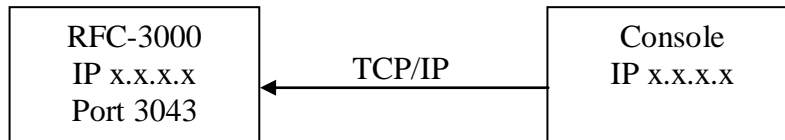


RFC-Console interconnection protocol

Both systems will communicate using the TCP/IP protocol. RFC will be the server. Console will be the client connecting to the server to port 3043, as shown below:



RFC will reply on the same socket to Singular.

Protocol definition:

' ': ASCII characters ('0'=binary byte of 30H)

“ ”: Binary character (30 means '0')

< >: Mandatory value

1. From RFC-3000 to Console:

- ***Confirm Transaction (authorize pump):***

'CT'<Pump#><Grade><Tran type><Rate type>

<Qty[1]-cash>|<Amt[1]-cash>|<Qty[1]-credit>|<Amt[1]-credit>|

.

<Qty[n]-cash>|<Amt[n]-cash>|<Qty[n]-credit>|<Amt[n]-credit>|

'_'<IDBOX_string>

Pump#: 2 chars (pump #3='03', pump #12 = '12')

Grade: 1 binary byte. Bit 0=nozzle 1 allowed, bit 1=nozzle 2 allowed, etc.

Rate type: '1'=Cash, '2'=Credit, '3'=Both

Tran type: '1'=Euromat, '3'=Attendant

Qty: Quantity limit for refuel (string representation of float number)

Amt: Amount limit for refuel (string representation of float number)

These lines are repeated for each nozzle in the pump (n= number of nozzles in pump)

'_': underscore character

NOTE:

- In Attendant transaction the values will be:

TranType = '3', RateType='1'.

In Euromat transaction the values will be :

TranType = '1', Ratetype='2'.

- The decimal separator for numbers (amounts and volumes is the '.' character)

Example:

CT03"5"1220.00|60.00|0|0|30.00|70.00|_7700025900029114

"5": binary 5 (0x5) character

means:

CT to pump 03. This pump has 3 nozzles, the first and the third are enabled. Therefore,

bit0=1, bit1=0, bit2=1: byte=5.

TranType = '1', Ratetype='2'

First nozzle is enabled to 20 liters, 60\$.

Second nozzle is enabled to 0 liters, 0\$(values ignored).

Third nozzle is enabled to 30 liters, 70\$.

IDBOX_string is the IDBOX identification # received

- **Deny Transaction (pump not authorized):**
'DT030' (Transaction denied- no 'ft' at the end).
Deny transaction (pump = 03)
- **Transaction details (fuel termination):**
The string structure is described in ts_eng.DOC, NULL terminated
- **Pause fuel (stop fuel flow):**
'PF'<pump#, 2 characters, leading zeros>
Example: 'PF03' – pause fuel on pump 3
- **Resume fuel:**
'RF'<pump#, 2 characters, leading zeros >
Example: 'RF03' – resume fuel on pump 3

2. From Console to RFC-3000:

- **Nozzle lifted:**
'nz'<Pump#><Nozzle status>
Pump#: 2 chars (pump #3='03', pump #12 = '12')
Nozzle status: '0'= nozzle reposed (in pump), '1'-'8'= Specifies nozzle extracted
- **Fuel Terminated:**
'ft'<Pump#>'0'<Nozzle#><Rate type><Price per liter>|<Total amount>|<Quantity>
Pump#: 2 chars (pump #3='03', pump #12 = '12')
Nozzle#: '1'-'8'
Rate type: '1'=Cash, '2'=Credit, '3'=Both
Price per liter/Total amount/Quantity: string representation of float number

Example:
ft080221.02|21.80|21.37
ft080221.02|0|0 (means a zero transaction)

Available sequence example:

Console: nz031
RFC: CT0351220.00|60.00|0|0|30.00|70.00|_7700025900029114
Console: nz030
Console: ft030121.02|21.80|21.37|
RFC: 137.00 142.860 KOK 5179 1 11 55664 12:32
08/11/06 Diesel ěăñ 00 ěÄääö 110155 7700025900029114
00904918 52007367 0.959... "0"