontattr = „very_high" – font 4 x higher,
6. fontattr = „very_width" – font 4 x wider,

cutter – takes values:
0 – off
1 – full cut
2 – partial cut

paper_feed – takes values:
0 – off
1 – on
.
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.

72. 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 = „readcrc” – 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

73. Readout of programmed graphics list.

In order to read indexes for programmed graphics you should send data packet as follow:

```
<packet>
  <graphic action="read_indexes"></graphic >
</packet >
```

where:

action – takes value:

1. action = „read_indexes” – read indexes, on which graphics were programmed,

Response:

```
<packet crc="C293F0A1">
<graphic action="read_indexes">
 0
 5
 6
</graphic>
</packet>
```

74. 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
size (0…256),
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>
```

75. Readout of dns addresses.

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

Action:
 - begin – initiates addresses readout. It sends back first DNS address
 - next – sends back the next DNS address. Successive addresses can be received until device sends back “none” in field value

Response:

```
    <packet crc="00835BC8">
        <dns_address action="begin" value="192.168.111.112" />
</packet>
```

76. 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 = „programme” – programming an animation,
name – animation name with extension
size (0…12),
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,
7. action = "readactive" – read name of active animation

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="3"  clockvisible="1"
  dispdate=”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

id – identifier of timesheet position
datetimestart – 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.
datetimestop – 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.
size (0…12),
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.

clocksettingvertical - placing the clock vertically (2 - up, 3- down, 4 - center),
clocksettinghorizontal – placing the clock horizontally (0 - left, 1- right, 4- center)
clockvisible – takes the following values:

1. clockvisible = „1” – clock visible
2. clockvisible = „0” – clock invisible.

dispdate = minimum time of advertisement displaying

77. Information about amount of printed paper.

To read amount of printed paper you should send to the printer the following data packet:

```
<packet>
  <paper action="read"></paper>
</packet>
```

where:

action – takes the following values:
1. action = „read” – read information,
2. action = „reset” – reset of printed paper counter,

78. Information about number of drawer openings.

To read number of drawer openings you should send to the printer the following data packet:

```
<packet>
  <drawercounter action="read"></drawercounter>
</packet>
```

where:

action – takes the following values:
1. action = „read” – read information,
2. action = „reset” – reset of drawer openings counter,

WARNING! Packet supported only by HD online printer.

79. Programming e-receipt settings.

In order to program e-receipt settings you should send to the printer the following data packet:

```
<packet>
<ereceipt action="set" server_address="" encryption="yes"
format="image_and_data" protocol="novitus" key="" mode="1" timeout="10"
message_timeout=”10” check_connectio=”yes” allow_change_address_menu=”0”
allow_change_address_packet =”0” auto_transaction_id =”0” allow_create =”0”
allow_send =”0” />
</packet>
```

where:

action – takes the following values:
- get – downloading e-receipt settings
- set – setting e-receipt configuration

type – takes values:

0 or none – basic data of e-receipt settings
1 – programming server address
2 – extended settings of e-receipt
3 – extended settings of e-receipt with possibility to set document types

server_address – submitting address of e-receipt server. Max. 200 characters.

encryption – takes the following values:
- yes
- no

format – takes the following values:
- image
- data
- image_and_data

protocol – takes the following values:
- novitus
- mf

key – submitting the key. Max. 64 characters.

mode – operating mode:

0 – when there is no confirmation of e-receipt receiving by the server, print the receipt and do not repeat attempts to send the e-receipt

1 – when there is no confirmation of e-receipt receiving by the server, repeat attempts to send the e-receipt until it is successful

2 - when there is no confirmation of e-receipt receiving by the server, print the receipt and repeat attempts to send the e-receipt until it is successful

3 – always print and put away to database to send it later

timeout - the number of milliseconds that the device waits for confirmation of e-receipt receiving the by the server. The maximum value is 30,000 ms - 30 seconds

message_timeout - Time for which e-receipt message will appear. The maximum value is 120,000 ms - 120 seconds

check_connection - Flag whether to check Internet connection before sending e-receipt. Takes values yes or no.

edocument_type – take values:

Bit-set types:

0x0001 – receipt,
0x0002 – cancelled receipt,
0x0004 – invoice,
0x0008 – cancelled invoice,
0x0010 – non-fiscal printout,
0x0020 – packaging settlement.

Field that use to read:

allow_change_address_menu – takes values:
0 – forbidden
1 - allowed
allow_change_address_packet – takes values:
0 – forbidden
1 – allowed
auto_transaction_id – takes values:
0 – no
1 - yes
allow_create – takes values:
0 – no
1 - yes
allow_send – takes values:
0 – no
1 - yes

80. Redout of e-receipt settings.

```
<ereceipt action="set" server_address="" encryption="yes" format="image_and_data"
protocol="novitus" key="" mode="1" timeout="10" message_timeout=”10” check_connectio=”yes”
allow_change_address_menu=”0” allow_change_address_packet =”0”  auto_transaction_id =”0”
allow_create =”0”  allow_send =”0”   />
</packet>
```

where:

action – takes the following values:
- get – downloading e-receipt settings
- set – setting e-receipt configuration

type – takes values:

0 or none – basic data of e-receipt settings
1 – extended data of e-receipt settings
2 – extended data of e- receipt settings with server settings
3 – extended data settings of e-receipt including types of send e-documents

server_address – submitting address of e-receipt server. Max. 200 characters.

encryption – takes the following values:
- yes
- no

format – takes the following values:
- image
- data
- image_and_data

protocol – takes the following values:
- novitus
- mf

key – submitting the key. Max. 64 characters.

mode – operating mode:

0 – when there is no confirmation of e-receipt receiving by the server, print the receipt and do not repeat attempts to send the e-receipt

1 – when there is no confirmation of e-receipt receiving by the server, repeat attempts to send the e-receipt until it is successful

2 - when there is no confirmation of e-receipt receiving by the server, print the receipt and repeat attempts to send the e-receipt until it is successful

3 – always print and put away to database to send it later

timeout - the number of milliseconds that the device waits for confirmation of e-receipt receiving the by the server. The maximum value is 30,000 ms - 30 seconds

message_timeout - Time for which e-receipt message will appear. The maximum value is 120,000 ms - 120 seconds

check_connection - Flag whether to check Internet connection before sending e-receipt. Takes values yes or no.

edocument_type – take values:

Bit-set types:

0x0001 – receipt,
0x0002 – cancelled receipt,
0x0004 – invoice,
0x0008 – cancelled invoice,
0x0010 – non-fiscal printout,
0x0020 – packaging settlement.

allow_change_address_menu – takes values:

0 – forbidden
1 - allowed

allow_change_address_packet – takes values:
0 – forbidden
1 – allowed

auto_transaction_id – takes values:
0 – no
1 - yes

allow_create – takes values:
0 – no
1 - yes

allow_send – takes values:
0 – no
1 - yes

81. Status of last e-receipt.

```
<packet>
        <ereceipt_state action="get"></ereceipt_state>
</packet>
```

Response:

```
<packet crc="9A9AB002">
        <ereceipt_state action="get" state="sended" printoutno="0" />
</packet>
```

Where:
state:

unknow – status unknown
sended – e-receipt was sent
printed – receipt was printed
print_and_send – receipt was sent and printed
added_to_send – e-receipt was added to send
print_and_added_to_send – receipt was sent and added to send
printed_and_new_kid_number_required – receipt was printed but it is required to get new kid number,

82. Readout of e-receipt.

```
<packet>
  <get_ereceipt mode=”1” search_mode =”0” number=”60” data_type =”0”  ></get_ereceipt>
</packet>
```

Response:

```
<packet>
  <get_ereceipt mode = ”1” search_mode =”0 number =”60” data_type=”0”  state= "1"
   total_size= "3089" position= "3089" size= "3089" data=
   "eyJhbGciOiJSUzI1NiIsIng1YyI6WyJNSUlESnpDQ0FnK2dBd0lCQWdJSUNOcXRES3RDRi
   tzd0RRWUp…"></ get_ereceipt >
</packet >
```

where:

- mode – takes values:

0 – start searching,
1 – next data,
2 – retransmission

- search_mode – takes values:

0 – search by JPKID,

1 – search by printout number,

- number - JPKID number or printout number, depending on selected mode,

- data_type – takes values:

0 – JWS,
1 – visualization,
2 – visualization without graphic,
3 – JWS + visualization,
4 – JWS + visualization without graphic

- state – takes values:,

0 – sending data,
1 – data end,
2 – no data,
3 – during processing,
4 – incorrect parameters,
8 – error, cannot start a new generation, first you must end the previous one,
9 – error.

- total_size – total data size to send,
- position – total size of sent data,
- size – size of sending data packet,
- data - document JWS data or printout visualization in format png, encoded in Base64.

Dla paragonu i paragonu anulowanego nagłówek JWS jest generowany w trybie - „dla dokumentów w postaci elektronicznej" (ma parametry: alg, jpkmetadata, x5c, eParagon.mf.gov.pl)
Dla faktury, faktury anulowanej, wydruku niefiskalnego nagłówek JWS jest generowany w trybie - „dla zbioru danych" (ma parametry: alg, jpkmetadata, jpkcertificate)

83. Readout of e-receipts base status.

```
<packet>
<ereceipt_base_info action="begin" protocol="0" status="0">
</ereceipt_base_info>
</packet>
```

where:
action – takes values:

count – number of documents in base,
begin – begin of readout,
next – the next record from base.

protocol – takes values:

0 – NOVITUS,
1 – MF,
2 – any

status – takes values:

0 – oczekujący na wysłanie i nie wydrukowany,
1 – oczekujący na wysłanie i wydrukowany,
2 – odrzucony przez serwer i nie wydrukowany,
3 – odrzucony przez serwer i wydrukowany,
4 – oczekujący na wysłanie,
5 – odrzucony przez serwer,
6 – dowolny

Odpowiedź (jeśli parametr „akcja" przyjmuje wartość „ilosc"):

```
<packet>
<ereceipt_base_info action="count" protocol="0" status ="0" recordscount="111" >
</ereceipt_base_info>
```

```
</packet>
```

Odpowiedź (jeśli parametr „akcja” przyjmuje wartość „poczatek” lub „kolejny”):

```
<packet>
<ereceipt_base_info action="begin" protocol="0" status ="0" jpkid =”111” >
</ereceipt_base_info>
</packet>
```

Gdzie:

action – przyjmuje wartości:

count – ilość dokumentów w bazie,
begin – początek odczytu,
next – kolejny rekord z bazy.
end – koniec danych,
bad_parameters – niepoprawne parametry wyszukiwania.

protocol – przyjmuje wartości:

0 – NOVITUS,
1 – MF,

status – przyjmuje wartości:

0 – oczekujący na wysłanie i nie wydrukowany,
1 - oczekujący na wysłanie i wydrukowany,
2 – odrzucony przez serwer i nie wydrukowany,
3 – odrzucony przez serwer i wydrukowany,

84. Programming parameters to generate e-receipt visualization.

```
<packet>
    <ereceipt_visualisation action="set" strategy="Z_FILTERED "
  compression_level =”Z_DEFAULT_COMPRESSION” filtration ="PNG_FILTER_NONE ">
  </ereceipt_visualisation >
  </packet>
```

Gdzie:

action – przyjmuje wartości:
- set – programowanie parametrów wizualizacji eParagonu,
- get – odczyt parametrów wizualizacji eParagonu,

strategy – przyjmuje wartości:

Z_DEFAULT_STRATEGY,
Z_FILTERED,
Z_HUFFMAN_ONLY,
Z_RLE,
Z_FIXED

compression_level – przyjmuje wartości:

Z_DEFAULT_COMPRESSION,
Z_NO_COMPRESSION,
Z_BEST_SPEED,
Z_BEST_COMPRESSION

filtration – przyjmuje wartości:

PNG_NO_FILTERS,

PNG_FILTER_NONE,
PNG_FILTER_SUB,
PNG_FILTER_UP,
PNG_FILTER_AVG,
PNG_FILTER_PAETH

Odpowiedź(jeśli parametr „action” przyjmuje wartość „get”):

```
<packet>
  <ereceipt_visualisation action="get" strategy="Z_FILTERED "
compression_level =”Z_DEFAULT_COMPRESSION” filtration ="PNG_FILTER_NONE "/>
</packet>
```

85. Transaction ID

```
<packet>
  <transaction_id action="add" id="123"  protocol=”0” mode=”0” />
</packet>
```

where:

id – transaction id
    size (0…100)

protocol – przyjmuje wartości:

0 – NOVITUS,
1 – MF

action:
- add – sets transaction ID
- clear – cancels transaction ID

mode – niewymagane, przyjmuje wartości:

0 – gdy brak potwierdzenia otrzymania e-paragonu przez serwer drukuj paragon I nie ponawiaj prób wysłania e-paragonu,
1 – gdy brak potwierdzenia otrzymania e-paragonu przez serwer nie drukuj i ponawiaj próby wysłania e-paragonu do skutku,
2 - gdy brak potwierdzenia otrzymania e-paragonu przez serwer drukuj paragon I ponawiaj próby wysłania e-paragonu do skutku,
3 – zawsze drukuj i próbuj wysłać do skutku.

86. Deleting rejected e-receipts from base.

W celu usunięcia odrzuconego edokumentu z bazy należy wysłać do drukarki pakiet w następującej formie:

```
<packet>
  <e_dokument action="delete" jpkid="1" >
</e_dokument >
</packet >
```

gdzie:

action – przyjmuje następujące wartości:
1. action = „delete”,
jpkid – numer jpkid kasowanego edokumentu.

87. Programming and readout of MF e-receipts HUB settings.

```
<packet>
```

```
    <ereceipt_mf_hub action="set" server_address="mf.gov.pl" size="204800">
     </ereceipt_mf_hub>

  </packet>
```

Gdzie:

action – przyjmuje wartości:
- set – ustawia adres serwera i rozmiar(opcjonalnie),
- get – odczytuje ustawienia

size – rozmiar eParagonu(domyślnie 200kb)

server_address – adres serwera()max 200 znaków,

Odpowiedź (jeśli parametr „akcja" przyjmuje wartość „odczytaj"):

```
<ereceipt_mf_hub action="get" server_address="mf.gov.pl" size="204800">
</ereceipt_mf_hub>
```

88. Pharmacy functions.

Warning:

All pharmacy functions in XML protocol are accessible since fiscal printer HDE 2.0 version.

88.1. Drug valuation.

To create drug valuation, the following data packet should be sent to the printer:

```
  <packet>
  <medicamentpricing action="set" name="sample drug" dose="0.5g" package="20pcs"
ptu="B" quantity="2" checkout="1" payment="cash" recipe="Rp. U30" refund="R3" issuer="Jan
Kowalski" date="2015-01-01" pharmacyqnt="" patient="Jan Biały" doctor="Adam Nowak"
refunder="KCh:17" province="malopolskie" value="29.99" price="29.99" taxvalue="2.10"
charge="11.23" limit="26.80" overlimit="3.19" extracharge="3.19" basecharge="20.00"
refunded="18.76">

  <line>additional line 1</line>
  <line>additional line 2</line>
  <line>additional line 3</line>

  </medicamentpricing>
  </packet>
```

where:

name – drug name,
dose – dose of valuated drug,
package – package of valuated drug,
ptu – name of valuated drug tax rate,
quantity – package quantity of a given drug,
checkout – checkout number,
payment – payment form,
recipe – recipe number,
refund – refund (description),
issuer – person who issues drug valuation,
pharmacyqnt – basic sales unit for a given drug,
patient – patient data,
doctor – doctor data,
refunder – drug refunder,
province – province name,
value – total drug value,
price – drug price,

taxvalue
charge – amount to be paid by patient,
limit – maximum refund value for the drug,
overlimit – difference of drug limit and value,
extracharge – quota for surcharge,
basecharge – charge after refund,
refunded – refunded amount for a given drug,

line– additional lines. They take on text values. Maximum number of additional lines is five.

88.2. Delete of drug valuation.

To delete drug valuation, the following packet should be sent to the printer:

```
<packet>
 <medicamentpricing action="delete" name="drug"></medicamentpricing>
</packet>
```

where:

name – name of drug being valuated.

88.3. Delete of valuation list.

To delete all valuation list, the following data packet should be sent to the printer:

```
<packet>
 <medicamentpricing action="deleteall"></medicamentpricing>
</packet>
```

88.4. Recipe valuation.

To make recipe valuation, the following data packet should be sent to the printer:

```
    <<packet>
    <medicamentpricing action="recipe" recipe="RC 90%" number="112/100" value="100" checkout="1"
cashier="Jan Kowalski" refund="16.35" extracharge="1.00" charge="25.00" margin="10"  taxalaborum="1"
patient="Jan Slaski" doctor= "Jerzy Nowak" refunder="MKCh"  province="malopolskie"
underlimitvalue="100" underlimitcharge="50">

    <medicament recipe="Rp. 100%" name="Acard" dose="0.025g" package="30 szt" quantity="2.000 op"
price="3.89" value="7.78" taxvalue="23"  refundedvalue ="0.00" limit="" limitvalue="" charge="3.89" >
                    <line>line1</line>
                    <line>line2</line>
                    <line>line3</line>
                    <line>line4</line>
                    <line>line5</line>
    </medicament>

    <medicament recipe="Rp. R" name="Lipanor" dose="0.025g"  package="30 szt" quantity="1.000 op"
price="32.96" value="32.96" taxvalue="19" refundedvalue ="0.00" limit=""  limitvalue="" charge="3.89">
                    <line>line1</line>
                    <line>line2</line>
                    <line>line3</line>
                    <line>line4</line>
                    <line>line5</line>
    </medicament>

    <line>line1</line>
    <line>line2</line>
    <line>line3</line>
    <line>line4</line>
```

```
        <line>line5</line>
        </medicamentpricing>
        </packet>
```

where:

receipt – receipt number
number – valuation number,
value – value of all drugs on recipe,
refund – refunded quota,
extracharge – quota for surcharge,
margin – margin put on customer,
taxalaborum – taxa laborum value,
underlimitvalue – value remains up to the limit,
underlimitcharge – payment up to the limit,

name – drug name,
dose – drug dose,
package – drug package,
quantity – number of packages,
price – package price,
<medicament>value – product of package number and price
taxvalue – percentage tax value,
refundedvalue – refunded quota,
limit – price limit for a given drug,
limitvalue – limit value,
charge - charge,

line – additional lines. It may appear below each drug from recipe (maximum 5) and at the end of valuation (maximum 5).

88.5. Valuation of drug list.

To make valuation of drug list, the following data packet should be sent:

```
        <packet>
          <medicamentpricing action="medlist" recipe="RC 90%" number="112/100" value="100"
refund="16.35" charge="25.00"  patient="Jan Slaski" doctor= "Jerzy Nowak" refunder="MKCh"
province="malopolskie" underlimitvalue="100" underlimitcharge="50">

        <medicament recipe="Rp. 100%" name="Acard" quantity="2.000 op" price="3.89" value="7.78"
taxvalue="23%"  refundedvalue ="0.00" limit="" limitvalue="" charge="3.89" ptu="A" >
        </medicament>

        <medicament recipe="Rp. R" name="Lipanor" quantity="1.000 op" price="32.96" value="32.96"
taxvalue="23%" refundedvalue ="0.00" limit=""  limitvalue="" charge="3.89" ptu="A" >
        </medicament>
        </medicamentpricing>
        </packet>
```

where:

recipe – recipe number
number – valuation number,
value – value of all drugs on recipe,
refund – refunded quota,
underlimitvalue - value remains up to the limit,
underlimitcharge - payment up to the limit,

name - drug name,
quantity - number of packages,
price - package price,
<medicament>value - product of package number and price
taxvalue - percentage tax value,
refundedvalue - refunded quota,
limit - price limit for a given drug,

limitvalue - limit value,
charge - charge,
ptu – tax rate name.

88.6. Valuation of recipe drug.

To make valuation of recipe drug, the following packet should be sent to the printer:

```
      <packet>
        <medicamentpricing action="made" name="Mixt. pepsini" recipe="RP" quantity="1" approved="Jan
Kowalski" assembledby="Andrzej Kos" issuer="Marcin Szpak" ptu="A" value="4.23" taxvalue="0.93"
taxalaborum="2.12" charge="5.00"  patient="Jan Slaski" doctor= "Jerzy Nowak" refunder="MKCh"
province="malopolskie" margin="2.00" >
      <element name="Ac. hydr. dil." quantity="3g" price="0.01">
          <line>line 1</line>
          <line>line 2</line>
          <line>line 3</line>
          <line>line 4</line>
          <line>line 5</line>
      </element>

      <element name="Pepsinum" quantity="5g" price="0.20">
          <line>line1</line>
          <line>line 2</line>
          <line>line 3</line>
          <line>line 4</line>
          <line>line 5</line>
      </element>

      <line>line 1</line>
      <line>line 2</line>
      <line>line 3</line>
      <line>line 4</line>
      <line>line 5</line>
      </medicamentpricing>
      </packet>
```

where:

name – drug name,
recipe – recipe number,
approved – data of person who approve a valuation,
assembledby - data of person who assembly a valuation,
issuer - data of person who dispense a drug,
ptu – tax rate name,
value – valuation value,
taxvalue – tax value,
province – province name,
margin - margin put on customer,

<element>name – component name,
<element>quantity – component quantity,
price – component price
line – additional lines. They can appear under each item of component (maximum 5), and in the valuation summary (maximum 5).

88.7. Order for recipe drug.

To make an order for recipe drug, the following data packet should be sent to the printer:

```
      <packet>
        <order  from="04-06-2004" to="06-06-2006" copies="1" number="123/42" cashier="Jan Kowalski"
patient="Jan Nowak" doctor="Adam Kowalski" charge="10.00">
      </order>
      </packet>
```

where

from – date of accepting order,
to – date of collecting order,
copies – number of printout copy,
number – order number,
cashier – cashier data,
patient – patient data,
doctor – doctor data,
charge – receipt number on which payment was confirmed. Charge must be in format: "Receipt No ...",

88.8. Cash report in pharmacy format.

To print cash report in pharmacy format, the following data packet should be sent to the printer:

```
        <packet>
        <report type="pharmaceutical" from="04-06-2004 06:30" to="06-06-2006 21:00" checkout="1"
cashier="Jan" startrcpno="1" endrcpno="10" grossA ="1000" grossB="100" grossC="1000" grossD="1000"
grossE="100" grossF="100" grossG="100" grosssum="3300" ptuA="10" ptuB="10" ptuC="10" ptuD="10"
ptuE="10" ptuF="10" ptuG="10" ptusum="40"  advancesign="" advance="100" startcashamount="10"
endcashamount="100" >
        </report>
        </packet>
```

where:

from , to – report timeframe
checkout – checkout number,
cashier – cashier data,
startcpno - number of first receipt in report,
endcpno - number of last receipt in report,
grossA, grossB, grossC, grossD, gross, grossF, grossG – sales amount in tax groups A-G,
grosssum – sum of all gross amounts,
ptuA, ptuB, ptuC, ptuD, ptuE, ptuF, ptuG - amount of input tax on sales amounts in given tax groups A-G.
ptusum – sum of all tax amounts,
advancesign – takes options "-" or " ",
advance – amount of in advance payment,
startcashamount – initial cash in report,
endcashamount – final cash in report.

Warning:

Pharmacy cash report can appear in two forms:
For defined five rates (A,B,C,D,G) – in packet should be sent only attributes for these five rates;
For defined seven rates (A,B,C,D,E,F,G) – in packet must appear attributes for all rates.

Example of packet for defined five rates:

```
        <packet>
          <report type="pharmaceutical" from="04-06-2004 06:30" to="06-06-2006 21:00" checkout="1"
cashier="Jan" startrcpno="1" endrcpno="10" grossA ="1000" grossB="100" grossC="1000" grossD="1000"
grossG="100" grosssum="3300" ptuA="10" ptuB="10" ptuC="10" ptuD="10"  ptuG="10" ptusum="40"
advancesign="" advance="100" startcashamount="10" endcashamount="100" >
        </report>
        </packet>
```

89. Ticket packets.

89.1. Ticket discounts.

```
<packet>
<reductions action="set">
<reduction reduction_index="1" group_index="1" value="20" name="Discount 1" />
<reduction reduction_index="2" group_index="2" value="30" name="Discount 2" />
<reduction reduction_index="3" group_index="4" value="35" name="Discount 3" />
<reduction reduction_index="4" group_index="8" value="50" name="Discount 4" />
</reductions>
```

```
</packet>
```

Discounts readout:

```
<packet>
<reductions action="get">
<reduction reduction_index="1" />
<reduction reduction_index="2" />
<reduction reduction_index="3" />
<reduction reduction_index="4" />
</reductions>
</packet>
```

Where:

action – takes values:
- get – reading programmed discounts
- set – programming discounts

group_index – index of the group to which discount will be assigned, a number from the range 1..10 specifying the group of discounts index to which the tax belongs

value – discount value
size (1-5)
name – discount name
size (1-40)

89.2. Groups of ticket discounts.

```
<packet>
<reductions_group action="set">
<group group_index="1" name="Pupil" />
<group group_index="2" name="Student" />
<group group_index="3" name="Pensioner" />
</reductions_group>
</packet>
```

```
<packet>
<reductions_group action="get">
<group group_index="1" />
<group group_index="2" />
<group group_index="3" />
<group group_index="4" />
<group group_index="5" />
<group group_index="6" />
<group group_index="7" />
<group group_index="8" />
<group group_index="9" />
<group group_index="10" />
</reductions_group>
</packet>
```

Where:

action – takes values:
- get – reading of programmed discount groups
- set – programming discount groups

group_index – group index, a number from the range 1..100 specifying the group of discounts index

name – group name
size (1-40)

89.3. Ticket receipt position.

```
<packet>
<ticket_item type="ordinary" passenger="Jan" kind="singlepass" name="ticket" price="10" quantity="1"
total="10" ptu="A" bus_stop_from="NSacz" bus_stop_to="Krakow" course_leave="123" course_return
="456" from="2020-01-02" to="2020-02-03">
</ticket_item>
</packet>
```

where:

type – takes following values:
- singlepass
- weekly
- weekend
- daily
- monthly
- quartely
- periodical

kind – takes following values:
- domestic
- crossborder
- abroad

passenger – passenger data
size (1-40)
name – ticket name
size (1-40)
price
quantity – how many
size (1-40)
total – total amount
ptu – tax symbol
size (1)
bus_stop_from – name of initial bus stop (not required)
size (1-40)
bus_stop_to – name of final bus stop (not required)
size (1-40)
course_leave – number of leave course (not required)
size (1-40)
course_return – number of return course (not required)
size (1-40)
from – initial validity date (not required)
size (8-10)
to – final validity date (not required)
size (8-10)
reduction_index – index of reduction taken into account when placing the item
text – text to ticket position
size (0-512)
charge - fee
refunded – takes values:
- yes
- no

89.4. Closing ticket position

```
<packet>
<ticket_item action="close" additionalinfo="here submit additional information" ></ ticket_item>
</packet>
```

Where:

Additionalinfo – additional information
size (0-64)

89.5. Readout of discount values for a given daily report

```
<packet>
<get_reduction_value action="get" number="3" ></get_reduction_value>
</packet>
```

where:

number – daily report number

89.6. Readout of discount values from daily reports range

Packet that initializing readout by the dates:

```
<packet>
<get_reduction_value action="get" from="2019-10-16" to="2019-10-16" ></
get_reduction_value>
</packet>
```

Packet that requests the further data:

```
<packet>
<get_reduction_value action="next"></ get_reduction_value>
</packet>
```

Packet that initializing readout by the numbers:

```
<packet>
<get_reduction_value action="get" from="5" to="8" ></ get_reduction_value>
</packet>
```

Packet that requests the further data:

```
<packet>
<get_reduction_value action="next"></ get_reduction_value>
</packet>
```

Where:

action – takes values:

-get

-next

from, to – data readout range, number or date in format: yyyy-mm-dd

89.7. Additional information to ticket receipt

```
<packet>
<info_ticket_receipt text="additional text"></ info_ticket_receipt>
</packet>
```

Where:

text – additional information

size (0-2048)

89.8. Periodical surcharge report

```
<packet>
<report typ="periodical_surcharge" from="2020-10-21" to="2020-10-25"></report>
</packet>
```

Where:

from, to – data readout range, number or date in format: yyyy-mm-dd

89.9. Monthly surcharge report

```
<packet>
<report typ="monthly_surcharge" from="2020-10-21" to="2020-10-25"></report>
</packet>
```

Where:
from, to – data readout range, number or date in format: yyyy-mm-dd

89.10. Daily surcharge report

```
<packet>
<report typ="daily_surcharge" from="2020-10-21" to="2020-10-25"></report>
</packet>
```

Where:
from, to – data readout range, number or date in format: yyyy-mm-dd

89.11. Report of discounts and discount groups

```
<packet>
<report typ="group_and_reduction" from="2020-10-21" to="2020-10-25"></report>
</packet>
```

Where:
from, to – data readout range, number or date in format: yyyy-mm-dd

89.12. Report of discount changes and discount groups

```
<packet>
<report typ="change_group_and_reduction" from="2020-10-21" to="2020-10-25"></report>
</packet>
```

Where:
from, to – data readout range, number or date in format: yyyy-mm-dd

89.13. Example of ticket receipt's position with discount

```
<packet>
<ticket_item type="singlepass" passenger="John" kind="domestic" name="ticket" price="10"
quantity="1" total="10" ptu="A" bus_stop_from="NSacz" bus_stop_to="Krakow" course_leave="123"
course_return="456" from="2020-01-02" to="2020-02-03">
<discount action="discount" value="20%"></discount>
</ticket_item>
</packet>
```

89.14. Example with multiple spaces following each other.

```
<packet>
  <nonfiscalprintout systemno="12345678">
<line type="line" font_id="1">NOVI &#32;&#32;&#32;&#32;&#32;&#32;&#32;TUS</line>
</nonfiscalprintout>
</packet>
```

To send one space, just insert it between words, e.g. „NOVIT US".
To send two or more spaces in succession, replace each subsequent space with a string „**&#32;**"
e.g.: „NOVIT **&#32;&#32;&#32;&#32;&#32;**US**"**

90. Graphic programming – example.

Graphic programming

```
<packet>
```

```
<graphic action="program" id = "1"
"type="all>012C003CFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC1FC7FFF80FF
F87FC3FE1FF0003FE1FC3FFF03FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0FC7FFE003FFC7FC7
FE1FF0003FE1FC3FFC01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0FC7FFC001FFC3FC7FE1FF0
003FE1FC3FF801FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0FC7FF83E0FFC3F87FE1FFF87FFE1F
C3FF0F9FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC47C7FF0FF0FFE3F8FFE1FFF87FFE1FC3FF0F
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC47C7FF0FF0FFE1F8FFE1FFF87FFE1FC3FF0FFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC63C7FF1FF87FE1F0FFE1FFF87FFE1FC3FF07FFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFC63C7FE1FF87FF1F1FFE1FFF87FFE1FC3FF80FFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFC71C7FE1FF87FF0F1FFE1FFF87FFE1FC3FFC03FFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFC71C7FE1FF87FF0E1FFE1FFF87FFE1FC3FFE01FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFC78C7FE1FF87FF8E3FFE1FFF87FFE1FC3FFFC0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FC78C7FE1FF8FFF863FFE1FFF87FFE1FC3FFFF0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC7C47
FF0FF0FFF843FFE1FFF87FFE1FC3FFFF0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC7C47FF0FF0F
FFC47FFE1FFF87FFE1FC3FFFF0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC7E07FF07C1FFFC07F
FE1FFF87FFF0F87FF3F0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC7E07FF8001FFFC07FFE1FFF8
7FFF0007FF001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC7F07FFC007FFFE0FFFE1FFF87FFF800
FFF003FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC7F87FFF01FFFFE0FFFE1FFF87FFFE03FFF80FF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
</graphic>

</packet>
<packet>
<graphic action="program" id = "1" type="next">
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFF</graphic>
</packet>

<packet>
<graphic action="program" id = "1" type= "end"></graphic>
</packet>
```

Graphic deleting

```
<packet>
<graphic action="delete" id="1"></graphic>
</packet>
```

91. Exemplary non-fiscal printouts.

91.1. Non-fiscal printout with QR code

```
<packet>
<nonfiscalprintout systemno="12345678">
<line type="line" fontid="1">line 1</line>
<line type="line" fontid="1">line 1</line>
<line type="line" fontid="1">line 1</line>
<line type="line" fontid="1">line 1</line>
<line type="line" fontid="1">line 1</line>
<line type="qrcode" bold="yes" center="yes" fontid="1" fontattr="big">NOVITUS</line>
</nonfiscalprintout>
</packet>
```

91.2. Non-fiscal printout with bar code.

```
<packet>
<nonfiscalprintout systemno="12345678">
<line type="barcode" bold="yes" center="yes" fontid="1" fontattr="big">NOVITUS</line>
</nonfiscalprintout>
</packet>
```

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

```
<packet>
<nonfiscalprintout systemno="12345678">
<line type="line" fontid="1">line 1</line>
<line type="line" fontid="1">line 1</line>
<line type="line" fontid="1">line 1</line>
<line type="line" fontid="1">line 1</line>
<line type="line" fontid="1">line 1</line>
<line type="qrcode" bold="yes" center="yes" fontid="1" fontattr="big">NOVITUS</line>
<line type="barcode" bold="yes" center="yes" fontid="1" fontattr="big">NOVITUS</line>
</nonfiscalprintout>
</packet>
```

91.4. Non-fiscal printout with empty line and underline

```
<packet>
<nonfiscalprintout systemno="123" nonfiscalheader="yes">
<line type="line" bold="yes" inwers="no" center="yes" fontid="1" fontattr="big">line1</line>
<line type="underline"></line>
<line>line3</line>
<line></line>
<line>line4</line>
</nonfiscalprintout>
</packet>
```

92. Examples – receipts

92.1. Payments

92.1.1. Receipt with many items

```
<packet>
<receipt action="begin" mode="online">
</receipt>
<position name="water" quantity="1" quantityunit="pcs" rate="A" price="100.00" recipe="" charge=""
plu="#" description="" action="sale"/>
<position name="juice" quantity="1" quantityunit="pcs" rate="A" price="150.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="sugar" ilosc="1" quantityunit="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%" descid="2" action="discount"></discount></position>
<position name="butter" quantity="1" quantityunit="pcs" rate="A" price="50.00" recipe="" charge=""
plu="#" description="" action="sale"/>
<position name="eggs" quantity="1" quantityunit="pcs" rate="A" price="50.00" recipe=""
charge=""plu="" description="" action="sale"/>
<position name ="beer" quantity="1" quantityunit="pcs" rate="A" price="50.00" recipe=""
charge=""plu="" description="" action="sale"/>
<discount value="10.00" name="special" action="markup"/></discount>
<position name="candy" quantity="1" quantityunit="pcs" rate="A" price="50.00" recipe=""
charge=""plu="" description="" action="sale"/>
<position name="fingersticks" quantity="1" quantityunit="pcs" rate="A" price="50.00" recipe=""
charge=""plu="" description="" action="sale"/>
<position name="bananas" quantity="1" quantityunit="pcs" rate="A" price="50.00" recipe=""
charge=""plu="" description="" action="sale"/>
<position name="nuts" quantity="1" quantityunit="pcs" rate="A" price="50.00" recipe=""
charge=""plu="" description="" action="sale"/>
<receipt action="close" systemnor="123" checkout="1" cashier="Jan Kowalski" total="655.00">
<discount value="39%" discid="6" action="discount"/></ discount >
</packet>
```

92.1.2. Goods sale

```
<packet>
<receipt action="begin" mode="online">
</receipt>
<position name="bread" quantity="1" quantityunit="pcs" rate="A" price="1.00" recipe=""
charge=""plu="" description="" action="sale"/>
<receipt action="close" systemno="" checkout="02" cashier="Adam Adam" total="1.00" />
</packet>
```

92.1.3. Payment with change

```
<packet>
<receipt action="begin" mode="online">
</receipt>
<item name="Chleb" quantity="1" quantityunit="pcs" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<payment type="cash" action="add" value="10.00" ></payment>
<receipt action="close" systemno="" checkout="02" cashier="Adam Adam" total="1.00" />
</packet>
```

92.1.4. Payment by card

```
<packet>
```

```
<receipt action="begin" mode="online">
</receipt>
<position name="bread" quantity="1" quantityunit="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" systemno="" checkout="02" cashier="Adam Adam" total="1.00" />
</packet>
```

92.1.5. Payment by card with change

```
<packet>
<receipt action="begin" mode="online">
</receipt>
<position name="bread" quantity="1" quantityunit="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" systemno="" checkout="02" cashier="Adam Adam" total="1.00" />
</packet>
```

92.1.6. Supporting forms of payment – many forms of payment

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="coca cola" quantity="1" quantityunit="" rate="A" price="10.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="pepsi" quantity="1" quantityunit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name ="fingersticks" quantity="1" quantityunit="" rate="C" price="30.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="kiwi" quantity="1" quantityunit="" rate="A" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name ="kiwi juice" quantity="1" quantityunit="" rate="A" price="10.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="water" quantity="1" quantityunit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name ="mineral water" quantity="1" quantityunit="" rate="C" price="30.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name ="sugar" quantity="1" quantityunit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name ="pepper" quantity="1" quantityunit="" rate="A" price="10.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="chili pepper" quantity="1" quantityunit="" rate="B" price="20.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="horseradish " quantity="1" quantityunit="" rate="C" price="30.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name ="tangerines " quantity="1" quantityunit="" 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="mobile" action="add" value="100.00" name="mobile"></payment>
<receipt action="close" systemno="" 123456” checkout="02" cashier="Adam Adam" total="300.00" />
</packet>
```

92.1.7. Supporting forms of payment – cancelling 3 from 4 payments

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

```
<position name="coca cola" quantity="1" quantityunit="" rate="A" price="10.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="pepsi" quantity="1" quantityunit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name ="fingersticks" quantity="1" quantityunit="" rate="C" price="30.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="kiwi" quantity="1" quantityunit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name ="kiwi juice" quantity="1" quantityunit="" rate="A" price="10.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="water" quantity="1" quantityunit="" rate="B" price="20.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name ="mineral water" quantity="1" quantityunit="" rate="C" price="30.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name ="sugar" quantity="1" quantityunit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name ="pepper" quantity="1" quantityunit="" rate="A" price="10.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="chili pepper" quantity="1" quantityunit="" rate="B" price="20.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="horseradish " quantity="1" quantityunit="" rate="C" price="30.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name ="tangerines " quantity="1" quantityunit="" 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="mobile" action="add" value="100.00" name="mobile"></payment>
< payment type="card" action="clear" value="20.00" name="card"></payment>
< payment type="cheque" action="clear" value="30.00" name="cheque"></payment>
< payment type="credit" action="clear" value="50.00" name="credit"></payment>
< receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="300.00" />
</packet>
```

92.2. Discounts / Markups

92.2.1. Sale with quota discount on the position

```
<packet>
<receipt action="begin" mode="online">
</receipt>
<position name ="water" quantity="1" quantityunit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10.00" name="special" discid="1" action="discount"></item>
<position name="soda water" quantity="1" quantityunit="" rate="A" price="5.00" recipe="" charge=""
plu="" description="" action="sale"/>
<receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="6.00">
</receipt></packet>
```

92.2.2. Sale with percentage markup on the position

```
<packet>
<receipt action="begin" mode="online">
</receipt>
<position name ="water" quantity="1" quantityunit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10%" name="special" descid="1" action="markup"></position>
<position name="soda water" quantity="1" quantityunit="" rate="A" price="5.00" recipe=""
charge="" plu="" description="" action="sale"/>
```

```
<receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="17.10">
</receipt></packet>
```

92.2.3. Sale with percentage discount on the position

```
<packet>
<receipt action="begin" mode="online">
</receipt>
<position name ="water" quantity="1" quantityunit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/></position>
<position name="soda water" quantity="1" quantityunit="" rate="A" price="5.00" recipe=""
charge="" plu="" description="" action="sale"/>
<receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam"  total=16.00">
<discount value="20%" name="special" discid="1"  action="discount"/>
</receipt></packet>
```

92.2.4. Sale with quota markup on the receipt

```
<packet>
<receipt action="begin" mode="online">
</receipt>
<position name ="water" quantity="1" quantityunit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/></position>
<position name="soda water" quantity="1" quantityunit="" rate="A" price="5.00" recipe=""
charge="" plu="" description="" action="sale"/>
<receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="16.00">
<discount value="5.00" name="special" discid="1" action=”markup”/>
</receipt></packet>
```

92.2.5. Discounts on the position and on sub-total and on the receipt

```
<packet>
<receipt action="begin" mode="online">
</receipt>
<position name ="water" quantity="1" quantityunit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10.00" name="special" discid="1" action="discount"></item>
<discount value="10%" name="" discid=""action="discount"/>
<position name ="water" quantity="1" quantityunit="" rate="A" price="11.00" recipe="" charge="" plu=""
description="" action="sale"/>
<discount value="10%" name="special" discid="1" action="discount"></item>
<discount value="1.80" name="special" discid="1" action="discount"/>
<receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="9.00">
<discount value="10%" name="special" discid="1"  action="discount"/>
</receipt></packet>
```

92.2.6. Percentage discount on goods in given PTU rate - A

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

```
<position name ="mineral water" quantity="1" quantityunit="" rate="C" price="30.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name ="sugar" quantity="1" quantityunit="" rate="D" price="40.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name ="pepper" quantity="1" quantityunit="" rate="A" price="10.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="chili pepper" quantity="1" quantityunit="" rate="B" price="20.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="horseradish " quantity="1" quantityunit="" rate="C" price="30.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name ="tangerines " quantity="1" quantityunit="" rate="D" price="40.00" recipe="" charge=""
plu="" description="" action="sale"/>
<discount value="20%" name="special" discid="1" rate=”A” action="discount"></discount>
<payment type="cash" action="add" value="300.00"></payment>
< receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam"
total="294.00"></receipt>
</packet>
```

92.2.7. Quota markup on goods in given PTU rate – C

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

92.3. Deposits

92.3.1. Container return on the receipt

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="soda water" quantity="1" quantityunit="pcs" rate="A" price="5.00" recipe=""
charge="" plu="" description="" action="sale"/>
<container action="sale" price="0.40" type="out" quantity="2"></container>
```

```
< receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="5.00"/>
</packet>
```

92.3.2. Reverse (storno) of container collected on the receipt

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="soda water" quantity="1" quantityunit="pcs" rate="A" price="5.00" recipe=""
charge="" plu="" description="" action="sale"/>
<container action="storno" price="0.40" type="out" quantity="1"></container>
< receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="5.00"/>
</packet>
```

92.3.3. Collections and refunds of container on the receipt

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

92.4. Reversals (Storno)

92.4.1. Cancelled receipt.

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

92.4.2. Receipt with storno

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name ="juice" quantity="1" quantityunit="pcs" rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="beer " quantity="1" quantityunit="pcs" rate="A" price="2.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name ="juice" quantity="1" quantityunit="pcs" rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="storno"/>
< receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="2.00"/>
</packet>
```

92.4.3. Receipt with storno after discount to sub-total

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name ="juice2" quantity="1" quantityunit="" rate="A" price="90.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name ="mango" quantity="1" quantityunit="" rate="A" price="50.00" recipe="" charge=""
plu="" description="" action="sale"/>
<discount value="10%" name="special" discount_id="1" action="discount">
```

```
<position name ="juice2" quantity="1" quantityunit="pcs" rate="A" price="90.00" recipe="" charge=""
plu="" description="" action="storno"/>
<receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="36.00"/>
</packet>
```

92.5. Others

92.5.1. Receipt with footer’s lines - 3 lines sys no. QR code

```
packet>
<receipt action="begin" mode="online"></receipt>
<position name="bread" quantity="1" quantityunit="" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="1.00">
<additional type="definition">
<line>line 1</line>
<line>line 2</line>
<line>@line3</line>
</additional>
</receipt></packet>
```

92.5.2. Receipt with additional non-fiscal part

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name ="bread" quantity="1" quantityunit="" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="1.00">
<additional type="definition">
<line>line 1</line>
<line>line 2</line>
<line>line 3</line>
</additional>
<additional type="line" id="1" value="567"/>
<additional type="line" id="2" value="602"/>
<additional type="line" id="3" value="777"/>
<additional type="line" id="4" value="Kowalski"/>
<additional type="line" id="5" value="ABCD"/>
<additional type="line" id="6" value="987"/>
<additional type="line" id="7" value="28.09.2020"/>
<additional type="line" id="8" value="75757"/>
<additional type="line" id="9" value="NCASHIER"/>
<additional type="line" id="10" value="100.50"/>
<additional type="line" id="11" value="PLN"/>
<additional type="line" id="12" value="4.000000"/>
<additional type="line" id="13" value="54545"/>
<additional type="line" id="14" value="56"/>
<additional type="line" id="15" value="worker name"/>
<additional type="line" id="16" value="56733"/>
<additional type="line" id="17" value="8002"/>
<additional type="line" id="18" value="7002"/>
<additional type="line" id="19" value="5676"/>
</receipt></packet>
```

92.5.3. In advance payment

```
<packet>
<receipt action="begin" mode="online"></receipt>
<rcpitemadvances action="rcpitemadvances" description="Television set" rate="A" value="100"/>
<receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="100.00"/>
</packet>
```

92.5.4. In advance payment storno (reversal)

```
<packet>
<receipt action="begin" mode="online"></receipt>
<rcpitemadvances action="rcpitemadvances" description="Television set" rate="A" value="100"/>
<rcpitemadvances action="stornorcpitemadvances" description="Television set" rate="A"
value="100"/>
<position name="bread" quantity="1" quantityunit="" rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="1.00" />
</packet>
```

92.5.5. Settlement of in-advance payment

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="Television set " quantity="1" quantityunit="" rate="A" price="500.00" recipe=""
charge="" plu="" description="" action="sale"/>
<rcpitemadvances action="settlementrcpitemadvances" description="Television set" rate="A"
value="100" surcharge=”400”/>
<receipt action="close" systemno="123456" checkout="02" cashier="Jan Kowalski" total="400.00"/>
</packet>
```

92.5.6. Reversal (Storno) of in advance payment settlement

```
<packet>
<receipt action="begin" mode="online"></receipt>
<position name="Television set " quantity="1" quantityunit="" rate="A" price="500.00" recipe=""
charge="" plu="" description="" action="sale"/>
<rcpitemadvances action="settlementrcpitemadvances" description="Television set" ptu="A"
value="100"/>
<rcpitemadvances action="stornosettlementrcpitemadvances" description="Telewision set" ptu="A"
value="100"/>
<receipt action="close" systemno="123456" checkout="02" cashier="Adam Adam" total="500.00" />
</packet>
```

93. Invoice examples

93.1. Invoice with 1 copy

```
<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
paymentname="cash"
paymentdate="10-10-2013" receiver="" issuer="" copies="0" margines="no"
customernameoptions="info" sellernameoptions="none" signarea="yes" paidlabel="" selldate="10-10-
2013">
<customer>receiver data 1</customer>
<option id="1"/>
<option id="2"/>
<option id="11"/>
</invoice>
<position name="fruit juice" quantity="2" quantityunit="pcs” rate="A" price="2.00" recipe="" charge=""
plu="" description="" action="sale"/>
<invoice action="close" total="4.00" system_number="123" checkout="02" cashier="Adam Adam"
buyer="jan jan" />
</packet>
```

93.2. Invoice with many positions

```
<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
paymentname="cash"
```

```
paymentdate="10-10-2013" recipient="" issuer="" copies="255" margins="no"
customernameoptions="info" sellernameoptions="none" signarea="yes" paidlabel="" selldate="10-10-
2013">
<customer>receiver data 1</customer>
<option id="1"/>
<option id="2"/>
<option id="11"/>
</invoice>
<position name="fruit juice" quantity="2" quantityunit="pcs” rate="A" price="2.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="beer no alcoh." quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="soda water" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="cookies" quantity="1" u=quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="pencil" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="pen" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="express tea" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="black tea" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="green tea" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="butter" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="finger sticks" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="apple" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<invoice actjon="close" total="15.00" system_number="123" checkout="02" cashier="Adam Adam"
buyer="jan jan" />
</packet>
```

93.3. Invoice with percentage discount on the invoice

```
<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
paymentname="cash"
paymentdata="10-10-2013" recipient="" issuer="" copies="255" margins="no"
customernameoptions="info" sellernameoptions="none" signarea="yes" paidlabel="" selldate="10-10-
2013">
<customer>receiver data 1</customer>
<option id="1"/>
<option id="2"/>
<option id="11"/>
</invoice>
<position name="fruit juice" quantity="2" quantityunit="pcs” rate="A" price="2.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="beer no alcoh." quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="soda water" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="cookies" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="pencil" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="pen" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
```

```
<position name="express tea" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="black tea" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="green tea" quantity="1" quantiyunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="butter" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="finger sticks" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="apple" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<invoice actjon="close" total="15.00" system_number="123" checkout="02" cashier="Adam Adam"
buyer="jan jan">
<discount value="10%" name="SPECIAL" discid="1" action="discount"/></invoice>
</packet>
```

93.4. Invoice with quota markup on the position

```
<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
paymentname="cash"
paymentdate="10-10-2013" receiver="" issuer="" copies="255" margins="no"
customernameoptions="info" sellernameoptions="none" signarea="yes" paidlabel="" selldate="10-10-
2013">
<customer>receiver data 1</customer>
<option id="1"/>
<option id="2"/>
<option id="11"/>
</invoice>
<position name="fruit juice" quantity="2" quantityunit="pcs” rate="A" price="2.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="beer no alcoh." quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="soda water" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="cookies" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="pencil" quantity="1" quantityunit="pcs" rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<discount value="10.00" name="SPECIAL" discid="1" action="markup"></position>
<position name="pen" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="express tea" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="black tea" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="green tea" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="butter" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="finger sticks" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="apple" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<invoice action="close" total="25.00" system_number="123" checkout="02" cashier="Adam Adam"
buyer="jan jan"/>
</packet>
```

93.5. Invoice with quota discount to the subtotal

```
<packet>
```

```
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
paymentname="cash"
paymentdate="10-10-2013" receiver="" issuer="" copies="255" margins="no"
customernameoptions="info" sellernameoptions="none" signarea="yes" paidlabel="" selldate="10-10-
2013">
<customer>receiver data 1</customer>
<option id="1"/>
<option id="2"/>
<option id="11"/>
</invoice>
<position name="fruit juice" quantity="2" quantityunit="pcs” rate="A" price="2.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="beer no alcoh." quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="soda water" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="cookies" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="pencil" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<discount value="5.00" name="SPECIAL" discid="1" action="discount"/>
<position name="pen" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge="" plu=""
description="" action="sale"/>
<position name="express tea" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="black tea" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="green tea" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="butter" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<position name="finger sticks" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe=""
charge="" plu="" description="" action="sale"/>
<position name="apple" quantity="1" quantityunit="pcs” rate="A" price="1.00" recipe="" charge=""
plu="" description="" action="sale"/>
<invoice action="close" total="10.00" system_number="123" checkout="02" cashier="Adam Adam"
buyer="jan jan" />
</packet>
```

93.6. Invoice with different forms of payment

```
<packet>
<invoice action="begin" number="120/2012" nip="1234567890" description="both"
paymentname="cash"
paymentdata="10-10-2013" receiver="" issuer="" copies="255" margins="no"
customernameoptions="info" sellernameoptions="none" signarea="yes" paidlabel="" selldate="10-10-
2013">
<customer>receiver data 1</customer>
<option id =”14”/>
</invoice>
<position name="Article 1" quantity="1" quantityunit="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" total="20.00" system_number="123" checkout="1" cashier="Adam Adam"
buyer=”jan jan” />
</packet>
```

93.7. Invoice with data and additional footer’s lines

```
<packet>
<invoice invoicetypee="invoice" action="begin" number="120/2012"nip="1234567890"
description="both"  paymentname="cash" paymentdate="18-06-2019" recipient="" issuer=""
copies="255" margins="yes" signarea="yes" customernameoptions="info" sellernameoptions="none"
paidlabel="paid cash" selldate="18-06-2019" buyerlabel ="buyer" additionalinfo="additional
information" >
<customer>receiver data 1</customer>
<option id="10"/>
<option id="15"/>
<option id="19"/>
</invoice>
<item name="Article 1" quantity="1" quantityunit="pcs" rate="A" price="10.00"recipe="" charge=""
plu="" description="" action="sale"/>
<invoice action="close" total="10.00" buyer="Ala" systemno="123" checkout="1" cashier="Jan
Kowalski">
<additional type="definition">
<line>line 1</line>
<line>line 2</line>
<line>line 3</line>
</additional>
</invoice>
</packet>
```

93.8. Invoice with additional non-fiscal part

```
<packet>
<invoice invoicetypee="invoice" action="begin" number="120/2012"nip="1234567890"
description="both"  paymentname="cash" paymentdate="18-06-2019" recipient="" issuer=""
copies="255" margins="yes" signarea="yes" customernameoptions="info" sellernameoptions="none"
paidlabel="paid cash" selldate="18-06-2019" buyerlabel ="buyer" additionalinfo="additional
information">
<customer>receiver data 1</customer>
<option id="10"/>
<option id="15"/>
<option id="19"/>
</invoice>
<item name="Article 1" quantity="1" quantityunit="pcs" rate="A" price="10.00"recipe="" charge=""
plu="" description="" action="sale"/>
<invoice action="close" total="10.00" buyer="Ala" systemno="123" checkout="1" cashier="Jan
Kowalski">
<additional type="definition">
<line>line 1</line>
<line>line 2</line>
<line>line 3</line>
</additional>
<additional type="line" id="1" value="567"></additional>
<additional type="line" id="2" value="602"></additional>
<additional type="line" id="3" value="777"></additional>
<additional type="line" id="4" value="Kowalski"></additional>
<additional type="line" id="5" value="ABCD"></additional>
<additional type="line" id="6" value="987"></additional>
<additional type="line" id="7" value="28.09.2020"></additional>
<additional type="line" id="8" value="75757"></additional>
<additional type="line" id="9" value="NCASHIER"></additional>
<additional type="line" id="10" value="100.50"></additional>
<additional type="line" id="11" value="PLN"></additional>
<additional type="line" id="12" value="4.000000"></additional>
<additional type="line" id="13" value="54545"></additional>
```

```
<additional type="line" id="14" value="56"></additional>
<additional type="line" id="15" value="worker name"></additional>
<additional type="line" id="16" value="56733"></additional>
<additional type="line" id="17" value="8002"></additional>
<additional type="line" id="18" value="7002"></additional>
</invoice>
</packet>
```

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

```
<packet>
<invoice invoicetype="invoice" action="begin" number="120/2012"nip="1234567890"
description="both"  paymentname="cash" paymentdate="18-06-2019" recipient="" issuer=""
copies="255" margins="yes" signarea="yes" customernameoptions="info" sellernameoptions="none"
paidlabel="paid cash" selldate="18-06-2019" buyerlabel ="buyer" additionalinfo="additional
information" >
<customer>receiver data 1</customer>
<option id="10"/>
<option id="15"/>
<option id="19"/>
</invoice>
<item name="Article 1" quantity="1" quantityunit="pcs" rate="A" price="10.00"recipe="" charge=""
plu="" description="" action="sale"/>
<invoice action="close" total="10.00" buyer="Ala" systemno="123" checkout="1" cashier="Jan
Kowalski"/></packet>
```

93.10. Invoice cancellation

```
<packet>
<invoice invoicetype="invoice" action="begin" number="120/2012"nip="1234567890"
description="both"  paymentname="cash" paymentdate="18-06-2019" recipient="" issuer=""
copies="255" margins="yes" signarea="yes" customernameoptions="info" sellernameoptions="none"
paidlabel="paid cash" selldate="18-06-2019" buyerlabel ="buyer" additionalinfo="additional
information" >
<customer>receiver data 1</customer>
<option id="1"/>
<option id="2"/>
<option id="11"/>
</invoice>
<item name="Article 1" quantity="1" quantityunit="pcs" rate="A" price="100.00"recipe="" charge=""
plu="" description="" action="sale"/>
<invoice action="cancel"></invoice></packet>
```