# COMMUNICATIONS PROTOCOL FOR CONCENTRADOR / 386 SCALE

*VERSION CONCENSC105.29 and later versions*
*(HSC105.014 June 1999)*

*PRELIMINARY VERSION*
*Version HNG.001o October-1998*

# TABLE OF CONTENTS


**1. READER'S GUIDE....6**

**2. COMMUNICATION FORMAT....6**

**3. LOCAL NETWORK (HOST) COMMUNICATION FORMAT....8**

**4. SPECIFIC MESSAGES FOR LOCAL/SOCKETS NETWORK....11**

4.1 SPECIFIC MESSAGES FOR LOCAL/SOCKETS NETWORK....11

*4.1.1 #<prf>::='/',47d,2Fh-FILE PROTOCOL....11*

**5. MESSAGES....14**

5.1 #<PRT>::='0',48D,30H-PROTOCOL (NOT 386)....14

*5.1.1 <act>::='0',48d,30h.Request of present protocol ....14*

*5.1.2 <sop>::='1',49d,31h.s Request for supported protocol (Not 8031)....15*

*5.1.3 <chg>::='2',50d,32h.Request for change of protocol.(Not 8031)....15*

5.2 #<EXC>::='1',49D,31H-EXCLUSIVE (NOT 8031) (NOT 386)....15

*5.2.1 <grp>:='0',48d,30h.Request of exclusive code....15*

*5.2.2 <qry>::='1',49d,31h.Exclusive message (addressed to a concrete network equipment)....15*

*5.2.3 <dta>::='2'.50d.32h Exclusive data (concrete equipment)....16*

5.3 #<SYS>::='2',50D,32H-SYSTEM....16

*5.3.1 <act>::='0',48d.30h.Programme updating (Not 8031) (Not 386)....16*

*5.3.2 <bor>::='1',49d,31h.Clearing internal databases .(Not 8031)....16*

*5.3.3 <kil>::='2',50d,32h.Switching off the equipment.(Not 8031) (Not 386)....17*

*5.3.4 <ref>::='3',51d,33h.Programme reference request (Not 8031) (Not 386)....17*

*5.3.5 <?network>::='4',52d,34h.Request of network state. (Not 8031)....18*

*5.3.6 <fmt>:='6',54d,36h Programming date and time format.(Not 8031) (Not 386)....18*

*5.3.7 <?fmt>::='7',55d,37h. Request of date and time format....19*

*5.3.8 <dte>::='8',56d,38h. Programming date and time ....19*

*5.3.9 <?dte>::='9',57d,39h.Request for date and time....19*

*5.3.10 <cpf>::=':'58d,3ah.Programming country code and manufacturer (Not 386)....19*

*5.3.11 <?cpf>::=';',59d,3bh.Request for country code and manufacturer (Not 386)....20*

*5.3.12 <?conf>::='<',60d,3ch. Request for configuration....20*

*5.3.13 <piva>::='=',61d,3dh. Programming types of VAT. ....21*

*5.3.14 <?civa>::='>',62d,3eh. Consulting the VAT types. ....21*

*5.3.15 <bck>::='?',63d,3fh. Backup of data files to LOCAL NETWORK WORK (Not 386)....21*

*5.3.16 <res>::=' ',64d,40h. Recuperating data files of LOCAL NETWORK unit (Not 386)....22*

*5.3.17 <sby>::='A',65d,41h. Modifying stanby (Not Concentrador) ....22*

*5.3.18 <?sby>::='B',66d,42h. Consulting standby (Not Concentrador) ....23*

*5.3.19 <vat>::='C',67d,43h. Programming scales VAT types (Not Concentrator) ....23*

*5.3.20 <?vat>::='D',68d,44h. Consulting the types of VAT of the scales (Not Concentrator) ....23*

*5.3.21 <prg>::='E',69d,45h. Change programming host activation to all (Not Concentrator)....23*

*5.3.22 <?prg>::='F',70d,46h. Consulting programming host activation to all (Not Concentrator)23*

*5.3.23 <borg>::='H',72d,47h.Clearing internal databases by groups. ....24*

5.4 #<SEC>::='3',51D,33H-SECTIONS (FOR CONCENTRATOR)....24

*5.4.1 prg>:='0',48d,30h.Create/Modify section,subsection,family (Not 8031)....24*

*5.4.2 <bor>:='1',49d,31h.Clearing section, subsection, family (Not 8031)....25*

*5.4.3 <?sec>::='2',50d,32h.Consulting information section, subsection, family (Not 8031) ....25*

*5.4.4 <pns>::='3',51d,33h. Programming number or sections (not Pc)....26*

*5.4.5 <?exi>::='4',52d,34h. Consulting if there is a section (not Pc)....26*

*5.4.6 <pvs>::='5',52d,34h. Programming VAT of section....26*

*5.4.7 <?pvs>::='6',53d,35h. Consulting VAT or the sections....26*

5.5 #<SEC>::='3',51D,33H-SECTIONS (COUNTERS). (FOR A SYSTEM OF SCALES)....26

*5.5.1 <prg>:='0',48d,30h.Create/Modify sections....26*

*5.5.2 <?sec>::='2',50d,32h.Consulting section or counter....27*

5.6 #<EQP>::='4',52D,34H-EQUIPMENTS....27

*5.6.1 <prg>:='0',48d,30h.Creation,Modification of equipment for Concentrator....27*

*5.6.2 <prg>:='0',48d,30h.Creation , Modification of Equipment for System Scales ....28*

*5.6.3 <eli>::='1'.49d.31h.Eliminating an equipment (Not 8031) ....28*

*5.6.4 <network>::='2',50d,32h.Connection/Disconnection of an equipment (Not 8031)* ..... *28*
*5.6.5 <pri>::='3',51d,33h.Change of priority of an equipment .(Not 8031) (Not 386)* ..... *29*
*5.6.6 <?sts>::='4',52d, 34h. Request for the status of an equipment* ..... *29*
*5.6.7 <?num>::='5',53d, 35h. Request for programmed number of equipment (Not 8031)* ..... *30*
*5.6.8 <act>::='6',54d, 36h.Order of programming from one scale to others (Only 386)* ..... *30*
*5.6.9 <?act>::='7',55d, 37h.Query of programming from one scale to others (Only 386)* ..... *30*
*5.6.10 <?ver>='8,56d, 38h. Consulting version and configuration data* ..... *30*
5.7 #<BAL>::='5'.53D.35H PROGRAMMING FOR SCALES (NOT 386) ..... 32
*5.7.1 <kbd>::='0',48,30h.Programming of direct keys of a scale or section.* ..... *32*
*5.7.2 <?kbd>::='1',49d,31h. Request of keys of a scale or section* ..... *33*
*5.7.3 <cbr>::='2', 50d, 32h.Programming of bar codes in scales or sections networkwork* ..... *33*
*5.7.4 <?cbr>::='3'.51d,33h.Request for codes in scales and section networkwork* ..... *34*
*5.7.5 <loc>::='4'.52d,34h. Order of requesting scales totals in local (Not 8031)* ..... *34*
*5.7.6 <kbd1>::='5',53d,35h. Programming of direct keys table of scale or section.* ..... *34*
*5.7.7 <?kbd1>::='6',54d,36h. Request for keys table of a scale or section.* ..... *35*
*5.7.8 <kbd2>::='7',55d,37h. Programming of direct keys tables of scale or section for a maximum of 255 keys per table.* ..... *35*
*5.7.9 <?kbd2>::='8',56d,38h. Request for tables of keys of a scale or section for tables of a maximum 255 keys* ..... *35*
*5.7.10 <cbr>::='9', 57d, 39h.Programming of bar codes in scale or section local* ..... *35*
*5.7.11 <?cbr>::=':'.58d,3ah. Request of bar codes in scales or section local* ..... *35*
5.8 #<BAL>::='5'.53D.35H PROGRAMMING OF BAR CODES AND KEYS (NO CONCENTRATOR) ..... 36
*5.8.1 <cbr>::='2', 50d, 32h.Programming of bar codes in scale or section networkworks* ..... *36*
*5.8.2 <?cbr>::='3'.51d,33h. Request for bar codes in the scales' networkwork.* ..... *36*
*5.8.3 <kbd2>::='7',55d,37h. Programming for direct keys tables of a scale or section for a maximum of 255 keys.* ..... *37*
*5.8.4 <?kbd2>::='8',56d,38h. Request for tables of keys of a scale of section for tables of maximum 255 keys.* ..... *37*
*5.8.5 <cbr>::='9', 57d, 39h.Programming of bar codes in scale or section networkwork total.* ..... *38*
*5.8.6 <?cbr>::=':'.58d,3Ah. Request for bar codes in scales local* ..... *38*
*5.8.7 <cbb>::=';', 59d, 3bh.Programming of bar codes in scales or section networkwork local.* ..... *39*
*5.8.8 <?cbb>::='<'.60d,3ch. Request for bar codes in the scales networkwork or local* ..... *40*
5.9 #<VND>::='6',54D,36H-VENDORS ..... 40
*5.9.1 <prg>::='0',48,30h.Creation/Modification of vendor. (Not 8031)* ..... *40*
*5.9.2 <bor>::='1',49d,31h.Clearing of vendor. (Not 8031)* ..... *40*
*5.9.3 <alt>::='2',50d,32h. Introducing a new vendor (Not 8031)* ..... *41*
*5.9.4 <baj>::='3',51d,33h.Temporary leave of vendor (Not 8031)* ..... *41*
*5.9.5 <cer>::='4',52d,34h.Closing of open totals of vendors.* ..... *41*
*5.9.6 <sts>::='5',53d,35h.Consulting vendor status (Not 8031)* ..... *41*
*5.9.7 <cvt>::='6',54d,36h.Vendor query by key (Not 8031)* ..... *42*
*5.9.8 ::='7',55d,37h.Consulting further data of vendor (Not 8031)* ..... *42*
5.10 #<ART>::='7',55D,37H-ITEMS ..... 43
*5.10.1 <prg>::='0',48d,30h. Basic programming of item.* ..... *43*
*5.10.2 <?prg>::='1',49d,31h.Request for information about item.* ..... *44*
*5.10.3 <bor>::='2',50d,32h.Erasing item.* ..... *44*
*5.10.4 <txt>::='3',51d,33h.Prog Item text* ..... *45*
*5.10.5 <?txt>::='4',52d,34h.Requesting item text* ..... *45*
*5.10.6 <prc>::='5',53d,35h. Change of item price (Not 8031).* ..... *46*
*5.10.7 Updating item stock (Not 386)* ..... *46*
*5.10.8 <?stk>::='7',55d,37h. Request item stock. (Not 386)* ..... *46*
*5.10.9 <env>::='8',56d,38h. Ear-mark item to send always to the scales. (Not 386)* ..... *46*
*5.10.10 <?env>::='9',57d,39h. Consulting marked item to be sent always to scale (Not 386)* ..... *47*
*5.10.11 <vst>::=':',58d,3ah. Consulting item stock data (Not 386)* ..... *47*
*5.10.12 <?vst>::=';',59d,3bh. Consulting item stock data. (Not 386)* ..... *47*
*5.10.13 <cts>::='<',60d,3ch. Programming item costs. (Not 386)* ..... *48*
*5.10.14 <?cts>::='=',61d,3dh. Consulting item costs (Not 386)* ..... *48*
*5.10.15 <?cv>::='>',62d,3eh. Clearing item sales. (Not 386)* ..... *48*
*5.10.16 <bve>::='?',63d,3fh. Clearing item sales. (Not 386)* ..... *49*
*5.10.17 <tve>::=' ',64d,40h. Consulting item sales total. (Not 386)* ..... *49*
*5.10.18 <?prgg>::='A',65d,41h.Requesting information about general item.* ..... *49*
*5.10.19 <?log>::='B',66d,42h.Consulting item logging line. (Not 386)* ..... *50*
*5.10.20 <iva>::='C',67d,43h. Programming item VAT.* ..... *50*

*5.10.21 <?iva>::=‘D’,68d,44h. Consulting item VAT. 51*
*5.10.22 <cpt>::=‘E’,69d,45h. Programming change of price for temporary offer 51*
*5.10.23 <cpt?>::=‘F’,70d,46h. Consulting change of price for temporary offer 51*
*5.10.24 <cpm>::=‘G’,71d,47h. Programming price change by margin (Not 386) 52*
*5.10.25 <cpm?>::=‘H’,72d,48h. Consulting change of price (Not 386) 52*
*5.10.26 <?pxc>::=‘I’,73d,49h. Consulting/Initializing the cost total for quantities bought of item . (Not 386) 52*
*5.10.27 <?ulog>::=‘J’,74d,4ah.Consulting user who has made the log (Not 386) 52*
*5.10.28 <pusu>::=‘K’,75d,4bh.Programming user of current job (Not 386) 53*
*5.10.29 <env>::='L',76d,4 Consulting item packaging data (Only 386) 53*
*5.10.30 <csd>::='L',76d,4Ch. Programming code as a Plu in a different section (no 386) 53*
*5.10.31 <env?>::='M',77d,4Dh. Consulting item packaging data (Only 386) 53*
*5.10.32 <ing>::='N',78d,4Eh. Programming item ingredients (Only 386) 54*
*5.10.33 <ing?>::='O',79d,4Fh. Consulting item ingredients (Only 386) 54*
*5.10.34 <mer?>::='P',80d,50h. Consulting goods entrance (Only 386) 54*
*5.10.35 <bmer>::='Q',81d,51h. Clearing all the goods inputs (Only 386) 55*
*5.10.36 <cba>::='R',82d,52h. Programming bar code by item (Only 386) 55*
*5.10.37 <cba?>::='S',83d,53h. Consulting bar code by item (Only 386) 55*
*5.10.38 <ped>::='T',84d,54h. Updating an item order. 55*
5.11 #<TOT>::=‘8’,56D,38H-TOTALS CONSULTING (NOT 386) 58
*5.11.1 <fch>::=‘0’,48d,30h.Query of time intervals in which there are totals. 58*
*5.11.2 <?fch>::=‘1’,49d,31h. Asing if there are totals in the indicated dates. 58*
*5.11.3 <bor>::=‘2’,50d,32h. Clearing and accumulating of great total. 58*
*5.11.4 <sec>::=‘3’,51d,33h.Total of equipment sales by sections, subsections and families. 59*
*5.11.5 <art>::=‘4’,52d,34h. Total in item sales in equipments 60*
*5.11.6 <vnd>::=‘5’,53d,35h. Total sales vendors. 60*
*5.11.7 <hor>::=‘6’,54d,36h. Total timetables in scales. 61*
*5.11.8 <bal>::=‘7’,55d,37h. General total of equipments. (Not 8031) 61*
*5.11.9 <dia>::=‘8’,56d,38h. General total of the concentrator (Not 8031). 62*
*5.11.10 ‘9’,57d,39h. Programming of type of totals (Not 8031). 62*
*5.11.11 <?ttot>::=‘:’,58d,3Ah. Consulting type of totals (Not 8031). 62*
*5.11.12 <tiva>::=‘;’,59d,3bh. VAT totals (Not 8031). 62*
*5.11.13 <tot?>::=‘<‘,60d,3ch. Totals saved in SC10 concentrator (Not 8031). 63*
*5.11.14 <hpc?>:=‘=‘,61d,3dh. Consulting parameters of automatic zero setting (Not 8031). 63*
*5.11.15 <hpc>:=‘>‘,62d,3eh. Programming parameters of automatic zero settings (Not 8031) 63*
5.12 # <PUB>::=‘9’,57D,39H-ADVERTISING. 63
*5.12.1 <pbc>::=‘0’,48d, 30h. Programming of continuous advertising of a scale or section 63*
*5.12.2 <pbo>::=‘1’,49d,31h.Programming of advertising offers of a scale. 64*
*5.12.3 <pbb>::=‘2’,50d,32h.Activation/Desactivation of the scale or section advertising 64*
*5.12.4 <?pbc>::=‘3’,51d,33h.Requesting continuous advertising of a scale or section. 64*
*5.12.5 <?pbo>::=‘4’,52d,34h.Requesting advertising offers of a scale or section. 64*
5.13 #<CBL>::=‘:’,58D,3AH. HEADINGS, LEGENDS AND FREE TEXTS. 65
*5.13.1 <cab>::=‘0’,48d,30h.Programming receipt texts of a scale or section 65*
*5.13.2 <?cab>::=‘1’,49d,31h.Requesting receipt texts of a scale or section. 65*
5.14 #<TOT>::='8',56D,38H- TOTALS (ONLY 386) 66
*5.14.1 <bor>::='2',50d,32h. Clearing of totals receipts (Zero setting) . 66*
*5.14.2 <sec>::='3',51d,33h.Total sales by section. 66*
*5.14.3 <art>::='4',52d,34h. Totals item sales 67*
*5.14.4 <vnd>::='5',53d,35h. Total vendors. 67*
*5.14.5 <hor>::='6',54d,36h. Total timetable. 68*
*5.14.6 <dia>::='8',56d,38h. Global total of the equipment’s system. 68*
*5.14.7 <tiva>::=';',59d,3bh. VAT total 68*
*5.14.8 <art>::='?',63d,3Fh. Total item range sales 69*
*5.14.9 <zart>::='@',64d,40h. Total different sales of items zero. 69*
5.15 # <TQT>::=‘;’,59D,3BH – RECEIPTS CONSULTING AND PARCELS SERIES 70
*5.15.1 <stm>::=‘0’,48d,30h.State of receipts memory (Not PC) (Not 386) 70*
*5.15.2 <cab>::=‘1’,49d,31h. Requesting receipts headings (Not 8031) (Not 386) 70*
*5.15.3 2’,50d,32h.Sequential request for receipt lines (Not PC) (Not 386) 71*
*5.15.4 <lin>::=‘3’,51d, 33h.Requesting receipt line (Not 8031) (Not 386) 71*
*5.15.5 <etq>::=‘4’,52d, 34h.Requesting label (Not 8031) (Not 386) 72*

*5.15.6 <lot>::='5',53d, 35h.Requesting of UBI labelling train series (Not 8031) (Not 386) ..... 72*
*5.15.7 <ca1>::='6',54d,36h. Requesting receipt heading per sec/ven/bal/fecha(Not 8031) (Not 386) 73*
*5.15.8 <li1>::='7',55d,37h. Requesting receipt lines as per sec/ven/bal/dat (Not 8031) (Not 386)74*
*5.15.9 <et1>::='8',56d,38h. Requesting label as per sec/ven/bal/dat (Not 8031) (Not 386)..... 74*
*5.15.10 <cag>::='9',57d,39h. Requesting heading of global receipt (Not 8031)..... 75*
*5.15.11 <lig>::=':',58d,3ah. Requesting global receipt lines (Not 8031) ..... 76*
*5.15.12 <cag>::=';',59d,3bh. Requesting for clearing global receipt (Not 8031) (Not 386)..... 77*
*5.15.13 <imt>::='<',60d,3ch. Requesting printing receipts (Not 8031) (Not 386) ..... 77*
*5.15.14 <nta>::='=',61d,3dh. Requesting alternative numbering (Not 8031) (Not 386) ..... 77*
*5.15.15 <ent>::='>',62d,3eh Requesting heading of input/output indicator (Not 8031) (Not 386) 78*
*5.15.16 <lent>::='?',63d,3fh. Requesting lines of input/output indicator (Not 8031) (Not 386)..... 78*
*5.15.17 <mar>::='?',63,3fh. Requesting receipt marking (Only 386) ..... 79*
5.16 #<TUR>::='<',60D,3CH-SPECIAL MESSAGES FOR TURN-NUMBERS.(NOT 8031) (NOT 386).....79
*5.16.1 <ean>::='0',48d, 30h.Requesting EAN codes of a customer code..... 79*
*5.16.2 <tqt>::='1',49d,31h.Request the receipt numbers of a customer. ..... 79*
*5.16.3 <dat>::='2',50d,32h. Consulting customer data ..... 80*
*5.16.4 <tqc>::='3',51d,33h. Requesting customer code number receipts ..... 80*
5.17 #<FMT>::='=',61D,3DH-LABEL AND RECEIPT FORMAT.....81
*5.17.1 <ett>::='0',48d, 30h.Label/receipt programming (Not 386)..... 81*
*5.17.2 <etq>::='1',49d,31h. Label programming (amplified) (Not 386) ..... 82*
*5.17.3 <bmp>::='0',48d, 30h. Logotype programming. (Only 386)..... 82*
5.18 #<ENV>: :='?',63D,3FH-PACKAGING DATA (ONLY 386).....83
*5.18.1 <ing>::='0',48d, 30h. Ingredients programming..... 83*
*5.18.2 <?ing>::='1',49d, 31h. Ingredients consulting..... 83*
*5.18.3 <bing>::='2',50d, 32h. Erasing ingredients..... 84*
*5.18.4 <inenv>::='3',51d, 33h. Initiating packaging (train)..... 84*
*5.18.5 <finenv>::='4',52d, 34h. Finalising packaging (train)..... 84*
5.19 #<TRN>::='>',62D,3EH-LABELLING TRAIN PROGRAMMING (NOT 386).....85
*5.19.1 <prg>::='0',48d,30h. Item train programming..... 85*
*5.19.2 <prg?>::='1',49d,31h.Sending to train control..... 85*
*5.19.3 <blot>::='2',50d,32h.Clearing series ..... 85*

**6. USE OF THE MESSAGES FOR THE CALCULATION OF TOTALS OF THE CONCENTRATOR (NOT 386) 86**

6.1 AMOUNT DUE, NUMBER OF OPERATIONS AND RECEIPTS IN TOTAL (NOT 386).....86
6.2 CALCULATION OF AMOUNT DUE AND NUMBER OF OPERATIONS (NOT 386).....87
6.3 QUERIES FOR THE CALCULATION OF THE NUMBER OF RECEIPTS .....88
(NOT 386).....88

**7. FORMAT SOCKETS COMUNICATION FORMAT..... 89**

7.1 PHYSICAL FORMAT SOCKETS.....89
7.2 WORK MODE WITH SOCKETS.....89
7.3 SOCKET MESSAGES FORMAT: .....90
7.4 REQUEST FOR DATA FROM THE SCALE UNITS.....90
7.5 SYSTEM PROGRAMMING: .....91
8.6 MODIFICATION OF MESSAGES FOR WORKING WITH SOCKETS:.....91

**8. BINARY PROTOCOL..... 91**

8.1 DESCRIPTION OF A BINARY PROTOCOL MESSAGE.....91
*8.1.1 Heading..... 91*
*8.1.2 Body ..... 92*
*8.1.3 End of message..... 92*
*8.1.4 CRC Field ..... 92*
*8.1.5 Routine of adjusting a message to the binary protocol ..... 92*

**9. ACCESS TO THE DATABASE ENGINE (IDF:000001)..... 93**

9.1 STRUCTURE OF AN ACCESS MESSAGE TO THE ENGINE.....93
*9.1.1 Function - Reply field..... 94*
*9.1.2 Method field ..... 94*
*9.1.3 Funmask field..... 95*
*9.1.4 Key field – main-access key..... 95*

*9.1.5 RKey field – extended-access or return key* ..... *95*
*9.1.6 Addition of keys* ..... *95*
*9.1.7 NDATA field* ..... *96*
*9.1.8 Data field* ..... *96*
9.2 EXAMPLES ..... 96
*9.2.1 Programming database elements:* ..... *96*
*9.2.2 Reply message to a programming operation:* ..... *102*
*9.2.3 Consulting database elements:* ..... *103*
9.3 DATABASES ..... 109
*9.3.1 Customers* ..... *110*
*9.3.2 Countries:* ..... *110*
*9.3.3 Price changes* ..... *110*
*9.3.4 Ingredients:* ..... *111*
*9.3.5 Coded tares:* ..... *111*
*9.3.6 Goods (Self-indexed)* ..... *112*
*9.3.7 VATs (Self-indexed):* ..... *112*
*9.3.8 Items + Totals* ..... *112*
*9.3.9 Descriptives (Chained):* ..... *118*
*9.3.10 Texts: headings, legends, free text and advertising (Self-indexed):* ..... *119*
*9.3.11 Bar code (Self-indexed):* ..... *119*
*9.3.12 Vendors + Totals:* ..... *119*
*9.3.13 Vendor keys:* ..... *122*
*9.3.14 Orders:* ..... *122*
*9.3.15 Bovine (self-indexed):* ..... *123*

**10. RECEIPT REQUEST ..... 124**

10.1 FORMAT REQUEST (IDF:000003) ..... 124
*10.1.1 Question format* ..... *124*
*10.1.2 Reply format* ..... *124*
10.2 RECEIPT REQUEST (IDF:000002) ..... 125
*10.2.1 Question format* ..... *125*
*10.2.2 Reply format* ..... *127*

**11. STATE OF THE SYSTEM (IDF:000004) ..... 128**

11.1 STATE OF THE SYSTEM REQUEST MESSAGE ..... 128
11.2 REPLY TO STATE OF THE SYSTEM REQUEST ..... 128

**12. LANGUAGE CONTROL MESSAGES (IDF:000005) ..... 128**

**13. APPENDICES ..... 129**

13.1 APPENDIX 1: BAR-CODE STRING ..... 129
13.2 APPENDIX 2: DATE-CONVERSION ROUTINES ..... 129

## 1. READER'S GUIDE

The notation used for the specification of this protocol is the following:

| | |
|---|---|
| <field> | Message field |
| <field (x)> | x=Definition of the field width in bytes. |
| <field>::=definition | Field definition |
| *XX*d | Number expressed in decimal base |
| *XX*h | Number expressed in hexadecimal base |
| 'x' | Character |
| Q:... | Request to the concentrator or scales system |
| A:... | Reply of the concentrator or scales system |
| [option] | Shows an optional field |
| \| (option 1 \| option 2) | Shows two or more possibilities |
| <field>...<field> | The field is repeated several times |

## 2. COMMUNICATION FORMAT

### Physical format

RS-232 channel serie
Configuration by default:
9600 Bauds, 8 data bits, 1 stop bit, parity even.
19200 Bauds (Euroscale).

### Message format:

<MESSAGE>::=<STX><IDG>[<IDM>]<DAT0>...<DATn><ETX><CHK><CR>

<STX>::=<BYTE> 2d,02h.
<IDG>::=<BYTE> Identifier of group of messages.
<IDM>::=<BYTE> Identifier of messages.
<DATx>::=<BYTE>...<BYTE><NUL> Message data of variable lenght. Ended by 0.
<NUL>::=<BYTE> 0.
<ETX>::=<BYTE> 3d,03h.
<CHK>::=<BYTE>Message check. Format by XOR of [<STX><IDG><IDM><DAT0>...<DATn><ETX>], XOR of the previous with lenght of the AND message of the result 3Fh y OR with 40h, so that the check is a number between 40h and F.
<CR>::=<BYTE> 13d ,0Dh.

The maximum lenght of a complete message is 255 bytes.

### Request of data to concentrator of scales system:

<REQUEST>::=<MESSAGE>

IDG and IDM define the request.
DATx are parameters associated to this data request.
The concentrator or scales' system replying in its turn with a message contatning the requested data headed by IDG or through a message with the following format.

A:<RSP(1)><ERR(2)> ( STX,RSP,ERR1,ERR2,ETX,CHK,CR )

<RSP>::=<ACK>|<NACK>|<BS>
<ACK>::=<2d> (2h)= Message headed correctly.
<NACK>::=<21d>(15h)=Communications error.
<BS>::=<8d>(08h)= Error in making the requested operation
<ERR>::= Has information which specifies the error produced (specific for message)

<ERR>::=<BYTE>,<BYTE>= Code of error or state.

‘00’ - No error.
‘01’ - Undefined error.
‘02’ - Uncontempled function.
‘03’ - Uncomprehensive data message .
‘04’ - Data non-found.
‘05’ - Database error.
'06' - Communications error (message between host and remote computer) (Only BNG)

‘10’ - Inexistant section (Only concentrator)
‘11’ - Inexistant subsection (Only concentrator)
‘12’ - Inexistant family (Only concentrator)
‘13’ - There is no superior level (Only concentrator)
(ex - There is no section of subsection)
‘14’ - Section, subsection or family marked as eliminated (Only concentrator)
‘15’ - Changing of section is not allowed (Only concentrator)
(ej - Changing the scales section is not allowed)
16’ - The section has programmed items (Only concentrator)
‘17’- There are inferior levels. (Only concentrador)
(ex.: The section has subsections)
‘18’- The section has scales assigned. (Only concentrator)

‘20’ - Wrong equipment number.
‘21’ - Wrong type of equipment.
‘22’ - Equipment connected.
‘23’ - Equipment without advertising.
‘24’ - Equipment without headings or legends.
‘25’ - Equipment disconnected.
‘26’ -Keypads are being programmed to an entire section . (Only concentrator)

‘30’ - Vendor marked as cleared (Only concentrator)
‘31’ - Vendor connected to the scales’ network.
‘32’ - Vendor not-connected to the scales’network.
‘33’ - The vendor has an open receipt.
‘34’ - The vendor doesn’t have an open receipt.
‘35’ – Vendor key already assigned to another.

‘40’ - Error in the form of sale (weight or units)
‘41’ - Form of sale cannot be changed (there are totals)
‘42’ - PLU code repeated (Only concentrator)
‘43’ - Price cannot be zero.
‘44’ - Item marked as cleared. (Only concentrator).

‘50’ - Type of descriptive incorrect.
‘51’ - Number of lines incorrect.
‘52’ - Type of letter incorrect.

‘60’ - Incorrect data format.
‘61’ - Incorrect time format.
‘62’ - Incorrect bar codes.

‘70’ - Database cannot be cleared.
‘71’ – Waiting time for confirmation finished.
‘72’ – Scales network connected.
‘73’ – Scales network disconnected.
‘74’ - There are vendors connected to the scale network.
‘75’ - Some scales without having sent the local total.
‘76’ - Date or time not recorded.
‘77’ - There is a request local total.
‘78’ - Concentrator is carrying out zero setting.

‘80’- Type of tycket numb ering non compatible.

'81'- Requested receipt not closed.
'82'- Receipt list is being made. (Only concentrator)

Errors of backup and data restore to unit local network unit. (Only concentrator)

'90' - No access route to the local network.
'91' - No data to recuperate in local network.
'92' - Writing error in local network unit.
'93' - Reading error in local network unit.
'94' – Request in process (386 only).

<ERR> is not sent if <RSP> is ACK.

When working in pounds, the weight is sent in ounze and the price per pound.

## Equipment programmings:

<PROGRAMMING>::=<MESSAGE>

IDG and IDM define the programming to be made. DATx are parameters associated to this programming.

The concentrator or scales system replies with the message:

A:<RSP(1)><ERR(2)> ( i.e. STX,RSP,ERR1,ERR2,ETX,CHK,CR )

<RSP>::=<ACK>|<NACK>|<BS>
<ACK>::=<2d>(02h)=Message received correctly.
<NACK>::=<21d>(15h)= Communications error.
<BS>::=<8d>(08h)=Error in the realization of the request.
<ERR>::=has got information that specifies the error made.
<ERR>::=<BYTE>,<BYTE>=Error or status code.
'00' - No error.
'01' - Undetermined error.
'02' - Uncontempled function.
'03' - Uncomprehensive message data.
'04' - Data non found.
'05' - Database error.
'xx' - Message specifics.

Other error codes are specific for the message and are detailed in each of them.
The concentrator or scales system can reply optionally with a message that specifies its new status. In this case, the reply message will come in the format specified by , <MESSAGE>.

## 3. LOCAL NETWORK (HOST) COMMUNICATION FORMAT

The notation used for the sepecification of this protocol is as follows:

| | |
|---|---|
| <field> | Field of a message |
| <field(x)> | x=Definition of the field width in bytes. |
| <field>::=definition | Definition of the field |
| *XX*d | Number expressed in decimal base |
| *XX*h | Number expressed in hexadecimal base |
| 'x' | Character |
| Q:... | Request to the concentrator or hng |
| A:... | Reply to the concentrator or hng |
| [option] | Indicates an optional field |
| \| (option 1 \| option 2) | Indicates 2 or more possibilities |
| <field>...<field> | The field is repeated several times |

The same information that can be requested to a scales system can also be requested to a computer with the scales programming (hng) connected to the system (through series channel or through scales network.)
The host requests this information to the scales and replies to the remote computer.

## Phisycal format

INDIVIDUAL LOCAL NETWORK boards connected through coaxial cables will be used. The concentrator programme or the hng computer must be configurated as <CUSTOMER> and will progressively acceed to a certain directory of a <SERVER> disc of the network. There will be two types of work files:

· File with commands which will be sent to the concentrator or to the host.
· File with replies from the concentrator or from the hng for the return of requested data or for report of the result of executing the previous requested command.

## Work mode

The files will have a serie of ASCII channels ended by CR, LF which will correspond to the different commands and to the results of these orders.

The work mode is the following:

1. The file will be created in the <SERVER> with all the commands that want to be given to the concentrator or to the hng.

2.The concentrator or hng, configurable every certain times (10 seconds by default), will check if any commands file has been created and if the results file exist. If the file makes orders, created but not open , and the results file doesn't exist; the concentrator or hng will open and will process order by order, in each bucle of the programme, all the messages of the orders's file. The same way the output file is openend and the results of the orders made are being written.These files will not be closed until they have been fully processed. Both the orders file and the results file will contain a number of order of 4 bytes to identify the replies with the orders.

3. Once all the orders file has been executed , the system will wait for the <SERVER> to clear, first the order file and forthwith the results file.Thereafter the initial waiting process will start again.

4. The receipts file will be automatically generated (depending on the hng configuration) containing information about headings and global lines. Receipts will be continually requested and will be stored on the receipts file with the same format as the messages <tqt><cag> and <tqt><lig> (not to be confused with the file for all receipts, see message <prf><tot>). This work mode will make it impossible to carry out the following operations on the scales:

- Reload receipt.
- Cancel receipt lines.

5.If the order has been requested, all the totals or receipts on the scale will beep and be stored in the corresponding file. (see <prf><tot>)

6.If there is a power failure, when the scale is turned on again it will check the state of the results file, comparing it with the order file. If there are orders still to be carried out, it will continue from the last order performed.

The default names of the local network files in the hng are:

- ORDER.DAT order file
- RESUL.DAT results file
- RECEIPT.DAT file of receipts continaully consulted at the scale
- TOTPLU.DAT item total file
- TOTTQT.DAT all receipts file
- TOTTQTC.DAT file of all receipts carried out by conditional consultation

## Messages format:

<MENSAJE>::=<IDO><IDG>[<IDM>]<DAT0>...<DATn><CR><LF>

<IDO>::=<BYTE><BYTE><BYTE><BYTE> Order number identifier.
<IDG>::=<BYTE> Messages group identifier.
<IDM>::=<BYTE> Message identifier.
<DATx>::=<BYTE>...<BYTE><NUL>Message data of variable lenght. Ended by < |>
<NUL>::=<BYTE> |, 124d, 7Ch.
<CR>::=<BYTE> 13d ,0Dh.
<LF>::=<BYTE> 10d ,0Ah.

The maximum lenght of a complete message is 255 bytes.

## Data request to concentrator or to equipment:

<REQUEST>::=<MESSAGE>
IDG and IDM define the request.
DATx are parameters associated to these data request.
The concentrator or system replies in turn with the message with the requested data headed by IDG or through a message with the following format:

A:<RSP(1)><ERR(2)> ( RSP,ERR1,ERR2,CR ,LF)

<RSP>::=<OK>|<KO>
<OK>::=<BYTE>,35d, 23h= Message treated correctly.
<KO>::=<BYTE>,36d, 24h= Error when making the request operation.
<ERR>::=Has information that specifies the error made.
<ERR>::=<BYTE>,<BYTE>= Error code or state code: see list in series communication section.
<ERR> is not sent if <RSP> is ACK.

## Concentrator or system programmings:

<PROGRAMMING>::=<MESSAGE>

IDG and IDM define the programming to be made, while DATx are parameters associated to this programming

The concentrator or the system replies at its turns with the message:

A:<RSP(1)><ERR(2)> ( i.e. RSP,ERR1,ERR2,CR,LF )

<RSP>::=<OK>|<KO>
<OK>::=<BYTE>,35d, 23h= Message treated correctly.
<KO>::=<BYTE>,36d, 24h= Error when making the requested.
<ERR>::=Contains information that specifies the error made.
<ERR>::=<BYTE>,<BYTE>=Error or status code: see list in serie communication section.
.

**COMMENTS :**
1.- The order number both in the order file and in the results' one, must be correlative.
2.- The error message has the BACKSPACE character (only for concentrator).

# 4. SPECIFIC MESSAGES FOR LOCAL/SOCKETS NETWORK

## 4.1 SPECIFIC MESSAGES FOR LOCAL/SOCKETS NETWORK

### 4.1.1 #<prf>::='/',47d,2Fh-FILE PROTOCOL

Group of messages relating to the communication protocol on the local/sockets network.

#### 4.1.1.1 <tmr>::= '0',48d,30h. Time between execution of orders

On the hng, the time between access to the order file and the scale is modified to request continuous receipts (not to be confused with the all-receipts file of the message <prf><tot>). This value is not stored on the configuration file.

In the case of sockets, it is also used to time waiting periods between receiving/sending.

Q:<prf><tmr><TMP(3)>
A:<RSP><ERR>

<TMP> ::= Time between line executions (from 000 to 999).

<RSP>::= Can be <OK> <KO>
<OK>::=<BYTE>,35d, 23h=Message treated correctly.
<KO>::=<BYTE>,36d, 24h=Error when carrying out requested operation.
<ERR> ::=

#### 4.1.1.2 <tot>::= '1',49d,31h. Order to request totals/receipts on file

Does not work for sockets.

Order to request all totals of data other than zero from the scale to be stored on the file.

Different totals can be requested at the same time. When there are no more totals/receipts, end the request until the new order. They will always be added to the end..

Line format can be widened by the end. The first version from which the datum will appear is written in italics between parenthesis.

Q:<prf><tot><PET(1)><TYP(1)>
A:<RSP><ERR>

<PET(1)>::= Type of totals to request
'0' - Item totals
'1'- All receipts
<TYP(1)::='A' – All types of totals/receipts
<ERR>::= Error code. May contain error number 94.

**Conditional consultation**: another question format: *(ver 004b, scale 9.000)*

Q:<prf><tot><PET(1)><TYP2(1)><PED(1)><LOC(1)><ENC(1)><STP(1)><CON(1)><CDO(1)><REC(1)><CAN(1)><COND(1)><DCOND(8)>
A:<RSP><ERR>

Values of data state (request conditions):
'0'- Do not take state into account
'1'- Not activated state
'2'- Activated state
E.g., local state datum <LOC>
'0'- Network and local
'1'- Network only
'2'- Local only

<PET(1)>::= Type of totals to request
'2'- All receipts with conditional consultation

<TYP(1)::=Type of receipts to request
'A' – All types of receipts
'0' – Scale receipts
'1'- Super mode scale labels
'2'- Packaging mode scale labels
'3'- Self-service mode scale labels
<PED>::= Mark as request
'0': do not mark
'1': mark as ordered
'2': mark as ordered and confirmed
<LOC>::= Datum state :receipts in local
<ENC>::=Datum state: receipts loaded
<STP>::=Datum state : receipts ordered
<STC>::= Datum state: receipts confirmed
<CER>::= Datum state : receipts closed
<REC>::= Datum state: receipts reloaded
<CAN>::= Datum state : receipts cancelled
<COND>::= Other consultation conditions *(ver 004d, scale 9.001)*
0: do not look at condition
1: same vendor
2: same counter
3: same date
4: same scale
5: same customer
<DCOND>::= Datum (item) of other consultation condition. *(ver 004d, scale 9.001)*

<ERR>::= Error code. Could contain error number 94.

**Item totals** (<PET>='0'):

Stored on the item totals file, the format can be widened.

*Item totals file:*

<COD(6)><NBAL(2)><TYP(1)><PES(9)><UND(9)><PTS(9)><OPS(8)>

<COD>::=Item code
<NBAL>::= Scale number if total is local
<TYP>::=Type of total
'0': Sales in network
'1': Sales in local
'5': Packaged in network
'6': Packaged in local
<OPS>::=Operations.
Scales: number of lines on sales receipt for this item.
Labelling train: Number of parcels of this item.
<PES> :: = Kilos sold
<UND>::= Units sold
<PTS>::= Total amount due sold.

**All receipts** (<PET>='1'):

Stored on the all-receipts file (TOTTQT.DAT by default).
A heading format with general data and another with lines are stored for each receipt. They can both be widened by the end.
They are not marked as orders.

*Heading format of receipt*

<TIP(1)><TQT(5)><SEC(2)><IDV(4)><BAL(2)><NLI(2)><NC(2)><PTS(7)><DTE(8)><HOR(8)><STS(1)><PED(1)><CLI(6)><DTO(2)><BTO(7)>

<TIP>::=Type of receipt:
'0': Scale in network receipt
'1': Super mode receipt for scale in network
'2': Packaging mode label for scale in network
'3': Local scale receipt
‘4’: Self-service mode label for scale in network
'5': Super mode label for local scale
'6': Packaging mode label for local scale
‘7’: Self-service mode label for local scale
<TQT>::=Receipt number.
<SEC>::=Section.
<IDV>::=Vendor identifier (0 on self-service scales)
<CLI>::Customer code, to configure customer turn
<BAL>::=Equipment at which scale was closed.
<NLI>::=Number of lines on receipt.
<NC>::=Number of lines cancelled.
<PTS>::=Amount due on receipt.
<HOR>::=Time receipt was closed, if available, otherwise blank
<STS>::=State of receipt.
'0':Closed, as has been finalised.
'1': Open inserting lines
'2':In total
'3': Not used
‘4’:Closed and loaded.
'5': Annulled
<PED>::=State of request of receipt.
'0'-Never requested.
'1'-Request, has been requested before (marked as requested).
<DTE>::=Date on receipt.
<DTO>::=Percentage of discount applied to receipt *(version 004b)*
<BTO>::=Gross amount due on receipt, without discount *(version 004b)*

*Line format of receipt*

<NLIN(2)><COD(6)><PRC(7)><PES(9)><STV(1)><STP(1)><STC(1)><PTS(7)><DTO(2)>

<NLIN>::=Line number
<COD>::= Item identifier
<PRC>::=Sale price.
<PES>::=Quantity sold.
<STV>::=Sale status.
'0':Add.
'1':Subtract.
<STP>::=Weight status.
'0':Weighted.
'1':Unweighted.
<STC>::=Cancellation status. If cancelled, there are neither additions nor subtractions in the total. Not used
'0':Not cancelled.
'1':Line cancelled, (neither added not subtracted in the total).
<PTS>::= Line amount due *(version 004b)*
<DTO>::=Percentage of discount applied to the line *(version 004b)*

**All conditional receipts** (<PET>=’2’):

Stored on the all-conditional receipts file (TOTTQTC.DAT by default).
A heading format with general data and another with lines are stored for each receipt. They can both be widened by the end.

Line and receipt formats are the same as <PET>=’1’.

4.1.1.3 <?tot>::= '2',50d,32h. State of request of totals/receipts on file order

Does not work for sockets.
Order to know the state of the request for totals.

Q:<prf><tot?><PET(1)>
A:<prf><CUR(1)>
A:<RSP><ERR>

<PET(1)>::= Type of totals whose state can be requested
'0' – Item totals.
'1'- All receipts
'2'- All conditional receipts
<CUR>::= State of the request
'0': Not being processed
'1': Being processed

4.1.1.4 <tqtc>::= '3',51d,33h. Activating continuous receipts request order

Does not work for sockets.
Order to activate or disactivate the request for continuous receipts in the host (not to be confused with the protocol order <prf><tot>).

Q:<prf><tqtc><ACT(1)>
A:<RSP><ERR>

<ACT>::= Activation order
'0': Disactivate
'1': Activate

4.1.1.5 <?tqtc>::= '4',52d,34h. State of order to activate continuous receipts request

Does not work for sockets.
Consult the state of the order to activate or disactivate the request for continuous receipts in the host (not to be confused with protocol order <prf><?tot>).

Q:<prf><?tqtc>
A:<prf><ACT(1)>
A:<RSP><ERR>

<ACT>::= Activation order
'0': Not activated
'1': Activated

# 5. MESSAGES

## 5.1 #<prt>::='0',48d,30h-PROTOCOL (Not 386)

Group of messages relative to the communications protocol, request of protocol and change of protocol.

### 5.1.1 <act>::='0',48d,30h.Request of present protocol .

The concentrator is asked which protocol is used in this sesion.

Q:<prt><act>
A:<prt><PRT(4)>
A:<RSP><ERR>

<PRT>::=Protocol code = pp.vv (pp=protocol,vv=version)

Any version accepts the messages of previous versions. Possible protocols are:

'S0' = Standard protocol.
'S1' = Host42 simago protocol.
'00' = 8031 stocks concentrator protocol.
'01' = 8031 tiquets I concentrator protocol.

#### 5.1.2 <sop>::='1',49d,31h.s Request for supported protocol (Not 8031)

The concentrator is asked if it accepts a specific protocol.

Q:<prt><sop><PRT(4)>
A:<prt><PRT(4)><ONF(1)>
A:<RSP><ERR>

<PRT> ::=Protocol code.
<ONF> ::=
'1' : Requested protocol supported .
'0' :.Requested protocol not supported

#### 5.1.3 <chg>::='2',50d,32h.Request for change of protocol.(Not 8031)

The concentrator is asked to use a new protocol once the actual transaction is ready (change of protocol).

Q:<prt><chg><PRT(4)>
A:<prt><PRT(4)>
A:<RSP><ERR>

<PRT> ::=Protocol code.

The next order will be done in the protocol replyed by the concentrator will change the protocol. The aplication should not take for granted that the concentrator will change the protocol, but rather stick to its reply.

### 5.2 #<exc>::='1',49d,31h-EXCLUSIVE (Not 8031) (Not 386)

Group of messages addressed to special equipments with their own communications protocol or with messages not included in the standard protocol.

#### 5.2.1 <grp>:='0',48d,30h.Request of exclusive code

Asks to a network equipment to return the code that identifies its own communication protocol.

Q:<exc><grp><BAL(2)>
A:<exc><BAL(2)><REF(3)>
A:<RSP><ERR>

<REF> ::=Communications reference of the equipment to which the message is addressed.
<ERR> ::=<BYTE>,<BYTE>=Error or status code.

#### 5.2.2 <qry>::='1',49d,31h.Exclusive message (addressed to a concrete network equipment).

A dialogue with concrete network equipment is requested through the concentrator. This dialogue will be made with the message protocol specific of the equipment.The message must be sent to a concrete mode network (BAL). Before doing this, it is necessary to know the configuration of network and the type of equipment connected to each mode. Without the <BAL> the ?? value is applied. The messages will be sent to all the equipments that reply to exclusive messages with the reference <REF>; in this case it is not guaranteed that all the equipments receive the message and that they work consequently with the orders given.

Q:<exc><qry><BAL(2)><ATT(1)><REF(3)><LEN(3)><DATA(?)>
A:<RSP><ERR>
A:<exc><BAL(2)><LEN(3)><DATA(?)> (Only inmediate reply)

<REF> ::=Communications reference of the equipment to which the message is addressed.
<BAL> ::=Identifier of the equipment in the network.
<ATT> ::=Awaits reply to the message.
'0'-Doesn't wait an reply.
'1'-Inmediate reply. The concentrator will reply with the data received by the equipment or with a timeout error.
'2'-Different reply . The specific reply of the equipment must be got through a posterior.
<LEN> ::=Lenght of the next message.
<DTA> ::=Specific data for an equipment.
<ERR> ::=Error code.

#### 5.2.3 <dta>::='2'.50d.32h Exclusive data (concrete equipment).

Request of the reply obtained from an equipment through a previous request . The concentrator keeps as maximum one request per equipment. Should be used to recollect requested information after an exclusive message.
Q:<exc><dta><NUM(2)>
A:<exc><NUM(2)><LEN(3)><DATA(?)>
A:<RSP><ERR>

Reply:
ACK 00: The information obtained is pending of sending to concrete equipment.
ACK 01: The information obtained is being processed in the equipment.
NACK ERR.
BS ERR: The equipment has replyed BS, ERR is the error code obtained..

### 5.3 #<sys>::='2',50d,32h-SYSTEM

Group of message relative to the system functioning and interval remote updating programme.

#### 5.3.1 <act>::='0',48d.30h.Programme updating (Not 8031) (Not 386)

It is requested to the concentrador is requested to update the programme or the databases through the serie channel. The concentrator will start the updating mecanism, and will wait for the data. If it has not been done in 3 seconds, the concentrator will be reset.
To make possible the updating, the two messages must be sent.
Q:<sys><act><'00'>
A:<RSP><ERR>
ACK '01' = Awaiting for the confirmation.
Q:<sys><act><'01'>
A:<RSP><ERR>
ACK '00' = Waiting for updating.

If the reply is ACK, the concentrator will remain waiting for the updating.
If it is <BS>, it means that the equipment doesn't bear the updating through the serie channel.

#### 5.3.2 <bor>::='1',49d,31h.Clearing internal databases .(Not 8031)

The concentrator or scales system is requested to clear its internal databases.

Q:<sys><bor><'00'><QUE(2)>
A:<RSP><ERR>
ACK '01' = Awaiting confirmation.
Q:<sys><bor><'01'><QUE(2)>
A:<RSP><ERR>
ACK '00' = Proceeds to the clearing.Maximum waiting time 5 seconds.

**For Concentrator:**

<QUE> ::=Indicates the items to be cleared:
‘00’:All.
‘01’:Sections
‘02’:Plus, descriptives.
‘03’:Scales vendors.
‘04’:Equipments.
‘05’:Advertising, headings and receipt’s legends of scale..
‘06’:Today timetable totals.
‘07’:Receipts and receipts lines of scale.
‘08’:Today’s total.
‘09’:Bar codes.
‘10’:Accumulated total of sections.
‘11’:Accumulaed totals of plus.
‘12’: Accumulated totals of vendors.
‘13’: Accumulated totals by items.
‘14’: Accumulated totals timetables.
‘15’: Accumulated totals per days.
‘16’:Items’ sales stadistics
‘17’:Price- quantity item’s sales.
‘18’:Items’ logging.

**For scales’ system:**

<QUE>::=Indicates the data to be cleared:
‘00’: All.
‘01’: Sections+totals
‘02’: Plus+totals, descriptive,ingredients,price change.
‘03’: Vendors + totals.
‘05’: Advertising,headings and legends of scale’s receipt, bitmap, free text.
‘07’: Receipts and scales’ receipt lines.
‘08’: System total.
’09’: Bar codes.
‘19’: Separated total by days of special family.
‘20’: tares
‘21’: customers
‘22’: ingredients
‘23’: countries
‘24’: keys
‘25’: Labels

#### 5.3.3 <kil>::=‘2’,50d,32h.Switching off the equipment.(Not 8031) (Not 386)

The concentrator is requested to switch off the electric network.

Q:<sys><kil>
A:<RSP><ERR>
<ERR> ::=Error code.

#### 5.3.4 <ref>::=‘3’,51d,33h.Programme reference request (Not 8031) (Not 386)

The concentrator programme reference is requested.

Q:<sys><ref>
A:<sys><ACT(1)><REF(N)>
A:<RSP><ERR>
<ACT> ::=
‘0’:The system doesn’t bear remote updating of programme.
‘1’:The system bears remote updating of programme.
<REF> ::=N alphanumercial digits ended by <NUL> that identify the programme.

**5.3.5 <?network>::='4',52d,34h.Request of network state. (Not 8031)**

The concentrator is asked if it's attending the equipment network or not.

Q:<sys><?network>
A:<sys><ONF(1)>
A:<RSP><ERR>

<ONF> ::=
'1'=Connected network.
'0'=Disconnected network.

Request to the scales if they are allowed to sell.

Q:<sys><?network>
A:<sys><ONF(1)>
A:<RSP><ERR>

<ONF>::=
'0'=Connected network.
'1'=Disconnected network.
<network>::='5',53d,35h.Connection/Disconnection. (Not 8031)

**For Concentrator:**

To attend the connected equipments or stop doing so:
- if a connection is sent and it was already connected , it continues connected
- if a disconnection is sent and it is disconnected, it continues disconnected.

Q:<sys><network><ONF(1)>
A:<RSP><ERR>
<ONF> ::=
'1'=Connecting the network.
'0'=Disconnecting the network.
<ERR> ::= Error code.

**For scales system:**

The equipments turn to the state in which they can make sales or to a state in which can not. The possible open receipts are cleared. If they are already in the requested state, they continue as they are.
Q:<sys><network><ONF(1)>
A:<RSP><ERR>
<ONF>::=
'0=Connect the network.
'1=Disconnetwork the network.
<ERR>::=There is no possibility of error.

**5.3.6 <fmt>:='6',54d,36h Programming date and time format.(Not 8031) (Not 386)**

Programme the date and time format in the messages. From the reception of this message on, any date and time are understood with the indicated format.

Q:<sys><fmt><FMD(1)><FMH(1)>
A:<RPS><ERR>

<FMD> ::=Date format
'0'-'DDMMYY' (010193) (Value by default)
'1'-'DD/MM/YY ' (01/01/93)
'2'-'MM/DD/YY' (01/01/93)
'3'-'MMDD/YY' (010193)
'4'-'YY/MM/DD' (93/01/01)
<FMH> ::=Time format

'0'-'HHMM' (1500) (Value by default)
'1'-'HH:MM' (15:00)
'2'-'HH:MM:SS' (15:00:00)
'3'-'HHMMSS' (150000)

**5.3.7 <?fmt>::='7',55d,37h. Request of date and time format.**

Request the date and time format used.

Q:<sys><?fmt>
A:<sys><FMD(1)><FMH(1)>

In scales' system, the reply will be the one of the first scale that replies to the request. This is the same format for all the equipments of the system.

The format is fixed :
Date format: '0'
Time format : '0'

**5.3.8 <dte>::='8',56d,38h. Programming date and time**

Actualize the date and time of the concentrator.
The reception of date and time in some equipments is not inmediate, because they are the ones that request the updating.

Q:<sys><dte><TME(8)><DTE(8)>
A:<RPS><ERR>

<DTE> ::=Date in the format specified by the message <fmt>. If no format is specified, the concentrator or system works with the last format sent.
<TME> ::=Time specified by the message <fmt>. If no format is specified, the concentrator or system works with the last format sent.
<ERR> ::= Error code.

**5.3.9 <?dte>::='9',57d,39h.Request for date and time**

Request the date and time format of the system.

Q:<sys><?dte>
A:<sys><TME(8)><DTE(8)>

<DTE> ::=Date in the format specified by the message <fmt>. In the case of concentrator, if no format is specified, it will work with the last format sent.
<TME> ::=Time in the format specified in the message <fmt>.In the case of concentrator, if no format is specified, it will work with the last format sent

In scales system, the reply will be the one of the scale that replies to the request. All the equipments of the system are syncronized.

**5.3.10 <cpf>::=':'58d,3ah.Programming country code and manufacturer (Not 386)**

Programming country code and manufacturer. These two codes are used in scales of type labeller in order to obtain the bars code if its printing is programmed (see message scale bar code programming).

Q:<sys><cpf><PAI(2)><FAB(5)>
A:<RPS><ERR>

<PAI> ::=Country code.
<FAB> ::=Manufacturer code.

**5.3.11 <?cpf>::=';',59d,3bh.Request for country code and manufacturer (Not 386)**

Consult the codes currently programmed.

Q:<sys><?cpf>
A:<sys><PAI(2)><FAB(5)>
A:<RPS><ERR>

<PAI> ::=Country code.
<FAB> ::=Manufacturer code.

**5.3.12 <?conf>::='<',60d,3ch. Request for configuration**

**For concentrator:**

Consult the configuration data useful for the concentrator.

Q:<sys><?conf>
A:<sys><DEC(1)><TIC(1)><ASP(1)><SUT(1)><TRD(1)><DIG(1)><TPE(1)><PVR(1)>
<VND(1)><EUR(10)><CIVA(1)>
A:<RPS><ERR>
<DEC> ::=Number of decimals : '0', '1', '2'
<TIC> ::=Numbering of receipt
'0': Consecutive
'1': By section
'2': By vendor
'3': By scale
'4': By date
<ASP> ::=Assignation of automatic PLU.
<SUT> ::=Type of turn-number
'0': Doesn't have
'1':Introduce turn-number customer when finishing the receipt.
'2': Introduce turn-number customer when initiatig the receipt
<TRD> ::=Type of rounding
'0' Truncate . '1' : Rounded to unit
'5': Rounded to zero-five.
<DIG> ::=Nr. of digits of PLU.
'3': three digits.
'4': four digits.
<TPE> ::=Type of weight.
'0': Kilos.
'1': Pounds.
<PVR> ::= Keep the real sales prices for the benefits calculation.
'0' : No.
'1' : Yes..
<CIVA> ::= Calculating VATs.
'0' : Don't calculate.
'1' : Calculating VATS by item.
'2' : Calculating VATS by section.
<VND> ::= Nr. of vendors..
'0': 12
'1': 24
<EUR> ::= Value of EURO.
'0000000001' until '4294967295'

**For scales' system:**

Consulting the configurating data of the equipments.All the equipments will have the same configuration.

Q:<sys><?conf>
A:<sys><DEC(1)><TIC(1)><ASP(1)><SUT(1)><TRD(1)><DIG(1)><TPE(1)><PVR(1)>
<CIVA(1)>

A:<RPS><ERR>

<DEC>::= Number of decimals
'0', '1', '2'

<TIC>::=Receipt's numbering.
'0': Consecutive
'1': By section
'2': By vendor
'3': By scale.
'4': By date.

<ASP>::= '0' Not used.

<SUT>::=Turn-number type.
'0': Doesnt' have.
'1':Introduce nr. of customer *turn-number* when finishing the receipt.
'2':Introduce nr. of customer *turn-number* when iniciating the receipt

<TRD>::= Type of rounding
'0' : Truncate.
'1' : Rounded to unit.
'5': Rounded to zero-five.

<DIG>::=Nr. of digits of PLU
'3': three digits
'4': four digits

<TPE>::=Type of weight
'0': Kilos
'1': Pounds

<PVR>::= '0' Not used.

<CIVA>::='0' Not used.

**5.3.13 <piva>::='=',61d,3dh. Programming types of VAT.**

Programming the VATS number and the values of the VATS.

Q:<sys><?piva><NIV(1)><VAT1(4)><VAT2(4)><VAT3(4)><VAT4(4)><VAT5(4)>
A:<RPS><ERR>

<NIV> ::= Nr. of VATS values, mximum five.The rest sent to zero.
<VATi> ::= VAT percentage expressed with two entire digits and two decimals. I.e : VAT = 5%, is expressed <5000>.

**5.3.14 <?civa>::='>',62d,3eh. Consulting the VAT types.**

Consulting the VAT values of the programmed VATS.

Q:<sys><?civa>
A:<sys><NIV(1)><VAT1(4)><VAT2(4)><VAT3(4)><VAT4(4)><VAT5(4)>
A:<RPS><ERR>

<NIV> ::= Nr. of VAT values, maximum 5.The rest are sent to zero.
<VATi> ::= VAT percentage expressed with two entire digits and two decimals. I.e : VAT = 5%, is expressed <5000>.

**5.3.15 <bck>::='?',63d,3fh. Backup of data files to LOCAL NETWORK WORK (Not 386)**

Q:<sys><bck>
A:<sys><FSEC(1)><FOPE(1)><FPLU(1)><FDPLU(1)><FEST(1)><FTIC(1)><FLIN(1)>
<FTUR(1)><FLOT(1)><FEQU(1)><FBAL(1)><FTEC(1)><FPUB(1)><FHOR(1)>
<FBAR(1)><FDIA(1)><FTSE(1)><FTPL(1)><FTOP(1)><FTHO(1)><FTBA(1)>
<FTDI(1)><FCV(1)><FLOG(1)><FPAR(1)><FPARH(1)><FUSUH(1)><NADA(8)>
A:<RPS><ERR>

Each field has one of the following numerical values according to the result of the copy:

0 : Correct.
90: There is no directory where the files can be copied.
91: Origin file not found.
92: Error when writing the destination file.
93: Error when writing the origin file

Names of each data file :

| | |
|---|---|
| <FSEC(1)> | FSECTION |
| <FOPE(1)> | FOPER |
| <FPLU(1)> | FPLU |
| <FDPLU(1)> | FDESCRIP |
| <FST(1)> | FSTATE |
| <FTIC(1)> | FTICKBAL |
| <FLIN(1)> | FLINBAL |
| <FTUR(1)> | FTURN |
| <FLOT(1)> | FLOT |
| <FEQU(1)> | FEQUIPMENTS |
| <FBAL(1)> | FSCALE |
| <FTEC(1)> | FKEYS |
| <FPUB(1)> | FPUBLICI |
| <FTIM(1)> | FHORABAL |
| <FBAR(1)> | FBARS |
| <FDAY(1)> | FDIARY |
| <FTSE(1)> | FTOTSEC |
| <FTPL(1)> | FTOTPLU |
| <FTOP(1)> | FTOTOPER |
| <FTHO(1)> | FTOTTIME |
| <FTBA(1)> | FTOTSCAL |
| <FTDI(1)> | FTOTDAY |
| <FCV(1)> | FCV |
| <FLOG(1)> | FLOGPLU |
| <FPAR(1)> | PARAM.SYS |
| <FPARH(1)> | PARAMH.SYS. Only for concentrator with screens. |
| <FUSUH(1)> | USER.SYS. Only for concentrator with screens. |
| <NOTHING(8)> | |

#### 5.3.16 <res>::=' ',64d,40h. Recuperating data files of LOCAL NETWORK unit (Not 386)

Q:<sys><res>
A:<sys><FSEC(1)><FOPE(1)><FPLU(1)><FDPLU(1)><FEST(1)><FTIC(1)><FLIN(1)>
<FTUR(1)><FLOT(1)><FEQU(1)><FBAL(1)><FTEC(1)><FPUB(1)><FHOR(1)>
<FBAR(1)><FDAY(1)><FTSE(1)><FTPL(1)><FTOP(1)><FTHO(1)><FTBA(1)>
<FTDI(1)><FCV(1)><FLOG(1)><FPAR(1)> <FPARH(1)><FUSUH(1)><NOTHING(10)>
A:<RPS><ERR>

#### 5.3.17 <sby>::='A',65d,41h. Modifying stanby (Not Concentrador)

Modifies the 'standby' state of the equipment (switched off without switch).

Q:<sys><sby><MOD(1)><DIR(2)><SBY(1)>
A:<RPS><ERR>

<MOD>::=Indicates if the programming is for scale or for section.
'0':Section.
'1':Scale.
'2':System
<DIR>::=Nr of scale ro section/counter..
<SBY>::=Standby state.
'0': Desactivating standby
'1': Activating standby

**5.3.18 <?sby>::='B',66d,42h. Consulting standby (Not Concentrador)**

Consults the stanby state of the equipment (switched off without switch).

Q:<sys><sby><MOD(1)><DIR(2)>
A:<sys><MOD(1)><DIR(2)><SBY(1)>
<MOD>::=Indicates if the consulting is for scale or for section.
'0':Section.
'1':Scale.
'2':System
<DIR>::=Nr.of scale or section/counter.
<SBY>::=Standby state
‘0’: is not in standby
‘1’: is in standby

**5.3.19 <vat>::='C',67d,43h. Programming scales VAT types (Not Concentrator)**

Programming the scales VAT types.

Q:<sys><vat><MOD(1)><DIR(2)><VAT(1)>
A:<RPS><ERR>
<MOD>::= Always ‘2’
<DIR>::=Not used.
<IVA>::= Type of VAT of the scales (0..4). Must be programmed.

**5.3.20 <?vat>::='D',68d,44h. Consulting the types of VAT of the scales (Not Concentrator)**

Consults the types of VAT of the scales.

Q:<sys><?vat><MOD(1)><DIR(2)>
A:<sys><MOD(1)><DIR(2)><vat(1)>
A:<RPS><ERR>
<MOD>::= Always ‘2’
<DIR>::=Not used.
<IVA>::= Type of VAT of the scales (0.4)

**5.3.21 <prg>::='E',69d,45h. Change programming host activation to all (Not Concentrator)**

Decide if the programming of the item data from the host is only done to the scale connected or to all of them.

Out of use. Programming is always for every one.

Q:<sys><prg><ACT(1)>
A:<RPS><ERR>
<ACT>: Change of activation.
‘0’: Only programming in the scale connected to the host.
‘1’: Programming all the scales.

**5.3.22 <?prg>::='F',70d,46h. Consulting programming host activation to all (Not Concentrator)**

Q:<sys><?prg>
A:<sys><ACT(1)>
A:<RPS><ERR>
<ACT>: Change of activation.
‘0’: Only programming in the scale connected to the host.
‘1’: Programming to all the scales.

Always responds ‘1’.

#### 5.3.23 <borg>::='H',72d,47h.Clearing internal databases by groups.

Ask the scale to clear the internal databases. (From scale version B000 on).

Q:<sys><borg><'00'><MOD(1)><DIR(2)><QUE(2)>
A:<RSP><ERR>
ACK '01' = Waiting for confirmation
Q:<sys><bor><'01'><MOD(1)><DIR(2)><QUE(2)>
A:<RSP><ERR>
ACK '00' =Proceeds to clearing. Maximum waiting time: 5 seconds.
<MOD>::=Indicates whether clearing is for scale, section or system.
'0':Section
'1':Scale
'2':System
<DIR>::=Number of scale or section/counter.

<QUE>::=Indicates data to clear:
'00': * All data.
'01': Sections+totals.
'02': * PLUs+totals, descriptives, ingredients, price change.
'03': * Vendors + totals.
'05': Advertising, headings and legends of the receipt on the scale, bitmaps, free texts.
'07': Receipts and receipt lines of the scale.
'08': System total + VAT totals + form-of-payment totals.
'09': Bar codes.
'19':Total separated by special family days.
'20': Tares.
'21': * Customers.
'22': * Ingredients.
'23': * Countries.
'24': Keys.
'25': Labels.
'26': * VATs.
'27': * Bovine.
'28': * Orders.
'29': * Goods.

NOTE: Data marked '*' will only be accepted under a system clearing.

### 5.4 #<sec>::='3',51d,33h-SECTIONS (FOR CONCENTRATOR)

Message relative to the configuration of sections.

#### 5.4.1 prg>:='0',48d,30h.Create/Modify section,subsection,family (Not 8031)

Programming section, subsections, and families. They are the first data of the database of the concentrator to be programmed. Since the items and scales have to to be assigned to a section.
The programming of sections, subsections and families must be done in the following order: firstly the sections, after that the subsections and then the families. The sections, subsections and families are used to group (together) items and to obtain totals of these items.

To create or modify a section, indicate its number in the field <SEC> , and <SUB> is initiated to <'000'> and <FAM> is initiated to <'0000'>.
To create of modify a subsection: indicate in <SEC> the section to which the subsection belongs, in <SUB> the number of subsection to be programmed and in <FAM> the value <'0000'>.
To create or modify a family, indicate the 3 values.

Q:<sec><prg><SEC(2)><SUB(3)><FAM(4)><TXT(21)><COD(6)>
A:<RPS><ERR>

<SEC> ::=Section
<SUB> ::=Subsection.
<FAM> ::=Family.
<TXT> ::=Text identifier of the section, subsection, family.
<COD> ::= Section code. It is only necessary for the section level.
<ERR> ::=Error code.

**5.4.2 <bor>:='1',49d,31h.Clearing section, subsection, family (Not 8031)**

Clearing of sections, subsections and families.

To clear a section, indicate its number in the field <SEC>, then <SUB> is initiated to <000> and <FAM> to <0000>.

To clear a subsection: indicate in <SEC> the section to which the subsection belongs, in <SUB> the number of subsection to be cleared and in <FAM> the value <'0000'>.

To clear a family, indicate the three values.

The clearing must be done step by step: firstly the families, secondly the subsections and finally the sections.

When a section has been cleared, the items and scales assigned become unaccessible.

The operation is always done as shown in the message.

Q:<sec><bor><SEC(2)><SUB(3)><FAM(4)><STS(1)>
A:<RPS><ERR>

<SEC> ::=Section
<SUB> ::=Subsection.
<FAM> ::=Family.
<STS> ::=If, when clearing there are data accumulated in the accumulated total (daily) can be marked as cleared, the physical clearing will be made when clearing the great absolute total of the concentrator.
'0':Cleared physically.
'1':Only marked as cleared.
'2':Unmarked of cleared.
<ERR> ::=Error code.

**5.4.3 <?sec>::='2',50d,32h.Consulting information section, subsection, family (Not 8031)**

The data programmed for a section, subsection or family are consulted.

To consult a section, indicate its number in the field <SEC>, and <SUB> is initiated to <'000'> and <FAM> is initiated to <'0000'>.

To consult a subsection, indicate in <SEC> the section to which the subsection belongs, in <SUB> the number of subsection to be consulted, in <FAM> the value is <'0000'>.

To consult a family, indicate the three values..

Q:<sec><?sec><QUE(1)><SEC(2)><SUB(3)><FAM(4)>
A:<sec><SEC(2)><SUB(3)><FAM(4)><TXT(21)><COD(6)><ELI(1)>
A:<RPS><ERR>

<QUE> ::=Indicates the data to be searched.
'0':Search equal.
'1':Search equal or more.
'2':Search more.
'3':Search equal or less.
'4':Search less.
<SEC> ::=Section.
<SUB> ::=Subsection.
<FAM> ::=Family.
<TXT> ::=Identificative text.
<COD> ::=Section code, only if the section is consulted.
<ELI> ::=State :
'0':Active.
'1':Cleared.
<ERR> ::=Error code.

**5.4.4 <pns>::='3',51d,33h. Programming number or sections (not Pc)**

Programming the number of sections in a 8031 concentrator.

Q:<sec><pns><NUM(1)>
A:<RPS><ERR>

<NUM> ::= Number of sections to be programmed.
'0': 6 sections
'1': 4 sections
'2': 8 sections
<ERR> ::=Error code.

**5.4.5 <?exi>::='4',52d,34h. Consulting if there is a section (not Pc)**

Ask the concentrator if there is a section.

Q:<sec><pns><SEC(2)>
A:<RPS><ERR>

<SEC> ::= Section to be consulted.
<ERR> ::=Error code.

**5.4.6 <pvs>::='5',52d,34h. Programming VAT of section**

Programming the type of VAT of the section.

Q:<sec><pvs><SEC(2)><IVA(1)>
A:<RPS><ERR>

<SEC> ::= Number of section to be programmed.
<IVA> ::= VAT type to be applied, index to the VAT board programmed previously.
Values from '1' up to maximum 5.
<ERR> ::=Error code.

**5.4.7 <?pvs>::='6',53d,35h. Consulting VAT or the sections**

Consulting the type of VAT of the section.

Q:<sec><?pvs><SEC(2)>
A:<sec><SEC(2)><VAT(1)>
A:<RPS><ERR>

<SEC> ::= Number of section to be consulted.
<VAT> ::= Type of VAT to be applied, index to the type of VAT previously programmed.
Values from '1' up to maximum 5.

<ERR> ::=Error code.

**5.5 #<sec>::='3',51d,33h-SECTIONS (COUNTERS). (FOR A SYSTEM OF SCALES)**

Messages relative to the configuration of sections or counters.
REMARK:The family programmed in the items is an independent group of the sections.

**5.5.1 <prg>:='0',48d,30h.Create/Modify sections**

Sections programming.

Q:<sec><prg><SEC(2)><SUB(3)><FAM(4)><TXT(21)><COD(6)>
A:<RPS><ERR>

<SEC>::=Section
<SUB>::=Subsection.Must be ‘000’, not used.
<FAM>::=Family.Must be ‘0000’, not used.
<TXT>::=Identifier text of the section.
<COD>::=Section code. Must be ‘000000’, not used.
<ERR>::=Error code.

The equipments assigned to this counter are indicated which is the identifying name of the counter.

#### 5.5.2 <?sec>::='2',50d,32h.Consulting section or counter

Q:<sec><?sec><QUE(1)><SEC(2)><SUB(3)><FAM(4)>
A:<sec><SEC(2)><SUB(3)><FAM(4)><TXT(21)><COD(6)><ELI(1)>
A:<RPS><ERR>

QUE> ::=Indicates the data to be searched.
‘0’:Search equal.
‘1’:Search equal or more.
‘2’:Search more.
‘3’:Search equal or less.
‘4’:Search less.
<SEC> ::=Section.
<SUB> ::=Subsection.It is ‘000’, not used.
<FAM> ::=Family.It is ‘0000’, not used.
<TXT> ::=Identificative text.
<COD> ::=Section code, it is ’000000’, not used.
<ERR>::=Error code.

Reply to the identificative which has been programmed for the section or counter.

### 5.6 #<eqp>::=‘4’,52d,34h-EQUIPMENTS

**For concentrator:**

Messages relative to the concrete configuration of an equipment in a network. The maximum number of supported equipments is 30. The numberation of the equipments is from 1 to 30.

**For scale system:**

Message relative to the concrete configuration of an item in a network. The maximum number of equipments supported is 60. The numbering of the equipments is from 1 to 60. Before any programming, all the equipments must be in the network with different equipment number. It is necessary to designate to each equipment the counter in which it will be working.

#### 5.6.1 <prg>:=‘0’,48d,30h.Creation,Modification of equipment for Concentrator

SCALE:
Q:<eqp><prg><TYP(1)><NUM(2)><ONF(1)<PRI(1)><STP(2)><SEC(2)>
A:<RPS><ERR>
<TYP> ::=Type of equipment that is being programmed. The rest of the message will depend on the equipment.
Possible values :
‘0’:Scale.
‘1’:Labelling train UBI
‘2’:Labelling train CL
<PRI> ::=Priority (‘1’ to’9’)(Not 8031)
<NUM> ::=Equipment number.(Not 8031)
<ONF> ::=Status of equipment.(Not 8031)
‘1’:Connected
‘0’:Disconnected.
‘2’:Connected but not responding.

<STP> ::=Programming of advertising, headings and legends (Not 8031 nor labelling train).
<SPT[1]>:
‘0’:Doesn’t have advertising (numerical scales).
‘1’:Has advertising (alphanumerical scales).
<STP[2]>:
‘0’:Doesn’t have headings and legends (Self Service scales).
‘1’: Contains headings and legends (receipt scales).
<SEC> ::=Section, determines the PLUS that can be sold in the scale (Not labelling train)
<ERR> ::=Error code.

#### 5.6.2 <prg>:='0',48d,30h.Creation , Modification of Equipment for System Scales

SCALE:
Q:<eqp><prg><TYP(1)><NUM(2)><ONF(1)<PRI(1)><STP(2)><SEC(2)>
A:<RPS><ERR>

<TYP>::='0':Scale.
‘1’: Labelling train
<PRI>::= ‘0’ Not used
<NUM>::=Equipment number.
<ONF>::=State of equipment.
'0':Connected, can make sales.
'1':Disconnected, cannot make sales.
<STP[1]>: Not used.
<STP[2]>: Not used.
<SEC>::= Counter. The PLUS assigned to the section (in case of access for PLU) will be sold.
The vendors share the equipments of a section.
<ERR>::=Error code.

#### 5.6.3 <eli>::=‘1’,49d.31h.Eliminating an equipment (Not 8031)

Q:<eqp><eli><NUM(2)>
A:<RPS><ERR>

<NUM> ::=Equipment number
<ERR> ::=Error code.

**Concentrator:** To eliminate a network equipment, either disconnect it first or disconnect the network. Bare in mind that the eliminated equipment may have accumulated totals. If another equipment is connected with the same number, the totals of two will be added.

**Scales’ system:** The designation of the section made with the ones that cannot be sold is cancelled. In the case of working with access to PLUS, it is necessary to send a clearing of PLUS, to enable the new designation of the equipment to another section (always when the user requests it).

#### 5.6.4 <network>::=‘2’,50d,32h.Connection/Disconnection of an equipment (Not 8031)

The equipment is programmed previously.
Q:<eqp><network><NUM(2)><ONF(1)>
A:<RPS><ERR>

<NUM> ::=Number of equipment.
<ONF> ::=
‘0’-Connect equipment.
‘1’-Disconnect the equipment
<ERR> ::=Error code.

**5.6.5 <pri>::='3',51d,33h.Change of priority of an equipment .(Not 8031) (Not 386)**

The concentrator interrogates the programmed and connected equipments. This is the moment when, if the equipment has something pending, it can send it to the concentrator . The interrogation is sequential from the first to the last one and so on.

Depending on the priority given to an equipment, the interrogation can change. If one equipment has priority 2, the concentrator interrogates once every 2 rounds for all the equìpments, if the priority is 3, once every 3 rounds and so on.

The maximum priority is 1, and the minimum is 9.

Q:<eqp><pri><NUM(2)><PRI(2)>
A:<RPS><ERR>

<NUM> ::=Equipment number.
<PRI> ::=Priority of attention to the equipment (1 to 9).
<ERR> ::=Error code.

**5.6.6 <?sts>::='4',52d, 34h. Request for the status of an equipment**

Consulting the data programmed for one equipment. The reply will depend on the type of equipment.

SCALES:
Q:<eqp><?sts><NUM(2)>
A:<eqp><NUM(2)><TYP(1)><STS(1)><ONF(1)><PRI(1)><REF(9)><SEC(2)><STP(2)>
<TYP> ::=Type of equipment.
'0' : Scale.
'1' : Labelling train UBI or 386
'2' : Labelling train CL
<NUM> ::=Equipment number.
<STS> ::=Type of scale.
' ':Don't know the type.
'0':V12 numerical with star printer.
'1':V12 numerical with enertec printer.
'2':Labeller in sale.
'3:Self service of 40 keys.
'4':Self service of 80 keys.
'5':Labeller in packing.
'9':V12 alphanumerical with star printer.
':':V12 alphanumerical with enertec printer.
For 386: '1': Fixed
<ONF> ::=State of equipment.(Not 8031)
'1':Connected.
'0':Disconnected.
For 386:
'1':Disconnected.
'0':Connected.
<PRI> ::=Priority (1 to 9).(Not 8031) (Not 386)
<SEC> ::=Section.
<REF> ::=Reference of the programme installed in the equipment.
<STP> ::=Programming of advertising, headings and legends.(Not 8031) (Not 386)
<SPT[1]>:
'0':Doesn't have advertising (numerical scales).
'1':Has advertising (alphanumercial scales).
<STP[2]>:
'0':Doesn't have headings and legends (Self Service scales).
'1':Has headings and legends (receipt scales).

**5.6.7 <?num>::='5',53d, 35h. Request for programmed number of equipment (Not 8031)**

Q:<eqp><?num>.
A:<eqp><LNS(2)><NUM(2)>..<NUM(2)>.
A:<RPS><ERR>.
<LNS> ::=Number of equipments programmed.
<NUM> ::= Numbers assigned to the equipments programmed.
There are always 60 but to zero from the equipment number LNS up to 60 (Only 386) .

**5.6.8 <act>::='6',54d, 36h.Order of programming from one scale to others (Only 386)**

An order is sent to a scale to programme the items data to the rest.
Out of use. Programming is always for every one.

Q:<eqp><act><NUM(2)><TOT(1)>
A:<RPS><ERR>
<NUM>: scale number that must programme to the others. If it is 0, it will be the one communicated to the host.
<TOT>: programme all the changes.
'0' : changes (non implemented)
'1': All

**5.6.9 <?act>::='7',55d, 37h.Query of programming from one scale to others (Only 386)**

Q:<eqp><prg><NUM(2)>
A:<eqp><ACT(1)>
A:<RPS><ERR>

<NUM>: scale number to be consulted. If it is 0, it will be the one communicated with the host.
<ACT>: says if it's programming.
'0': it's not programming
'1': programming in progress

Always responds '1'.

**5.6.10 <?ver>='8,56d, 38h. Consulting version and configuration data**

The reply can be widened at the end, for this reason it can be consulted by size. According to the size, it may be that the last field is not complete and without the null.

Q:<eqp><?ver><NUM(2)><TAM(4)>
A:<eqp><NUM(2)><VER(24)>
<STIMPR(1)><STETQ(1)><STFIP(1)><STALFA(1)><STLEDS(1)><STNEURON(1)><STPES(1)><STTEC(1)><STTREN(1)><TQTA(1)><ETQA(1)><TEC(1)><RES1(1)><RES2(1)><TXTMON(9)><DECPR(1)><DECIMP(1)><REDONDEO(1)><PAISEURO(2)><FASEEURO(1)><FECHAEURO(8)><RES3(1)><RES4(1)><DIGIMPLST(2)><DIGIMPDISP(2)><DIGPRC(2)><DIGPLU(1)><FLOTANTE(1)><VERCOD(1)><RES5(1)><RES6(1)><TIPOBAL(1)><ACCESO(1)><TIPOTALON(1)><CODBARTQT(1)><CODBARTAL(1)><CODBARENV(1)><TIPONUMTQT(1><SUPERIMP(1)><ALTABAJAVND(1)><PRCCERO(1)><CAMBIOPRC(1)><CAMBIOAUT(1)><CAJON(1)><GUARDARTQT(1)><ACUMULARTOT(1)><HAYIVA(1)><IVAENTQT(1)><RETPESO(1)><FIX(1)><TECLAPLUPRC(1)><TIPOENV(1)><CUSTOMERENTQT(1)><DTOENTQT(1)><RES7(1)><RES8(1)>

A:<RPS><ERR>

Bools can be:
'0': disconnected or not activated
'1': connected or activated

<NUM>:Number of scale to be consulted.
<TAM>:Size of data to be consulted.
‘0’: only <NUM> and <VER>
‘9999’: all data
<RESn>: Reserved

Hardware configuration:
<VER>: Scale version
<STIMPR>: printer state (bool)
<STETQ>:labeller state (bool)
<STFIP>: fip- display state (bool)
<STALFA>: alphanumeric-display state (bool)
<STLEDS>: LEDS-display state (bool)
<STNEURON>: neurone state (bool)
<STPES>: weight-cell state (bool)
<STTEC>:keypad state (bool)
<STTREN>: labelling-train state (bool)
<TQTA>: Where to print receipts
‘0’: printer
‘1’: labeller
<ETQA>: Where to print labels
‘0’: printer
‘1’: labeller
<TEC>: Type of keypad connected
‘1’: 66 keys
‘2’: 99 keys
‘3’: labelling train
‘4’: 80-key self-service
‘5’: 60-key self-service

Currency configuration:
<TXTMON>: Currency text
<DECPR>: Price decimals
<DECIMP>: Amount due decimals
<REDONDEO>: Type of round up:
‘0’: no round up
’1’: round up to unit
‘5’: round up to five
<PAISEURO> : Country to apply Euro
<FASEEURO>: Current Euro phase
‘0’: Do not apply Euro
‘1’: Euro as auxiliary currency
‘2’: Euro as principal currency
‘3’: Euro only
<FECHAEURO>: Date to begin applying Euro

Digit configuration:
<DIGIMPLST>: Amount due digits to print
<DIGIMPDISP>: Amount due digits to display
<DIGPRC>: Price digits
<DIGPLU>: Item digits
<FLOTANTE>: Allows floating decimal point
<VERCOD>: See code on display together with description

Sale configuration:
<TIPOBAL>: Current form of sale
‘1’: Sale
‘2’: Super
‘3’: Packaging
‘4’: Self-service
‘5’: Mixed
<ACCESO>: Access to items

‘0’: By PLU number
‘1’: By item code
<TIPOTALON>: Type of cheque
‘0’: no cheque
‘1’: cheque
<CODBARTQT>: Bar code on receipt
‘0’: Not activated
‘1’: Activated
<CODBARTAL>: Bar code on cheque
2: Without bar code
3: With bar code
<CODBARENV>: Type of bar code in packaging ‘1’ or ‘2’
<TIPONUMTQT>: Type of numbering of receipts
‘0’: Counter
‘1’: System
‘2’: Scale
‘3’: Date
‘4’: Vendor
‘5’: Single local network
<SUPERIMP>: Print in super mode
‘0’: Printer and labeller
‘1’: labeller only
‘2’: printer only
<ALTABAJAVND>: Subscribe/unsubscribe vendor by key (bool)
<PRCCERO>: Allows zero price (bool)
<CAMBIOPRC>: Allows item price change (bool)
<CAMBIOAUT>: Change function (bool)
<CAJON>: Drawer function (bool)
<GUARDARTQT>: Store receipts (bool)
<ACUMULARTOT>: Accumulate totals (bool)
<HAYIVA>: Working with VAT (bool)
<IVAENTQT>: VAT broken down on receipt (bool)
<RETPESO>: Return to weight mode following a sale (bool)
<FIX>: Price fixed by default (bool)
<TECLAPLUPRC>: Allows PLU/PRC key (bool)
<TIPOENV>: Type of packing
‘0’: manual
‘1’: automatic
<CUSTOMERENTQT>: Introduce customer on receipt (bool)
<DTOENTQT>: Introduce discount on receipt (bool)

## 5.7 #<bal>::=‘5’.53d.35h PROGRAMMING FOR SCALES (NOT 386)

### 5.7.1 <kbd>::=‘0’,48,30h.Programming of direct keys of a scale or section.

The keys are programmed as shown in the message. If one of the indicated PLUS is not programmed, the key remains without designated PLU, although it had one before. The key numbers that can be designated depend on the type of scale connected with this number. The user is the one who must know the number of keys of the scale.

Q:<bal><kbd><MOD(1)><DIR(2)><NKS(2)><KEY(2)><PLU(4)>..<KEY(2)><PLU(4)>
A:<RSP><ERR>
<MOD> ::=Nr. of scale or section to which the keys are programmed. If it is to section, the programming is copied to all the scales of the sections.
‘0’:Section number.
‘1’:Scale number.
<DIR> ::=Scale number to which the keys are programmed. If it is to section, the programming is copied to all the scales of the section.
<NKS> ::=Number of sections sent in the message. 1 for one key.
<KEY> ::=Key to be programmed.
<PLU> ::=Plu associated to key.
<ERR> ::=Error code.

**5.7.2 <?kbd>::='1',49d,31h. Request of keys of a scale or section.**

Query which are the PLUs associated to each key of the scales.
Q:<bal><?kbd><MOD(1)><DIR(2)><NKS(2)><KEY(2)>
A:<bal><MOD(1)><DIR(2)><NKS(2)><KEY(2)><PLU(4)>..<KEY(2)><PLU(4)>
A:<RSP><ERR>

The keys with numbers higher or the same as the <KEY> given in the question, are returned. The equipment returns as many keys as <NKS> asks or until no more are found. If the quantity requested is superior to the quantity that a message can contain, the maximum possible will be returned.

<MOD> ::=Programming of scale or section.
'0':Section number.
'1':Scale number .(Not 8031)(Not 8031)
<DIR> ::= Scale or section number whose keys are requested.
<NKS> ::=Number of keys requested and number or keys sent in the reply message. 1 for one key.
<KEY> ::=Key.
<PLU> ::=Plu associated to a key.
<ERR> ::=Error key.

**5.7.3 <cbr>::='2', 50d, 32h.Programming of bar codes in scales or sections networkwork**

The configuration of the bar codes in the network is programmed for the different work modes that the scales have.
Q:<bal><bar><MOD(1)><DIR(2)><VNT(12)><SUP(12)><MIX(12)><EV1(12)><EV2(12)>
A:<RPS><ERR>
<MOD> ::=The programming is addressed to a concrete scale or to a whole section.
'0'-a section
'1'-an equipment (Not 8031)
<DIR> ::=If MOD=1 indicates the number os scale, if MOD=0 the number of section.
<VNT> ::=Bar code used in sale.
<SUP> ::=Bar code used in super mode.
<MIX> ::=Bar code used in mixt mode.
<EV1> ::=Bar code used in packaging mode 1.
<EV2> ::=Bar code used in packaging mode 2.
<ERR> ::=Error code.
A string of positions with the following characters must be sent for all the codes:
S:detection of overflow in a field.
W:a weight digit
Z: a price digit
I: a import digit
T: a receipt number digit
P: a PLU number digit
C: a code of PLU digit
V: a vendor number digit
A: a the number of items in the receipt digit
L: a number of lot digit
F: a manufacturer code digit
M: a country code digit
X: a *their turn* customer code digit
D: a weight digit expressed with 2 decimals
0-9:the numbers from 0 to 9.

Examples: The required bar code is : 25, section, receipt number and cost. It is coded (in this case the section is 2) :252TTTIIIII. 25,section,receipt number, weight : 252TTTWWWWW. 2,section, import: 200001IIIII (the section is 00001). 2 plu code weight : 2CCCCCPPPPP. UPC code, plu code ,import: 0CCCCCIIIII.

Since the fields can be programmed with less digits than necessary (i.e cost with 3 numbers), there is an additional field that indicates overflow. This field is coded with the letter "S".

**5.7.4 <?cbr>::='3'.51d,33h.Request for codes in scales and section networkwork**

Q:<bal><?cbr><MOD(1)><DIR(2)>
A:<bal><MOD(1)><DIR(2)><VNT(12)><SUP(12)><MIX(12)><EV1(12)><EV2(12)>

<MOD> ::= Requests information from a concrete equipment or a section.
. '0'-a section
'1'-an equipment (Not 8031)
<DIR> ::= Id MOD=1 indicates the number of equipment, if MOD=0 the section number.
<VNT> ::=Bar codes used in sale.
<SUP> ::=Bar code used in super mode.
<MIX> ::=Bar code used in mixt mode.
<EV1> ::=Bar code used in packaging mode 1.
<EV2> ::=Bar code used in packaging mode 2.
<ERR> ::=Error code.

A string of position 12 will be received with the same characters used for programming (see message above).

**5.7.5 <loc>::='4'.52d,34h. Order of requesting scales totals in local (Not 8031)**

Q:<bal><loc><MOD(1)><DIR(2)><STS(1)>.
A:<RPS><ERR>.

<MOD> ::= Requests information from a concrete equipment or from a section.
'0'-a section
'1'-an equipment
<DIR> ::=If MOD=1 indicates the number of equipment, if MOD=0 the section number.
<STS> ::= Operation to be made.
'0': Initiating the totals request in local to the scales.
'1':Clear the pending local request.

**5.7.6 <kbd1>::='5',53d,35h. Programming of direct keys table of scale or section.**

The keys are programmed as shown in the message. If one of the indicated items is not programmed, the key will not have a designated item, although it had one before.

The key numbers that can be designated depend on the type of scale connected with this number. The user is the one who must know the number of keys of the scale.

Q:<bal><kbd1><MOD(1)><DIR(2)><TAB(2)><MOA(1)><NKS(2)><KEY(2)><IDA(6)>
.
<KEY(2)><IDA(6)>
A:<RSP><ERR>

<MOD> ::=Programming of scale or section.
'0':Section number.
'1':Scale number.
<DIR> ::=Scale or section number to which the keys are programmed.
<TAB> ::=Board of 0 to 9 keys
<MOA> ::= Selection of access key of item
'0'-<IDA> is interpreted as <00><PLU(4)>
'1'-<IDA> is interpreted as <COD(6)>
<NKS> ::= Number of keys sent in the message. 1 for one key.
.<KEY> ::=Key to be programmed.
<IDA> ::=Identifier of access key of item.
Changes according to the selector of access key of item.

<ERR> ::=Error code.

**5.7.7 <?kbd1>::='6',54d,36h. Request for keys table of a scale or section.**

Consulting which are the items associated to each scale key boards.

Q:<bal><?kbd><MOD(1)><DIR(2)><TAB(2)>><MOA(1)> <NKS(2)><KEY(2)>
A:<bal><MOD(1)><DIR(2)><TAB(2)><MOA(1)><NKS(2)><KEY(2)><IDA(6)>
<KEY(2)><IDA(6)>
A:<RSP><ERR>

The keys whose number is higher or the same as <KEY> given in the question, are returned. Returns as many keys as requested <NKS> or until no more are found. If there are more requested than the allowed in the message, the maximum possible are returned.

<MOD> ::=Programming of scale or section.
'0':Section number.
'1':Scale number.
<DIR> ::=Scale or section number from which the keys are requested.
<TAB> ::=Board of 0 to 9 keys.
<MOA> ::= Selection of access key of item
'0'-<IDA> is interpreted as <00><PLU(4)>
'1'-<IDA> is interpreted as <COD(6)>
<NKS> ::=Number of keys requested and number of keys sent in the reply message. 1 for one key.
<KEY> ::=Key.
<IDA> ::=Identifier of item's access key of item.
Changes according to the selector of item access key of item.
<ERR> ::=Error code.

**5.7.8 <kbd2>::='7',55d,37h. Programming of direct keys tables of scale or section for a maximum of 255 keys per table.**

The message is the same as the message '5'. The only difference is the length of the message.

Q:<bal><kbd2><MOD(1)><DIR(2)><TAB(2)><MOA(1)><NKS(2)><KEY(3)><IDA(6)>
<KEY(3)><IDA(6)>
A:<RSP><ERR>

**5.7.9 <?kbd2>::='8',56d,38h. Request for tables of keys of a scale or section for tables of a maximum 255 keys**

Q:<bal><?kbd><MOD(1)><DIR(2)><TAB(2)>><MOA(1)> <NKS(2)><KEY(3)>
A:<bal><MOD(1)><DIR(2)><TAB(2)><MOA(1)><NKS(2)><KEY(3)><IDA(6)>
<KEY(3)><IDA(6)>
A:<RSP><ERR>

The message is the same as message `6`. The only difference is the lenght of the message.

**5.7.10 <cbr>::='9', 57d, 39h.Programming of bar codes in scale or section local**

Programmes the bar codes configuration in local for the different work modes .

Q:<bal><bar><MOD(1)><DIR(2)><VNT(12)><SUP(12)><MIX(12)><EV1(12)><EV2(12)>
A:<RPS><ERR>
The same as the programming of the bar codes in network.

**5.7.11 <?cbr>::=':'.58d,3ah. Request of bar codes in scales or section local**

Q:<bal><?cbr><MOD(1)><DIR(2)>
A:<bal><MOD(1)><DIR(2)><VNT(12)><SUP(12)><MIX(12)><EV1(12)><EV2(12)>

The same as the consulting of bar codes in network.

## 5.8 #<bal>::='5'.53d.35h PROGRAMMING OF BAR CODES AND KEYS (NO CONCENTRATOR)

### 5.8.1 <cbr>::='2', 50d, 32h.Programming of bar codes in scale or section networkworks

Programmes the bar codes configuration for the different work modes that the scales in network have.

Q:<bal><bar><MOD(1)><DIR(2)><VNT(12)><SUP(12)><MIX(12)><ENV1(12)><ENV2(12)>
A:<RPS><ERR>
<MOD>::= The programming is addressed to a concrete scale or to an entire section.
'0'-a section
'1'-an equipment
<DIR>::=If MOD=1 indicates the scale number, if MOD=0 the section number.
<VNT>::=Bar codes used in sales mode.
<SUP>::=Bar codes used in super mode.
<MIX>::=Bar codes used in mixt mode.
<ENV1>::= Bar codes used in packaging mode 1.
<ENV2>::= Bar codes used in packaging mode 2.
<ERR>::=Error code.

A string of positions with the following characters must be sent for all the bar codes:
S: Detection of overflow in a field.
W: a weight digit
U: a unit digit
Z:a price digit
I: an amount due digit
J: an amount due digit in national currency (Euro area)
T: a number or receipt digit
P:a PLU number digit
C: a PLU code digit
V: a vendor's number digit
A: a digit of number or items in the receipt
L:a number or lot digit
F:a manufacturer's code digit.
M: a country code digit
X: a customer's code *their turn* digit
D: a weight digit expresssed in decens of grammes
N:an equipments number digit
Y:a counter's number digit
O: an order number digit
K: intermediate check of first 6 positions (only valid in position 7)
0-9:the numbers from 0 to 9

Examples: The required bar code is : 25, section, receipt number and cost. It is coded (in this case the section is 2) :25YTTTIIIII. 25,section,receipt number, weight : 252TTTWWWWW. 2,section, cost: 200001IIIII (the section is 00001). 2 plu code weight : 2CCCCCPPPPP. UPC code, plu code ,cost: 0CCCCCIIIII.

Since the fields with less digits than necessary can be programmed (i.e cost with 3 numbers) there is an additional field that shows overflow, which is coded with letter S.

### 5.8.2 <?cbr>::='3'.51d.33h. Request for bar codes in the scales' networkwork.

Q:<bal><?cbr><MOD(1)><DIR(2)>
A:<bal><MOD(1)><DIR(2)><VNT(12)><SUP(12)><MIX(12)><ENV1(12)><ENV2(12)>

<MOD>::= Requests information from a concrete equipment. It is only allow to consult one scale, since per section it only coincides in case the programming has been made per section.
'1'-one equipment
<DIR>::= Indicates the equipment number consulted.
<VNT>::=Bar codes used in sales.

<SUP>::=Bar codes used in super mode.
<SUP>::=Bar codes used in super mode.
<MIX>::=Bar codes used in mixt mode.
<ENV1>::= Bar codes used in packaging mode 1.
<ENV2>::= Bar codes used in packaging mode 2.
<ERR>::= Error code.

A string of positions 12 will be received with the same characters used for programming (see message below).

**5.8.3 <kbd2>::='7',55d,37h. Programming for direct keys tables of a scale or section for a maximum of 255 keys.**

Q:<bal><kbd2><MOD(1)><DIR(2)><TAB(2)><MOA(1)><NKS(2)><KEY(3)><IDA(6)>
..
<KEY(3)><IDA(6)>
A:<RSP><ERR>

The keys are programmed as shown in the message. If one of the indicated items is not programmed, the key will not have a designated item, although it had one before.
The key numbers that can be designated depend on the type of scale connected with this number. The user is the one who must know the number of keys of the scale.

<MOD>::=Programming of scale or section.
'0':Section number.
'1':Scale number.
<DIR>::=Number of scale or section to which the keys are programmed.
<TAB>::= 0 to 9 board number to be programmed
<MOA>::= Selection of access key of item
'0'- <IDA> is interpreted as <00><PLU(4)>
'1'- <IDA> is interpreted as <COD(6)>
This parameter has to correspond with the one of the configurated scales.

<NKS>::=Number of keys sent in the message.1 for one key.
<KEY>::=Key to be programmed.
<IDA>::=Identifier of access key of item.
<ERR>::=Error code.

**5.8.4 <?kbd2>::='8',56d,38h. Request for tables of keys of a scale of section for tables of maximum 255 keys.**

Consulting which are the items associated to each key of key boards of the scales.

Q:<bal><?kbd><MOD(1)><DIR(2)><TAB(2)>><MOA(1)> <NKS(2)><KEY(3)>
A:<bal><MOD(1)><DIR(2)><TAB(2)><MOA(1)><NKS(2)><KEY(3)><IDA(6)>..
<KEY(3)><IDA(6)>
A:<RSP><ERR>

The keys whose number is higher or the same as <KEY> given in the question, are returned. Returns as many keys as requested <NKS> or until no more are found. If there are more requested than the allowed in the message, the maximum possible are returned.

<MOD>::='1':Scale number, the query is only allowed by scale number.
<DIR>::=Scale number from which the keys are requested.
<TAB>::= Board number.
<MOA>::= Selection of access key of item.
'0'- <IDA> is interpreted as <00><PLU(4)>
'1'- <IDA> is interpreted as <COD(6)>
This parameter has to correspond to the one of the configurated scales.

<NKS>::=Number of keys requested and number of keys sent in the reply message.1 for one key.
<KEY>::=Key to be programmed.
<IDA>::=Identifier of access key of item.
<ERR>::=Error code.

**5.8.5 <cbr>::='9', 57d, 39h.Programming of bar codes in scale or section networkwork total.**

The bar codes configuration is programmed for the different work modes that the scales in local have.

Q:<bal><bar><MOD(1)><DIR(2)><VNT(12)><SUP(12)><MIX(12)><ENV1(12)><ENV2(12)>
A:<RPS><ERR>

<MOD>::= Requests information from a concrete equipment. It's only allowed to consult one scale, since per section it only coincides in case the programming has been made per section.
'1'-one equipment
<DIR>::= Indicates the equipment number consulted.
<VNT>::=Bar codes used in sales.
<SUP>::=Bar codes used in super mode.
<SUP>::=Bar codes used in super mode.
<MIX>::=Bar codes used in mixt mode.
<ENV1>::= Bar codes used in packaging mode 1.
<ENV2>::= Bar codes used in packaging mode 2.
<ERR>::= Error code.

A string of positions with the following characters must be sent for all the bar codes:
S: Detection of overflow in a field.
W: a weight digit
U: a unit digit
Z:a price digit
I: an amount due digit
J: an amount due digit in national currency (Euro area)
T: a number or receipt digit
P:a number of PLU digit
C: a PLU code digit
V: a vendor's number digit
A: a difit number or items in the receipt
L:a number or lot digit
F:a manufacturer's code digit
M: a country code digit
X: a customer's code *their turn* digit
D: a weight digit expresssed in decens of grammes
N:a the equipments number digit
Y:a counter's number digit
O: an order number digit
K: an intermediate check of the first 6 positions (only valid in position 7)
0-9:the numbers from 0 to 9

Examples: The required bar code is : 25, section, receipt number and cost. It is coded (in this case the section is 2) :25YTTTIIIII. 25,section,receipt number, weight : 252TTTWWWWW. 2,section, cost: 200001IIIII (the section is 00001). 2 plu code weight : 2CCCCCPPPPP. UPC code, plu code ,cost: 0CCCCCIIIII.

Since the fields with less digits than necessary can be programmed (i.e cost with 3 numbers) there is an additional field that shows overflow. This field is coded with letter S.

**5.8.6 <?cbr>::=':'.58d,3Ah. Request for bar codes in scales local**

Q:<bal><?cbr><MOD(1)><DIR(2)>
A:<bal><MOD(1)><DIR(2)><VNT(12)><SUP(12)><MIX(12)><ENV1(12)><ENV2(12)>

<MOD>::= Requests information from a concrete equipment. It is only allowed to consult one scale, since per section it only coincides in case the programming has been made per section.
'1'-one equipment
<DIR>::= Indicates the number of equipment consulted.
<VNT>::=Bar codes used in sales.
<SUP>::=Bar codes used in super mode.

<SUP>::=Bar codes used in super mode.
<MIX>::=Bar codes used in mixt mode.
<ENV1>::= Bar codes used in packaging mode 1.
<ENV2>::= Bar codes used in packaging mode 2.
<ERR>::= Error code.

A string of positions 12 will be received with the same characters used for programming (see message below).

**5.8.7 <cbb>::=';', 59d, 3bh.Programming of bar codes in scales or section networkwork local.**

The bar codes configuration is programmed for the different work modes that the scales in local have.

Q:<bal><cbb><MOD(1)><DIR(2)><QUE(2)><VNT(12)><SUP(12)><MIX(12)><ENV1(12)><ENV2(12)>
<TOT1(12)><TOT2(12)>
A:<RPS><ERR>

<MOD>::=Programming of scale or section.
'0':Section number.
'1':Scale number.
<DIR>::=Number of scale or section to which the keys are programmed.
<TAB>::= 0 to 9 board number to be programmed
<MOA>::= Selection of access key of item
'0'- <IDA> is interpreted as <00><PLU(4)>
'1'- <IDA> is interpreted as <COD(6)>
This parameter has to correspond with the one of the configurated scales.

<NKS>::=Number of keys sent in the message.1 for one key.
<KEY>::=Key to be programmed.
<IDA>::=Identifier of access key of item.
<ERR>::=Error code.

A string of positions with the following characters must be sent for all the bar codes:
S: Detection of overflow in a field.
W: a weight digit
U: a unit digit
Z:a price digit
I: an amount due digit
J: an amount due digit in national currency (Euro area)
T: a digit of number or receipt
P:a PLU number digit
C: a PLU code digit
V: a vendor's number digit
A: a digit of number or items in the receipt.
L:a digit of number or lot
F:a manufacturer's code digit
M: a country code digit
X: a customer's code *their turn* digit
D: a weight digit expresssed in decens of grammes.
N:an equipments number digit
Y:a digi counter's number digit
O: an order number digit
K: intermediate check of the first 6 positions (only valid in position 7)
0-9:the numbers from 0 to 9

Examples: The required bar code is : 25, section, receipt number and cost. It is coded (in this case the section is 2) :25YTTTIIIII. 25,section,receipt number, weight : 252TTTWWWWW. 2,section, cost: 200001IIIII (the section is 00001). 2 plu code weight : 2CCCCCPPPPP. UPC code, plu code ,cost: 0CCCCCIIIII.

Since the fields with less digits than necessary can be programmed (i.e cost with 3 numbers) there is a supplementary field that shows overflow, which is coded with letter S

**5.8.8 <?cbb>::='<'.60d,3ch. Request for bar codes in the scales networkwork or local**

Q:<bal><?cbb><MOD(1)><DIR(2)><QUE(2)>
A:<bal><MOD(1)><DIR(2)><VNT(12)><SUP(12)><MIX(12)><ENV1(12)><ENV2(12)><TOT1(12)>
<TOT2(12)>

<MOD>::= Requests information of a concrete equipment. It is only allowed to consult one scale, since per section it only coincides in case the programming has been made per section.
'1'-one equipment
<DIR>::= Indicates the equipment number consulted.
<VNT>::=Bar codes used in sales.
<SUP>::=Bar codes used in super mode.
<SUP>::=Bar codes used in super mode.
<MIX>::=Bar codes used in mixt mode.
<ENV1>::= Bar codes used in packaging mode 1.
<ENV2>::= Bar codes used in packaging mode 2.
<ERR>::= Error code.

A string of positions 12 will be received with the same characters used for programming (see message below.

**5.9 #<vnd>::='6',54d,36h-VENDORS.**

Message relative to the creation and maintenance of the scales vendors.

**5.9.1 <prg>::='0',48,30h.Creation/Modification of vendor. (Not 8031)**

Creates or modifies the scales vendor.

Q:<vnd><prg><IDV(4)><TXT(30)><COD(6)>
A:<RPS><ERR>

<IDV> ::=Identification of vendor.
8031 concentrator [SSVV] (Section*100+Vendor) <IDV>::=<SS><VV> (This message is not attended and is replyed ACK).
PC concentrator, unique vendor Code of three digits.
<TXT> ::=Name of vendor.
<COD> ::=Vendor's password. This code will be used in the operation of Registration of vendor to work, made form the scale (Not 386)
<ERR> ::=Error code.

The vendors will be kept in all the scales of the system. They will work only in a section. (Only 386)

**5.9.2 <bor>::='1',49d,31h.Clearing of vendor. (Not 8031)**

Clear a vendor of the Database.

Q:<vnd><bor><IDV(4)><STS(1)>
A:<RPS><ERR>

<IDV> ::=Identification of vendor
8031 Concentrator [SSVV] (Section*100+Vendor) <IDV>::=<SS><VV> (This message is not attended and is replyed ACK).
PC concentrador , unique vendor code.
<STS> ::= If at the moment of clearing there are accumulated data in the accumulated total (daily), can be marked as cleared. The physical clearing will be made when Clearing the great absolute total of the Concentrator (not 386).
'0':Is cleared physically.
'1': Only marked as cleared.
'2': Unmarked of cleared.
<ERR> ::=Error code.

**5.9.3 <alt>::='2',50d,32h. Introducing a new vendor (Not 8031)**

This message enables the designation of a vendor to a vendor key of the scales so that it is possible to work. This operation can be made directly in the scales.

Q:<vnd><alt><IDV(4)><SEC(2)><KEY(2)>
A:<RPS><ERR>

<IDV> ::=Identifier of vendor.
8031 concentrator , [SSVV] (Section*100+Vendor) <IDV>::=<S><VV> (This message is not attended and is replyed ACK).
PC concentrador, unique vendor code.
<SEC> ::= Section in which the new vendor is introduced. It will work in any of the scales of the section.
<KEY> ::=Key of the scales of the section used by the vendor.
.<ERR> ::=Error code.

**5.9.4 <baj>::='3',51d,33h.Temporary leave of vendor (Not 8031)**

The vendor stops working in the scales vendor key that was assigned to him.
To test that it is made correctly,it would be necessary to make a clearing of open receipts (only 386).

Q:<vnd><baj><IDV(4)>
A:<RPS><ERR>

<IDV> ::=Identification of vendor
If <IDV> is zero, all the vendors leave, except if there are some with open totals.
<ERR> ::=Error code.

**5.9.5 <cer>::='4',52d,34h.Closing of open totals of vendors.**

To retire a vendor from his key, he shouldn't have open receipts. If the vendor has finished working with a non-closed receipt, the receipts can be closed with this option.

Q:<vnd><cer><IDV(4)>
A:<RPS><ERR>

<IDV> ::=Identifier of vendor.
In PC concentrator and in 386 if <IDV> is zero, the totals of all the vendors are closed.
In 8031 <SSVV> if <VV> is zero, the totals of all the vendors are closed.

<ERR> ::=Error code.

**5.9.6 <sts>::='5',53d,35h.Consulting vendor status (Not 8031)**

Consulting the data associated to the vendor. Apart from the programming data, the data calculated by the concentrator are shown and those that indicate its working situation at the moment of the query.

Q:<ven><sts><QUE(1)><IDV(4)>
A:<ven><IDV(4)><TXT<30><COD(6)<STS(1)><TME(8)><OPN(1)><SEC(2)><KEY(2)><ELI(1)>

<QUE> ::=Indicates the data to be searched.
'0':Search the same.
'1':Search more or the same.
'2':Search more.
'3':Search less or the same.
'4':Search less or the same.
<IDV> ::=Identification of vendor
8031 concentrator, [SSVV] (Sectión*100+Vendor) <IDV>::=<S><VV>
PC concentrator, unique vendor code.
<TXT> ::=Name of vendor.
<COD> ::=Vendor password . (Not 386)
<STS> ::=State.

‘0’-leave,vendor cannot work in the scales.
‘1’-new vendor, vendor working in the scales.

<TME> ::= Last time in which a new vendor or leave was made (not 386).
<OPN> ::=‘1’- Receipt on course, ‘0’ in case of the contrary. (Not 386)
<SEC> ::=Working section if it is registered.
<KEY> ::= Designated key if it is registered.
<ELI> ::=State : (Not 386)
‘0’:Active.
‘1’:Cleared.
<ERR> ::=Error code.

**5.9.7 <cvt>::=‘6’,54d,36h.Vendor query by key (Not 8031)**

Consulting the data associated to the vendor designated to the scales sales key .
In case of Concentrator, apart from the programming data, the data calculated by the concentrator are shown and those that indicate its working situation at the moment of the query.

The vendor key by key is only possible if the vendor is registered in the section consulted (registration for working in scales) or if the concentrator is configurated for automatic connexion of vendors. In this case, there is always a direct relation between vendor code and section+associated key.

Q:<ven><ctv><QUE(1)><SEC(2)><KEY(2)>
A:<ven><IDV(4)><TXT<30><COD(6)<STS(1)><TME(8)><OPN(1)><SEC(2)><KEY(2)><ELI(1)>

<QUE> ::=Indicates the data to be searched.
‘0’:Search the same. (386 only accepts this value)
‘1’:Search more or the same.
‘2’:Search more.
‘3’:Search less or the same.
‘4’:Search less or the same.
<IDV> ::=Identification of vendor
8031 concentrator, [SSVV] (Section*100+Vendor) <IDV>::=<S><VV>
PC concentrator, unique vendor code.
<TXT> ::=Name of vendor.
<COD> ::=Vendor password. (Not 386)
<STS> ::=State.
‘0’-leave,vendor cannot work in the scales.
‘1’-new vendor, vendor working in the scales.
<TME> ::= Last time in which a new vendor or leave was made (not 386).
<OPN> ::=‘1’- Receipt on course, ‘0’ in case of the contrary. (Not 386)
<SEC> ::=Working section if it is registered.
<KEY> ::= Designated key if it is registered.
<ELI> ::=State : (Not 386)
‘0’:Active.
‘1’:Cleared.
<ERR> ::=Error code.

For scales system, the data are the same as in the previous consulting.

**5.9.8 ::=‘7’,55d,37h.Consulting further data of vendor (Not 8031)**

Q:<ven><cmv><QUE(1)><IDV(4)>
A:<ven><IDV(4)><TME(8)><TQT(5)><NLI(2)><CON(1)>

<QUE> ::=Indicates the data to be searched.
‘0’:Search the same.
‘1’:Search more or the same.
‘2’:Search more.
‘3’:Search less or the same.
‘4’:Search less or the same.
<IDV> ::=Identification of vendor
8031 concentrator, [SSVV] (Sectión*100+Vendor) <IDV>::=<S><VV>
PC concentrator, unique vendor code.

<TXT> ::=Name of vendor.
<COD> ::=Vendor password . (Not 386)
<STS> ::=State.
‘0’-leave,vendor cannot work in the scales.
‘1’-new vendor, vendor working in the scales.
<TME> ::= Last time in which a new vendor or leave was made (not 386).
<OPN> ::=‘1’- Receipt on course, ‘0’ in case of the contrary. (Not 386)
<SEC> ::=Working section if it is registered.
<KEY> ::= Designated key if it is registered.
<ELI> ::=State : (Not 386)
‘0’:Active.
‘1’:Cleared.
<ERR> ::=Error code.

### 5.10 #<art>::=‘7’,55d,37h-ITEMS

Messages associated with the programming and information of items contained in the internal database of the concentrator or in the system.

#### 5.10.1 <prg>::=‘0’,48d,30h. Basic programming of item.

**In the case of PC concentrator,** the item is introduced if it doesn’t exist or modified if it exists. Before this, create the section, subsection and familiy to which the item is designated.

**In the case of scales system,** the item is inserted if it doesn’t exist or modified if it exists. The item is programmed to all the scales,

- if the access mode in the scales is CODES, use as identification the field COD
- if the access mode is PLU, it is identified with the PLU+SEC field. It is compulsory to programme the section (necessary if an independent numbering per section is wanted)

The family is an independent group of section.

Q:<art><prg><COD(6)><SEC(2)><PLU(4)><SUB(3)><FAM(4)><PRC(7)><WGH(1)><CAD(3)><TLT(1)><TXT(25)><TAR(15)>
A:<RSP><ERR>
<COD> ::=Identifier (code) of item (6 digit).
PC concentrator : unanimous internal code. If there is not previous codification, it is convenient to use <SSCCCC> in which SS are two digits for the section code and CCCC 4 digits for the PLU code. Thus <COD> can be interpreted as <SEC><PLU>. If this codification is carried out as <C0D(6)>==<SEC(2)><PLU(4)>.
8031 concentrator :unanimous internal code.
If there is no previous codification, it is convenient to use [0S0PPP] Section*1000+Plu
(Not 386)
<SEC> ::=Section code (2 digits). In the case of 386, the introduction is not compulsory.
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section. If in the concentrator the option of designate it automatically is actiVATed, introduce a value different to zero when wanting to do it.
<SUB> ::=Subsection code (Group) (3 digits). Not used neither in 8031 concentrator nor in 386, which is 0.
<FAM> ::=Family code. In the case of 386 its introduction is not compulsory.
<PRC> ::=Price of item
<WGH> ::=Type of sale:
‘W’= weight item.
‘U’= non-weight item.
<CAD> ::= Item expiry date of item (Not 386).
<TLT> ::= Type of letter of descriptive, interpreted in each scale (Not 386).
‘0’=Standard letter
‘1’=Letter twice as high.
‘2’=Letter twice as wide.
‘3’=Letter twice as high and twice as wide.
<TXT> ::=Descriptive line.
<TAR> ::=Associated tare. (Not 8031)

<ERR> ::=Error code.

**5.10.2 <?prg>::='1',49d,31h.Request for information about item.**

Consulting the data programmed for an item.

Q:<art><?prg><QUE(1)><MOD(1)><IDA(6)>
A:<art><COD(6)><SEC(2)><PLU(4)><SUB(3)><FAM(4)><PRC(7)><WGH(1)><CAD(3)><TLT(1)><TXT(25)><TAR(5)><PRN(7)><DTE(8)><ELI(1)>
A:<RSP><ERR>

<QUE> ::=Indicates the data to be searched.
'0':Search the same.
'1':Search more or the same.
'2':Search more.
'3':Search less or the same.
'4':Search less.

<MOD> ::=Selection of password.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
In 386, this parameter has to correspond with the configurated scales.

<IDA> ::= Identifier of password. Changes according to the selector of password.
In 386, has to correspond with the access mode of the scales.

<COD> ::=Identifier of item (6 digits)

<SEC> ::=Section code (2 digits).

<PLU> ::=Plu code (4 digits). Item request code in a section given scale.

<SUB> ::=Subsection code (3 digits). Non used neither in 8031 concentrator 8031 nor in 386.

<FAM> ::=Family code.

<PRC> ::= Sale price of item.

<WGH> ::=Type of sale:
'W'= weight item.
'U'= non-weight item.

<CAD> ::= Item expiry date (Not 386).

<TLT> ::= Type of letter of descriptive, interpreted in each scale (Not 386).
'0'=Standard letter
'1'=Letter twice as high.
'2'=Letter twice as wide.
'3'=Letter twice as high and twice as wide.

<TXT> ::=Descriptive line.

<TAR> ::=Associated tare. (Not 8031) (Not 386)

<PRN> ::=New price .(Not 8031) (Not 386)

<DTE> ::=Date in which the change of price will take place (Not 8031).

<ELI> ::=State :(Not 8031) (Not 386)
'0':Active.
'1':Cleared.

<ERR> ::=Error code.

**5.10.3 <bor>::='2',50d,32h.Erasing item.**

Q:<art><bor><MOD(1)><IDA(6)><STS(1)>
A:<RSP><ERR>

<MOD> ::=Selection of password.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
In 386, this parameter has to coincide with the one of the scales configuration.

<IDA> ::=Identifier of password. Changes according to the password selector <MOD>
In 386, it has to coincide with the access mode of the scales.

<STS> ::= If when clearing there are accumulated data in the accumulated total (daily), can be marked as cleared. The physical clearing will be made at the moment of clearing the global absolute total of the concentrator:
'0':Cleared physically.
'1':Only marked as cleared.(Not 8031)
'2':Desmarked as cleared.(Not 8031)

In 386, must be ´0´. Not used.

<ERR> ::=Error code.

**5.10.4 <txt>::='3',51d,33h.Prog Item text**

**For Concentrator:**

Q:<art><txt><MOD(1)><IDA(6)><TYP(1)><LIN(2)><LNS(1)><TLT(1)>
<TXT(max25)>..<TLT(1)><TXT(max25)>
A:<RSP><ERR>

**For scales system:**

Q:<art><txt><MOD(1)><IDA(6)><TYP(1)><LIN(2)><LNS(1)><TLT1(1)>
<TXT1(máx 25)>..<TLT5(1)><TXT5(máx 25)>
A:<RSP><ERR>

It is sent in lines of 5 as a maximum.

<MOD> ::= Selection of password.
'0'- <IDA> is interpreted as <SEC(2)><PLU(4)>
'1'-<IDA> is interpreted as <COD(6)>

In 386, this parameter has to correspond to the configurated scales.
<IDA> ::=Identifier of password. Changes according to the selector of password.
<COD> ::=Item identifier (6 dígits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<LIN> ::= Number of the first line sent begining by zero.
<LNS> ::= Number of descriptive lines. In 386, if zero, clears all the lines.
<TYP> ::= Descriptive used in sale or in packaging.
'0' Short descriptive. Value by default when creating the item of one line. (Not 386)
'1' Descriptive four . Maximum four lines of descriptive (Not 386)
(8031 concentrator: packaging section)
'2' Large descriptive, of some lines. (Not 8031)

In the 8031 concentrador are only used type 0 and 1 are used.
In the PC concentrador, when receiving the message with <LIN>=0 , any descriptive of the same type existing for the item is cleared and the new programme lines are inserted. The programming must be consecutive to programme all the lines.
<TXT>::=Text of the descriptive. Maximum 25 characters. (Only 386)
<ERR> ::=Error code.

**5.10.5 <?txt>::='4',52d,34h.Requesting item text.**

Q:<art><?txt><MOD(1)><IDA(6)><TYP(1)><LIN(2)>
A:<art><IDA(6)><TYP(1)><LIN(2)><LNS(1)><TLT(1)><TXT(max25)>.. <TLT(1)><TXT(max25)>
A:<RSP><ERR>
In the case of scales system, the lines are sent in groups of 5 as a maximum.

<MOD> ::=Selection of password.
'0': <IDA> is undestood as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>

In 386, this parameter has to correspond to the one of the scales that are configurated.

<IDA> ::=Identifier of password. Changes according to the selector of password.
<COD> ::=Identifier of item (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<LIN> ::= Number of the first line sent.
<LNS> ::=Number of lines. In concentrator, if it is '0', it doesn't have programmed lines of this type.
<TXT>::=Text of 25 characters as a maximum. (Only 386)
<ERR> ::=Error code.

**5.10.6 <prc>::='5',53d,35h. Change of item price (Not 8031).**

Q:<art><prc><MOD(1)><IDA(6)><PRC(7)><DTE(8)>
A:<RSP><ERR>

<MOD> ::=Selection of password.
'0': <IDA> is undestood as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
In 386, this parameter has to correspond with the one of the scales that are configurated.
<IDA> ::=Identifier of password. Changes according to the selector of password.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section or counter code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section
<DTE> ::= Date in which the new price must be activated. The activation will be made at the moment of the sale in one of the equipments. If the actiVATion date is posterior to than the actual date of the concentrator. If the date is the same as the actual date or if the field is not informed, left blank or zeros, the price change is inmediate. In 386 it is not used (use message "E").
<ERR> ::=Error code.
.

**5.10.7 Updating item stock (Not 386)**

Q:<art><prc><MOD(1)><IDA(6)><PRC(7)><DTE(8)>
A:<RSP><ERR>
<MOD> ::=Selection of password.
'0': <IDA> is undestood as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>

<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::= Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<ACT> ::=Type of updating.
'0': Introduction absolute stock STK
'1': Increase stock in quantity STK (Can be positive or negative).
'2': Decrease stock in quantity STK.(Not 8031)
'3': Clear stock of the section.
'4': Clear stock of all the items (Not 8031)
<STK> ::=Stock quantity.
<ERR> ::=Error code.

**5.10.8 <?stk>::='7',55d,37h. Request item stock. (Not 386)**

Q:<art><?stk><MOD(1)<IDA(6)>
A:<art><IDA(6)><STK(9)>

<MOD> ::=Selection of password.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<STK> ::=Stock quantity.

**5.10.9 <env>::='8',56d,38h. Ear-mark item to send always to the scales. (Not 386)**

Q:<art><env><MOD(1)<IDA(6)><ENV(1)>
A:<RSP><ERR>

<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.

<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section
<ENV> ::=Marking status to send to scales (will be always in local):
'0': Don't send always to the scales.
'1': Send always to the scales.
<ERR> ::=Error code.

**5.10.10 <?env>::='9',57d,39h. Consulting marked item to be sent always to scale (Not 386)**

Q:<art><?env><MOD(1)<IDA(6)>
A:<art><IDA(6)><ENV(1)>

<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section
<STK> ::=Quantity in stock.
<ENV> ::=Marking status to send to scales (will be always in local):
'0': Don't send always to the scales.
'1': Send always to the scales.

**5.10.11 <vst>::=';',58d,3ah. Consulting item stock data (Not 386)**

Q:<art><vst><MOD(1)<IDA(6)><MIN(9)><OPT(9)>
A:<RSP><ERR>

<MOD> ::= Password selection
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<MIN> ::= Minimum stock.
<OPT> ::= Optimum stock .
<ERR> ::=Error code.

**5.10.12 <?vst>::=';',59d,3bh. Consulting item stock data. (Not 386)**

Q:<art><?vst><MOD(1)<IDA(6)>
A:<art><IDA(6)><MIN(9)><OPT(9)><ULT(8)>
A:<RSP><ERR>

<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<MIN> ::= Minimum stock.
<OPT> ::= Optimum stock . .
<ULT> ::= Date of last sale.
<ERR> ::=Error code.

**5.10.13 <cts>::='<',60d,3ch. Programming item costs. (Not 386)**

Q:<art><cst><MOD(1)<IDA(6)><ACT(1)><CNT(9)><PCT(7)>
A:<RSP><ERR>

<MOD> ::=Selecting password.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the selector of password.
'000000' : Zero starting of stocks and costs of all the items of the database.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a section given.
<NUM> ::= Number of prices and quantities that follow 0.10.
<ACT> ::= Way of stock updating.
'0': Absolute CNT introduction.
'1': CNT quantity increase.
'2': CNT quantity decrease.
<CNT> ::= Quantity of stock to update.
<PCT> ::= Cost.
<ERR> ::=Error code.

**5.10.14 <?cts>::='=',61d,3dh. Consulting item costs (Not 386)**

Remark : Message with added <ULT>, compatible with previous versions.

Q:<art><?cts><MOD(1)<IDA(6)>
A:<art><IDA(6)><MED(7)><MIN(7)><MAX(7)><ULT(7)>
A:<RSP><ERR>

<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Access identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<MED> ::= Medium price of item cost.
<MIN> ::= Minimum price of item cost.
<MAX> ::= Maximum price of item cost
<ULT> ::= Last price of item purchasing.
<ERR> ::=Error code.

**5.10.15 <?cv>::='>',62d,3eh. Clearing item sales. (Not 386)**

Q:<art><?cv><MOD(1)<IDA(6)><NUM(2)><PRC(7)><QUE(1)>
A:<art><IDA(6)> <NUM(2)><CNT(9)><PCT(7)>..<CNT(9)><PCT(7)>
A:<RSP><ERR>

<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 dígits). Item request code in a scale of a section given.
<NUM> ::= Number of prices and quantities that follow. The sale prices higher or the same as the requested one are consulted. Maximum 10.
<CNT> ::= Quantity sold.
<PCT> ::= Sale price of the quantity.
<QUE> ::=Indicates the prices to be searched.
'0': Search the same.

'1':Search more or the same.
'2':Search more.
'3':Search less or mo re.
'4':Search less.

<ERR> ::=Error code.

**5.10.16 <bve>::='?',63d,3fh. Clearing item sales. (Not 386)**

Q:<art><bcv><MOD(1)<IDA(6)>
A:<RSP><ERR>

<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits).Item request code in a scale of a given section.

**5.10.17 <tve>::=' ',64d,40h. Consulting item sales total. (Not 386)**

Q:<art><tcv><MOD(1)<IDA(6)>
A:<RSP><ERR>
A:<art><IDA(6)><CNT(9)>

. <MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<CNT> ::= Total of sales to the different prices.

**5.10.18 <?prgg>::='A',65d,41h.Requesting information about general item.**

Consulting the data programmed for an item using any of the possible access ways.

Q:<art><?prgg><QUE(1)><MOD(1)><COD(6)><SEC(2)><PLU(4)><SUB(3)>
<FAM(4)> <TXT(max 25)>

A:<art><COD(6)><SEC(2)><PLU(4)><SUB(3)><FAM(4)><PRC(7)><WGH(1)><CAD(3)><TLT(1)><TXT(max 25)><TAR(5)><PRN(7)><DTE(8)><ELI(1)>
A:<RSP><ERR>

<QUE> ::=Indicates the prices to be searched.
'0': Search the same.
'1':Search more or the same.
'2':Search more.
'3':Search less or more.
'4':Search less.
<MOD> ::=Selection of password.
'0': By Plu section.
'1': By code.
'2': By code section . (Not 386)
'3': By section, subsection, family, plu (Not 386).
'4':. By section, subsection, family, code (Not 386)
'5': By descriptive. (Not 386)
'6': By family/code. (Only 386)

<COD> ::=Item identifier (6 digits)
<SEC> ::= Code of section or counter (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<SUB> ::=Subsection code (3 digits). Not used neither in 8031 concentrator nor 386.

<FAM> ::=Family code.
<PRC> ::= Item sale price.
<WGH> ::=Type of sale:
'W': Weighed item.
'U': Non-weighed item.
<CAD> ::=Item expiry dates. (Not 386)
<TLT> ::=Type of letter of descriptive (Not 386)
'0': Nomal letter.
'1': Double-hight letter.
'2: Double-width letter.
'3': Double-hight and with double width letter.
<TXT> ::=Line of descritptive.
<TAR> ::=Associated tare. (Not 8031)
<PRN> ::=New price .(Not 8031) (Not 386)
<DTE> ::=Date in which the change of price will take place.(Not 8031) (Not 386)
<ELI> ::=State :(Not 8031) (Not 386)
'0': Active.
'1': Cleared.
<ERR> ::=Error code.

**5.10.19 <?log>::='B',66d,42h.Consulting item logging line. (Not 386)**

Q:<art><?log><QUE(1)><NUM(4)>
A:<art><NUML(4)><ORI(1)><TYP(1)><COD(6)><SEC(2)><PLU(4)><PRC(7)>
<PRN(7)><DT1(8)><DT2(8)><MRG(7)><COS(7)><TXT(max5)><DTE(8)>
<HOR(8)>
A:<RSP><ERR>
<QUE> ::=Indicates the data to be searched.
'0': Search the same.
'1': Search more or the same.
'2': Search more.
'3': Search less or the same.
'4': Search less.
<NUM> ::= Number of log, received operation sequence of price changes.
.<ORI> ::= Who makes the modification.
'0' : HOST.
'1' : MASTER.
<TYP> ::= Type of change of price made.
'0': Inmediate change.
'1': With date of change.
'2': Interval of dates.
'3': By margin.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code. (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section
<PRC> ::=Previous sales price of item.
<PRN> ::=Sales price modified.
<DT1> ::=Date of actiVATion of the new price, or initial date of temporary price change.
<DT2> ::= Initial date of temporary offer.
<MRG> ::= Margin used in the change of price by margin.
<COS> ::=Cost price for the change of price by margin.
<TXT> ::=Descriptive line.
<DTE> ::=Date in which the operation has been received from the HOST or the MASTER.
<HOR> ::=Time in which the operation has been received from the HOST or the MASTER.
<ERR> ::=Error code.

**5.10.20 <iva>::='C',67d,43h. Programming item VAT.**

Q:<art><iva><MOD(1)<IDA(6)><IVA(1)>
A:<RSP><ERR>
<MOD> ::=Password selection.

'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>

<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section
<IVA> ::= Type of VAT to apply.

<ERR> ::=Error code.

**5.10.21 <?iva>::='D',68d,44h. Consulting item VAT.**

Q:<art><?iva><MOD(1)<IDA(6)>
A:<art><IDA(6)><IVA(1)>
A:<RSP><ERR>

<MOD> ::=Selection of password.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<IVA> ::= Type of VAT programmed.
<ERR> ::=Error code.

**5.10.22 <cpt>::='E',69d,45h. Programming change of price for temporary offer**

Q:<art><cpt><MOD(1)><IDA(6)><PRN(7)><DT1(8)><DT2(8)>
A:<RSP><ERR>

<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 dígits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a section given.
<PRN> ::=New price.
<DT1> ::=Date in which the new price must be activated. The activation will be made at the moment of the sale in one of the equipments.
<DT2> ::=Final date of price change. The previous price in the temporary offer is activated at the moment of the sale in one of the equipments. In 386, if DT2 is the day before of DT1, the activation will remain.
<ERR> ::=Error code.

**5.10.23 <cpt?>::='F',70d,46h. Consulting change of price for temporary offer**

Q:<art><cpt?> <MOD(1)<IDA(6)>
A:<art><IDA(6)><PRN(7)><DT1(8)><DT2(8)>
A:<RSP><ERR>

<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector <MOD>.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::= Plu code (4 digits). Item request code in a scale of a given section
<PRN> ::= New price.
<DT1> ::= Date in which the new price must be activated. The activation will be made in the moment of the sale in one of the equipments.

<DT2> ::= Final date of the price change. The activation will be made in the moment of the sale in one of the equipments.
<ERR> ::=Error code.

**5.10.24 <cpm>::='G',71d,47h. Programming price change by margin (Not 386)**

Q:<art><cpm><MOD(1)><IDA(6)><PRN(7)><PRC(7)><MRG(6)>
A:<RSP><ERR>
<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<PRN> ::=New price.
<PRC> ::=Price cost.
<MRG> ::= Benefits margin ( 3 entire and 2 decimal numbers).
<ERR> ::=Error code.

**5.10.25 <cpm?>::='H',72d,48h. Consulting change of price (Not 386)**

Q:<art><cpm?> <MOD(1)<IDA(6)>
A:<art><IDA(6)><PRN(7)><PRC(7)><MRG(6)>
A:<RSP><ERR>
<MOD> ::=Selection of password.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Identifier of password. Changes according to the selector of password.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 dígits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a section given.
<PRN> ::=New price.
<PRC> ::=Price cost.
<MRG> ::= Benefits margin ( 3 entire and 2 decimal numbers).
<ERR> ::=Error code

**5.10.26 <?pxc>::='I',73d,49h. Consulting/Initializing the cost total for quantities bought of item . (Not 386)**

Remark: Message with attached <PULT> , compatible with previous versions.
Q:<art><?cts><MOD(1)<IDA(6)><STS(1)>
A:<art><IDA(6)><VAL(9)>
A:<RSP><ERR>
<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of a given section.
<STS> ::=
'0': Consult the value.
'1': Clears the value.
<ERR> ::=Error code.

**5.10.27 <?ulog>::='J',74d,4ah.Consulting user who has made the log (Not 386)**

Q:<art><?ulog><NUM(4)>
A:<art><NUM(4)><USU(3)><IDENT(16)>
A:<RSP><ERR>
<NUM> ::= Number of log inputs to be consulted. If log number is '0000' , the current active user is consulted.
<USU> ::= Number of the user who has made the operation about items.

<IDENT>::= User identifier.

**5.10.28 <pusu>::='K',75d,4bh.Programming user of current job (Not 386)**

Q:<art><pusu><USU(3)><IDENT(16)>
A:<RSP><ERR>
<USU> ::= Number of user active at this moment. To desactivate user, programme active user as '000'.
<IDENT>::= User identifier.

**5.10.29 <env>::='L',76d,4 Consulting item packaging data (Only 386)**

Q:<art><env><MOD(1)><IDA(6)><CAD(4)><PRE(4)><ETQ(2)><MAX(5)><MIN(5)>
A:<RSP><ERR>
<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of given section.
<CAD>::=Expiry dates.
<PRE>::= Best before days.
<ETQ>::= Labeller number of item (0 to 9). Specials:
- 98: Labeller of equipment by default.
- 99: Not printing receipt.
<MAX>::= Maximum weight allowed. Not used.
<MIN>::= Minimum weight alone. Not used.
<ERR>::=Error code.

**5.10.30 <csd>::='L',76d,4Ch. Programming code as a Plu in a different section (no 386)**

Q:<art><csd><FUN(1)><COD(6)><SEC(2)><SUB(3)><FAM(4)><PLU(4)
A:<RSP><ERR>
<FUN>::= '0'- Programme or modify
'1'- Clear (does not clear main code)
<COD>::=Plu code (6 digits)
<SEC>::= Section (2 digits).
<SUB>::= Subsection (3 digits, send '000').
<FAM>::= Family (4 digits, send '0000').
<PLU>::= Plu (4 digits)
The 'SUB' and 'FAM' fields are not used, but reserved for future expansion.

**5.10.31 <env?>::='M',77d,4Dh. Consulting item packaging data (Only 386)**

Q:<art><env?><MOD(1)><IDA(6)>
A:<art><IDA(6)><CAD(4)><PRE(4)><ETQ(2)><MAX(5)><MIN(5)><RSV(3)>
A:<RSP><ERR>

<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of given section.
<CAD>::=Expiry dates.
<PRE>::= Best before days.
<ETQ>::= Labeller number of item (0 to 9). Specials:
- 98: Labeller of equipment by default.
- 99: Not printing receipt.
<MAX>::= Maximum weight allowed. Not used.
<MIN>::= Minimum weight alone. Not used.
<RSV>::= Not used.

<ERR>::=Error code.

**5.10.32 <ing>::='N',78d,4Eh. Programming item ingredients (Only 386)**

Q:<art><ing><MOD(1)><IDA(6)><ING(2)><NING(2)><ING1(4)>…<ING15(4)>
A:<RSP><ERR>

<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::= Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of given section.
<ING>::= Initial ingredient. Not used.
<NING>::= Number of ingredients. Value to '15'. If zero, clear all the ingredients.
<INGn>::= Ingredient code.
<ERR>::=Error code.

There can only be ingredients or descriptives. If one of the two is programmed, the other one will be cleared.

**5.10.33 <ing?>::='O',79d,4Fh. Consulting item ingredients (Only 386)**

Q:<art><ing?><MOD(1)><IDA(6)><ING(2)>
A:<art><IDA(6)><ING(2)><NING(2)><ING1(4)>…<ING15(4)>
A:<RSP><ERR>
<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password ingredients. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of given section.
<ING>::= Initial ingredient. Not used.
<NING>::= Number of ingredients. Value to '15'.If zero, clear all the items.
<INGn>::= Ingredient code.
<ERR>::=Error code.

**5.10.34 <mer?>::='P',80d,50h. Consulting goods entrance (Only 386)**

Q:<art><mer?><QUE(1)><MER(6)><MOD(1)>
A:<art><MER(6)><IDA(6)><NUM(12)><PES(6)><DTE(8)>
A:<RSP><ERR>

<MOD> ::=Password selection.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>

<QUE>::=Indicates the data to be searched.
'0':Search the same.
'1':Search more or the same.
'2':Search more.
'3':Search less or the same.
'4':Seach less.
<MER>::=Order number of the goods entrance.
<IDA> ::=Password identifier . Changes according to the selector of password.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of given section.
<NUM>::=Certificate number.
<PES>::= Kilos received.
<DTE>::= Input date.

<ERR>::=Error code.

**5.10.35 <bmer>::='Q',81d,51h. Clearing all the goods inputs (Only 386)**

Q:<art><bmer>
A:<RSP><ERR>

<ERR>::=Error code.

**5.10.36 <cba>::='R',82d,52h. Programming bar code by item (Only 386)**

Q:<art><cba><MOD(1)><IDA(6)><CBA(12)>
A:<RSP><ERR>

<MOD> ::=Password selection. '1' at the moment.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>
'1': <IDA> is interpreted as <COD(6)>
<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of given section.
<CBA>::=Bar code for each item. In blank to clear. It is coded the same way as the sales bar codes.
<ERR>::=Error code.

**5.10.37 <cba?>::='S',83d,53h. Consulting bar code by item (Only 386)**

Q:<art><cba?> <MOD(1)<IDA(6)>
A:<art><IDA(6)><CBA(12)>
A:<RSP><ERR>

<IDA> ::=Password identifier. Changes according to the password selector.
<COD> ::=Item identifier (6 digits)
<SEC> ::=Section code (2 digits).
<PLU> ::=Plu code (4 digits). Item request code in a scale of given section.
<CBA>::=Bar code for each item. Blank doesnt' have. It is coded the same way as the sales bar codes.

<ERR>::=Error code.

**5.10.38 <ped>::='T',84d,54h. Updating an item order.**

Can only be performed from the local network (order file)

Q:<art><?stk><MOD(1)><IDA(6)><ACT(1)><PED(9)>
A:<RSP><ERR>

<MOD>::=Select access code
'0'- <IDA> is interpreted as <SEC(2)><PLU(4)> (not implemented).
'1'-<IDA> is interpreted as <COD(6)>.
<IDA>::=password identifier. Changes according to the password selector <COD>::= Item identifier (6 digits).
<SEC>::=Section code (2 digits).
<PLU>::=Plu code (4 digits).Item request code in a scale of given section.
<ACT>::=Type of update.
'0'= introduction of absolute order PED. Not implemented.
'1'= increase quantity of PED order (May be positive or negative). Not implemented.
'2'= decrease quantity of PED order. Not implemented.
'3'= Clear section order. Not implemented.
'4'= Clear orders of all items.
<PED>::= Quantity ordered.
<ERR>::=Error code.

## 5.10.38.1 <?ped>::='U',85d,55h. Requesting an item order.

Can only be performed from the local network (order file)

Q:<art><?stk><MOD(1)<IDA(6)>
A:<art><IDA(6)><PED(9)>

<MOD>::=Selection of access code
'0'- <IDA> is interpreted as <SEC(2)><PLU(4)>. Not implemented.
'1'- <IDA> is interpreted as <COD(6)>.
<IDA>::= Password identifier. Changes according to the password selector <COD>:=Item identifier (6 digits).
<SEC>::=Section code (2 digits).
<PLU>::=Plu code (4 digits).Item request code in a scale of given section.
<PED>::=Quantity ordered.

## 5.10.38.2 <prgc>::='V',86d,56h.Programming complete items.

Can only be performed from the local web (order file)

Programme all item data using any possible form of access. This message can be widened by the end.

Q:<art><prgc><COD(6)><SEC(2)><PLU(4)><SUB(3)><FAM(4)><PRC(7)><WGH(1)><TAR(5)><IVA(1)><CA
D(4)><PRE(4)><ETQ(2)><PRS(3)><FCB(1)><CBA(12)><TXT(max 25)>
<NTXT(2)>< TXT1(max 25)>… <TXTn(max 25)>
<NING(2)><ING1(4)>…<INGn(4)>
A:<RSP><ERR>

<COD>::=Item identifier (6 digits)
<SEC>::=Section or counter code (2 digits).
<PLU>::=Plu code. Item request code in a scale of given section.
<SUB>::=Not used.
<FAM>::=Item family.
<PRC>::=Sale price of item. The scale uses a maximum of 6 digits.
<WGH>::=Type of sale:
'W'=weighted item.
'U'=unweighted item.
<TXT>::=Description line.
<TAR>::=Associated tare.
<IVA>::=Type of VAT to apply.
<CAD>::=Expiry dates.
<PRE>::=Best before dates.
<ETQ>::= Item label number (0 to 9). Specials:
- 98: label by equipment default.
- 99: do not print label.
<PRS>::=Number of parcels pre-selected for labels.
<FCB>::=Bar-code format
'0': EAN13
<CBA>::=Bar code.
<NTXT>::=Number of lines. Value from 0 to 10 or 20 (depending on scale version).
<TXT>::=Maximum 25-character text. There are description lines <LNS>.
<NING>::=Number of ingredients. Value from de 0 to 15
<INGn>::= Ingredient code. There are <NING> ingredients.

There can only be description or ingredients.

<ERR>::=Error code.

## 5.10.38.3 <?prgc>::='W',87d,57h Request for complete-item information.

Can only be performed from the local network (order file) and for scales starting at programme version 9.

Consult programmed item data using any possible form of access. This message can be widened by the end.

Q:<art><?prgc><QUE(1)><MOD(1)><COD(6)><SEC(2)><PLU(4)><SUB(3)><FAM(4)>
<TXT(max 25)>
A:<art><COD(6)><SEC(2)><PLU(4)><SUB(3)><FAM(4)><PRC(7)><WGH(1)><TAR(5)><IVA(1)><CAD(4)><PRE(4)><ETQ(2)><PRS(3)><FCB(1)><CBA(13)><TXT(max 25)>
<NTXT(2)>< TXT1(max 25)>… <TXTn(max 25)>
<NING(2)><ING1(4)>…<INGn(4)>

A:<RSP><ERR>

<QUE>::=Indicates the data to search.
'0':Search same.
'1':Search more or same.
'2':Search more.
'3':Search less or same.
'4':Search less.
<MOD>::=Selection of access code
'0'- By Plu section.
'1'- By code.
'5'- By description. Nor used
'6'- By family/code

<COD>::=Item identifier (6 digits).
<SEC>::=Section or counter code (2 digits).
<PLU>::=Plu code. Item request code in a scale of given section.
<SUB>::=Not used.
<FAM>::=Item family.
<PRC>::=Sale price of item.
<WGH>::=Type of sale:
'W'=weighted item.
'U'=unweighted item.
<TXT>::=Descriptive line.
<TAR>::=Associated tare.
<IVA>::=Type of VAT to apply.
<CAD>::=Expiry dates.
<PRE>::=Best before dates.
<ETQ>::= Item label number (0 to 9). Specials:
- 98: label by equipment default.
- 99: do not print label.
<PRS>::=Number of parcels pre-selected for labels.
<FCB>::=Bar-code format
'0': EAN13
<CBA>::=Bar code.
<NTXT>::=Number of lines. Value from 0 to 10 or 20 (depending on scale version).
<TXT>::=Maximum 25-character text. There are descriptive lines <LNS>.
<NING>::=Number of ingredients. Value from de 0 to 15
<INGn>::= Ingredient code. There are <NING> ingredients.

There can only be descriptive or ingredients.

<ERR>::=Error code.

### 5.11 #<tot>::='8',56d,38h-TOTALS CONSULTING (NOT 386)

Messages that has to do with the accumulated totals request in the concentrator for each type of data.

#### 5.11.1 <fch>::='0',48d,30h.Query of time intervals in which there are totals.

Q:<tot><?fch>
A:<tot><DT1(8)><DT2(8)>
A:<RPS><ERR>.

For PC concentrator:
<DT1> ::=initial date, lowest found of any kind of total possible. In the programmed date format.
<DT2> ::=Iniital date, bigger found of any kind of total possible. In the programmed date format.
If there are no accumulated totals, it returns <TODAY><TODAY>.

#### 5.11.2 <?fch>::='1',49d,31h. Asing if there are totals in the indicated dates.

Prior consulting to know if there are totals in the required dates.

Q:<tot><fch><LNS(1)><DTE1(8)>.<DTE(8)>
A:<tot><LNS(1)><DT1(8)><ST1(1)>.<DTn(8)><STn(1)>
A:<RPS><ERR>.

<LNS> ::=Number of dates.
<Dtx> ::=Dates.
<Stx> ::=
'0': There are no accumulated totals of any kind in the date.
'1': There are totals of some sort on the date.

#### 5.11.3 <bor>::='2',50d,32h. Clearing and accumulating of great total.

Message for clearing and accumulate totals in the concentrator. There are two types of accumulated: absolute, which corresponds to a sales group that one wants to sum up together. It will usually be one day, but depending on the date indicated in the accumulated total clearing, it can correspond to the period of time desired.For example, to require accumulated by weeks, it is necessary to make an absolute total clearing once a week or to make a total clearing with the same date during the whole week. In the case of not wanting to keep the totals, the date must be sent with zeros.

The receipt numbering in the concentrator is sequential. It is only indicated after each receipt clearing.

Q:<tot><bor><TIP(1)><TIQ(1)><DTE(8)>
A:<RSP><ERR>

<TIP> ::=Type of total to be cleared.
'0': Absolute total or daily
For PC concentrator, clear the great total that will usually be daily and accumulate it with the indicated date. The totals cleared are accumulated for plus, sections, vendors, timetables and equipments.
'1'. Accumulated totals.
For PC concentrator, the accumulated totals are cleared every day.
<TIQ> ::= Clear receipts.
'0': Not cleared.
'1': All the receipts are cleared. The receipts are cleared and the numberation of them begin.

In the PC concentrator, the maximum possible number of receipts is 9.999. In the case of arriving at this number, the concentrator can no longer attend the scales.
<DTE> (Not used by 3081) the acumulation date of the totals when clearing the absolute big total and accumulate it in the accumulated Total. In the programmed date format. If it is all in zeros (the 8 digits), the totals are not kept, therefore, it is only possible to consult the daily totals.

**5.11.4 <sec>::='3',51d,33h.Total of equipment sales by sections, subsections and families.**

Consulting the sales totals in the interval of dates indicated of a sections, subsections and families.

The sections which are returned will be the ones of <SEC>,<SUB>,<FAM> more or the same as the requested. For PC, the identifier group <SEC><SUB><FAM> is always considered. For 8031 only <SEC>.

Q:<tot><sec><DT1(8)><DT2(8)><SEC(2)><SUB(3)><FAM(4)><TYP(1)><LNS(1)>
A:<tot><LNS(1)>
<SEC(2)><SUB(3)><FAM(4)><PES(9)><UND(9)><PTS(9)><OPS(8)><TQT(8)>.
<SEC(2)><SUB(3)><FAM(4)><PES(2)><PTS(9)><UND(9)><OPS(8)><TQT(8)>
A:<RSP><ERR>

<DT1> ::=Initial date. In the programmed date format.
PC concentrator:
If it is blank, it is the date of today.
8031 Concentrator:
If it is at zeros, it gives the great accumulated total.
If the date is the current date, it gives the great absolute total.
<DT2> ::=Inital date. (Not 8031)
PC Concentrador. If it is blank, it is the date of today
<SEC>
<SUB>
<FAM> ::=Section, subsection, identifier of totals that come later in the reply.
8031 concentrator: Doesn't have subsections nor families.
<LNS> ::=Number of lines of totals requested or that follow in the reply of the concentrator.
<TYP> ::= Type of requested total:
'0': Total in network of coded items.
For PC informs with: <OPS><PTS><TQT>.
For 8031 informs with: <OPS><PTS><PES>.
'1': Total in local of coded items. (Only PC).
For PC informs with: <OPS><PTS><TQT>.
'2': Total in direct sale to section in network, sales made in scales of the section with the PLU ZERO (introduction of direct price). (Only PC).
For PC informs with: <PTS><OPS>
'3': Total in direct sales to sections in local, sales made in the scales of the section with the PLU ZERO (introduction of direct price). (Only PC).
For PC informs with : <PTS><OPS>
'4': Totals in Self Service scales of the section. (Only PC)
For PC informs with : <PTS><TQT>
'5': Totals in labelling scales in sales of the section. (Only PC).
For PC informs with : <PTS><TQT>
'6':Totals in labelling train.
For PC informs with : <PTS><TQT>
'7': Totals in labelling scales in items packaging of the section. (Only PC).
For PC informs with : <PTS><TQT>
'8': Totals in direct operations in labellers in packaging of the section. (Only PC).

<PES> ::=Weight. Total in item sales by weight.
<UND> ::=Units. Total in sales of unitary items.
<OPS> ::=Operations.
Scales: Number of item lines of this section.
Labelling train : Number of parcels of items of this section.
<PTS> ::=Total amount due.
<TQT> ::=Number of receipts.
Scales: Number of Labels of scales. Calculated by sections.
UBI Labelling train: Number of series of items parcels of this section.
. CL Labelling Train: Value to zero.

The number of totals of the reply will not be higher than 4.

**5.11.5 <art>::='4',52d,34h. Total in item sales in equipments**

Consulting the sales totals between the two days of the consulting of an item group. The items returned will be those with value <IDA(6)>, bigger or the same as the requested one.

Q:<tot><art><DT1(8)><DT2(8)><MOD(1)><IDA(6)><TYP(1)><LNS(1)>
A:<tot><LNS(1)>
<IDA(6)><PES(9)><UND(9)><PTS(9)><OPS(8)> .
<IDA(6)><PES(9)><UND(9)><PTS(9)><OPS(8)>
<DT1> ::=Initial date.(Not 8031)
PC Concentrator: If it is blank, it is the current date.
<DT2> ::=Final date.(Not 8031)
PC concentrador: If it is blank, it is the current date.
If there are no totals in the requested date intervals, '0'will be returned to <LNS> of the reply.
<MOD> ::=Selection of password.
'0': <IDA> is interpreted as <SEC(2)><PLU(4)>.
'1': <IDA> is interpreted as <COD(6)>.
<TYP> ::=Type of total requested:
'0':Totals carried out in the networkwork scales.
'1':Totals carried out in the scales in local.
'2':Totals in Self Service scales.
'3':Totals in Labelling scales.
'4':Totals in Labelling Train.
'5':Totals in packaging.
<OPS> ::=Operations.
Scales: Number of sales receipt lines of this item.
Labelling train: Number of parcels of this item.
<PES>
<UND>::= Quantity sold. Will be units or weight depending on the form of sale. Only one of the fields is informed.
For the 8031: Information will be entered into both fields in case of requesting totals with the value <IDA>=<SS0000>. Therewith, the amount dues of direct sales are sent back to the section.
<PTS> ::=Total amount due sold.
<LNS> ::=Number of items with totals that come later. Will be '0' if there are no more totals or if there are no totals in the dates. The number of items will not be higher than 4.

**5.11.6 <vnd>::='5',53d,35h. Total sales vendors.**

Consults the totals carried out between the requested dates of a group of vendors. The vendors returned will be the <IDV> ones, more or the same as the requested.
Q:<tot><vnd><DT1(8)><DT2(8)><IDV(4)><TYP(1)><LNS(1)>
A:<tot><LNS(1)>
<IDV(4)><PES(9)><UND(9)><PTS(9)><OPS(8)><TQT(8)><DES(8)><REM(8)>.
<IDV(4)><PES(9)><UND(9)><PTS(9)><OPS(8)><TQT(8)><DES(8)><REM(8)>.
A:<RSP><ERR>
<DT1> ::=Initial date.(Not 8031)
PC concentrator: If it is blank. it is the current date.
<DT2> ::=Final date.(Not 8031)
PC concentrator: If it is blank. it is the current date
<IDV> ::=Identification of vendor.
8031 concentrador, [SSVV] (Sectión*100+Vendor) <IDV>::=<SS><VV>
PC concentrador, code of unique vendor.
<TYP> ::=Type of total:
'0':Totals in network
'1':Negative totals in network.
'2':Totals cancelled in network.
'3':Totals by credit. (Not PC)
'4':Totals in Labelling scales (Not 8031)
'5':Direct totals in receipts network. (Roll).
'6':Direct totals in Labels network (Labelling).
'7':Totals in packaging.
'8':Direct totals in packaging.

<LNS> ::=Number of total lines, vendors that follow.

<PES> ::=Weight. Total in item sales by weight.
<UND> ::=Units. Total in sales of unitary items.
<OPS> ::=Operations.
<PTS> ::=Amount due.
<TQT> ::=Receipts (Not used in 8031)
<DES> ::=Expenditures (Not PC)
<REM> ::=Refund (Not PC)

The number of vendors of the reply will not be higher to 3.

**5.11.7 <hor>::='6',54d,36h. Total timetables in scales.**

The total timetables carried out in network by section from the time indicated in the period of the dates requested.

Q:<tot><hor><DT1(8)><DT2(8)><SEC(2)><TME(8)><LNS(1)>
A:<tot><SEC(2)><LNS(1)>
<TME(8)><PES(9)><UND(9)><PTS(9)><OPS(8)<TQS(8)>.
<TME(8)><PES(9)><UND(9)><PTS(9)><OPS(8)<TQS(8)>
A:<RSP><ERR>

The value of a field will be in zeros '0' if the concentrator doesn't calculate the indicated values.

<DT1> ::=Initial date.(Not 8031)
PC concentrator: If it is blank. it is the current date.
<DT2> ::=Final date.(Not 8031)
PC concentrator: If it is blank. it is the current date
<SEC> ::=Sections. '00' indicates that the totals are global.
<LNS> ::=Number of totals sent.
<TME> ::=Inital time HHMM. The resolution of time depends on the concentrator.Will be rounded at the most closer time to this resolution.The temporary increasing also depends on the concentrator resolution.
<PES> ::=Total weight. Sale of item by weight. (Not PC).
<UND> ::=Total units. Sales of unitary items.(Not PC).
<OPS> ::=Operations.
<TQS> ::=Receipts carried out.
<PTS> ::=Amount due.

**5.11.8 <bal>::='7',55d,37h. General total of equipments. (Not 8031).**

Request of the totals carried out in local in the equipments.

Q:<tot><locb><DT1(8)><DT2(8)><NUM(2)><TYP(1)><LNS(1)>
A:<tot><LNS(1)>
<NUM(2)><PTS(9)><OPS(8)><PTD(9)><OPD(8)><TIQ(8)>.
<NUM(2)><PTS(9)><OPS(8)><PTD(9)><OPD(8)><TIQ(8)
A:<RPS><ERR>

<DT1> ::=Initial date.(Not 8031)
PC concentrator: If it is blank. it is the current date.
<DT2> ::=Final date.(Not 8031)
PC concentrator: If it is blank. it is the current date
<NUM> ::=Equipment number.
<LNS> ::=Number of requested equipments.
<TYP> ::=Type of total requested.:
'0':Totals in network.
'1':Totals in local.
'3':Totals in packaging.
<OPS> ::=Operations in items sales.
Labelling train: Parcels done in the equipment.
<PTS> ::=Amount due of items.
<PTD> ::=Direct sales.

Scales: Amount due in direct sales to section, with the PLU ZERO.
Labelling train: Items not coded in the concentrator.

<OPD> ::= Direct operations.
Scales: Number of operations of direct sales to section.
Labelling train: Parcels of non-coded items in the concentrator.

<TIQ> ::= Receipts.
Scales: Number or receipts requested in total in each scale.
. UBI Labelling Train: Number of parcels series of each labelling train.
CL Labelling train: Number or totals (parcels) formed by different items.

The number ot totals of the reply will not be higher to 4.

#### 5.11.9 <dia>::='8',56d,38h. General total of the concentrator (Not 8031).

Requesting the daily totals of each type. The types are the same that sections.

Q:<tot><dia><DT1(8)><DT2(8)><TYP(1)><LNS(1)>
A:<tot><LNS(1)>
<DTE(8)><PTS(9)><UND(9)><OPS(8)><TIQ(8)>.
<DTE(8)><PTS(9)><UND(9)><OPS(8)><TIQ(8)>.

<DT1> ::=Initial date.(Not 8031)
PC concentrator: If it is blank. it is the current date.

<DT2> ::=Final date.(Not 8031)
PC concentrator: If it is blank. it is the current date

The number of totals of the reply will not be higher than 4.

#### 5.11.10 '9',57d,39h. Programming of type of totals (Not 8031).

Programming the type of totals to be consulted. It is not necessary to programme each time a query is done.

Q:<tot><ttot><TIP(1)>
A:<RSP><ERR>

<TIP> ::= Type of total to be consulted:
0: Consulting all the data (sections, items , vendors) with or without totals.
1: Only consulting data if they have totals.

#### 5.11.11 <?ttot>::=';',58d,3Ah. Consulting type of totals (Not 8031).

Consults type of totals.

Q:<tot><ttot>
A:<tot><TIP(1)>

<TIP> ::= Type of total to be consulted:
0: Consulting all the data (sections, items , vendors) with or without totals.
1: Only consulting data if they have totals.

#### 5.11.12 <tiva>::=';',59d,3bh. VAT totals (Not 8031).

Requesting the daily totals of each type of VAT.

Q:<tot><iva><DTE(8)><DT2(8)>
A:<tot><DTE(8)><IMP0(9)><IVA0(9)><IMP1(9)><IVA1(9)><IMP2(9)><IVA2(9)>
<IMP3(9)><IVA3(9)><IMP4(9)><IVA4(9)><IMP5(9)><IVA5(9)>

<DTE> ::=Date of the query.
PC concentrador: If it is left in blank, it is the date of today.
<IMPx>::= Taxable of each type of VAT.
<IVAx>::=Total of each type of VAT of the programmed ones.
<IMP0>::= Total without VAT.

**5.11.13 <tot?>::='<',60d,3ch. Totals saved in SC10 concentrator (Not 8031).**

Q:<tot><tot?>
A:<tot><SEC(1)><PLU(1)><OPE(1)><HOR(1)><EQU(1)><DIA(1)><IVA(1)>

<SEC> ::=Totals of section.
'0': No
'1': Yes
<PLU> ::= Item totals.
<OPE> ::= Vendor totals.
<HOR> ::= Daily sale totals.
<EQU> ::=Totals by equipment.
<DIA> ::= Totals of the day.
<IVA> ::=VAT totals.

**5.11.14 <hpc?>:='=',61d,3dh. Consulting parameters of automatic zero setting (Not 8031).**

Q:<tot><hpc?>
A:<tot><TME(8)><TQT(1)><ACT(1)>

<TME> ::=Time of the automatic zero setting, '99:99', zero setting is not carried out.
<TQT> ::=
'0': Clear the receipts when the automatic zero setting is carried out.
'1' : Don't clear the receipts.
<ACM> ::=
'0' : Accumulate the data when the automatic zero setting is carried out with the current date.

'1' : Don't accumulate totals.

**5.11.15 <hpc>:='>',62d,3eh. Programming parameters of automatic zero settings (Not 8031).**

Q:<tot><hpc><TME(8)><TQT(1)><ACT(1)>
A:<RSP><ERR>

<TME> ::=Time of the automatic zero setting, '99:99', the automatic zero setting is not carried out.
<TQT> ::=
'0': Clear the receipts when the automatic zero setting is carried out.
'1' : Don't clear the receipts.
<ACM> ::=
'0' : Accumulate the data when the automatic zero setting is carried out with the current date.

'1' : Don't accumulate totals.

**5.12 # <pub>::='9',57d,39h-ADVERTISING**

**5.12.1 <pbc>::='0',48d, 30h. Programming of continuous advertising of a scale or section**

Q:<pub><pbc><MOD(1)><DIR(2)><CHR(3)><PBC(100)>
A:<RPS><ERR>
<MOD> ::=Indicates if the programming is for scale or for section.
'0': Section.
'1': Scale.(Not 8031)
<DIR> ::=Number of scale or section/counter.
<CHR> ::=Number of characters.
<PCB> ::=Text of continuous advertising.
<ERR> ::=Error code.

**5.12.2 <pbo>::='1',49d,31h.Programming of advertising offers of a scale.**

Q:<pub><pbo><MOD(1)><DIR(2)><NOF(2)><OFF1(16)>.<OFF10(16)>
A:<RPS><ERR>

<MOD> ::=Indicates of the programming is for scale or for section.
'0': Section.
'1': Scale.(Not 8031)
<DIR> ::=Number or scale or section/counter.
<NOF> ::=Number of lines of offer.
<OFFx>::= Lines of advertising of offers, maximum 10.
<ERR> ::=Error code.

**5.12.3 <pbb>::='2',50d,32h.Activation/Desactivation of the scale or section advertising.**

Q:<bub><pbb><MOD(1)><DIR(2)><STS>
A:<RPS><ERR>

<MOD> ::=Indicates if the programming is for scale or section.
'0': Section.
'1': Scale (Not 8031)
<DIR> ::=Number of scale or section.
<STS> ::=State of the advertising.
'0' (30h): Advertising desactivated.
'1' (31h): Advertising CONTINUOUS activated.
'2' (32h): Advertising OFFERS activated.
<ERR> :=Error code.

**5.12.4 <?pbc>::='3',51d,33h.Requesting continuous advertising of a scale or section.**

Q:<pub><?pbc><MOD(1)><DIR(2)>
A:<pub><MOD(1)><DIR(2)><CHR(3)><PBC(100)>

<MOD> ::=Indicates of the programming is for scale or for section.
'0': Section (Not 386).
'1': Scale.(Not 8031)
<DIR> ::=Number or scale or section/counter.
<NOF> ::=Number of lines of offer.
<OFFx>::= Lines of advertising of offers, maximum 10.
<ERR> ::=Error code.

**5.12.5 <?pbo>::='4',52d,34h.Requesting advertising offers of a scale or section.**

Q:<pub><?pbc><MOD(1)><DIR(2)>
A:<pub><MOD(1)><DIR(2)><NOF(2)><OFF1(16)>.<OFF10(16)>

<MOD> ::=Indicates of the programming is for scale or for section.
'0': Section (Not 386).
'1': Scale.(Not 8031)
<DIR> ::=Number or scale or section/counter.
<NOF> ::=Number of lines of offer.
<OFFx>::= Lines of advertising of offers, maximum 10.
<ERR> ::=Error code.

**5.13 #<cbl>::=':',58d,3ah. HEADINGS, LEGENDS AND FREE TEXTS.**

**5.13.1 <cab>::='0',48d,30h.Programming receipt texts of a scale or section.**

Q:<cbl><cab><MOD(1)><DIR(2)><TYP(1)><LNS(1)><TLT(1)><TXT(Max25)>..<TLT(1)><TXT(Max25)>
A:<RPS><ERR>
In 386, they are sent in groups of 5.

<MOD> ::=Indicates if the programming is for scale or for section/counter.
'0': Section.
'1': Scale.(Not 8031)

<DIR> ::=Number or scale or section.
<TYP> ::=Type of advertising to be programmed.
'0': Headings.
'1': Legends.
'2': Lines of free text (first 5) (Only 386)
'3': Lines of free text (last 5) (Only386)
<LNS> ::=Number of heading lines that are sent.
<TLT> ::=Type of letter of a line.
'0': Normal letter.
'1': Double hight letter.
'2': Double width letter.
'3': Double hight and double width letter.

For 386, values: 48d.127d

<TXT> ::=Text that composes the line. (Maximum 5 lines)
<ERR> ::=Error code.

**5.13.2 <?cab>::='1',49d,31h.Requesting receipt texts of a scale or section.**

Q:<cbl><?cab><MOD(1)><DIR(2)><TYP(1)>
A:<cbl><MOD(1)><DIR(2)><TYP(1)><LNS(1)><TLT(1)><TXT(Max25)>.
.<TLT(1)><TXT(Max25)>
A:<RPS><ERR>
In 386, they are sent in groups of 5.

<MOD> ::=Indicates if the programming is for scale or for section.
'0': Section. (Not 386)
'1': Scale.(Not 8031)
<DIR> ::=Number of scale or section.
<TYP> ::=Type of advertising to be programmed.
'0': Headings.
'1': Legends.
'2': Lines of free text (first 5) (Only 386)
'3': Lines of free text (last 5) (Only386)

<LNS> ::=Number of heading lines that are sent.
<TLT> ::=Type of letter of a line.
'0': Normal letter.
'1': Double hight letter.
'2': Double width letter.
'3': Double hight and double width letter.

For 386, values: 48d.127d

<TXT> ::=Text that composes the line. (Maximum 5 lines)
<ERR> ::=Error code.

**5.14 #<tot>::='8',56d,38h- TOTALS (ONLY 386)**

**5.14.1 <bor>::='2',50d,32h. Clearing of totals receipts (Zero setting) .**

The receipts numbering is sequential: It is only iniciated after the clearing request with inizializing of receipts. In the case that the receipt number is not inizialized, the receipts are cleared and the numerating goes on.

The accumulated totals are only by counter or system.

Q:<tot><bor><TIP(1)><TIQ(1)><DTE(8)>
A:<RSP><ERR>
<TIP>::='0'-Clearing of all the totals and daily receipts.
'1'- Clearing of all teh totals and accumulated receipts.
'2'- Clearing receipts. Not implemented.
'3'- Clearing of daily totals of the counter. Not implemented.
'4'- Clearing of accumulated totals of the counter. Not implemented.
'5'- Clearing of item totals. Not implemented.
'6'- Clearing of vendors totals. Not implemented.
'7'- Clearing of timetable totals. Not implemented.
'8'- Clearing of daily totals of the system. Not implemented.
'9'- Clearing of accumulated totals of the system. Not implemented.
<DTE>::= Not used.
<TIQ>::= Inicializing receipt number (if receipt clearing).
'0': No.
'1':Yes.

**5.14.2 <sec>::='3',51d,33h.Total sales by section.**

Consulting the sales totals in the date interval indicated in a section.

In the current scale system there is only the distincion of section (counter).
There are only two types of totals for counter: daily and accumulated.
Q:<tot><sec><DTE1(8)><DTE2(8)><SEC(2)><SUB(3)><FAM(4)><TYP(1)><LNS(1)>
A:<tot><LNS(1)><SEC(2)><SUB(3)><FAM(4)><PES(9)><DTO(9)><PTS(9)><OPS(8)><TQT(8)>
A:<RSP><ERR>
<DTE1>::=Initial date. In the programmed date format.If it is in blank, it is the current date.
If it is '999999' it is the accumulated.
<DTE2>::=Final date. If it is in blank, it is the current date. NOT USED.
<SEC><SUB><FAM>::=Section, subsection, initial family. <SEC> only used.
<LNS>::=Fixed value '1'.
<TYP>::=Type of requested total:
'0': Total of scale sales in network.
'1':Total of scale sales in local.
'2':Total of direct sales to scales section in network. Not implemented.
'3': Total of direct sales to scales section in local. Not implemented.
'4':Not used.
'5':Not used.
'6':Totals in labelling train. Not implemented.
'7':Totals of packaging.
'8':Total of direct packaging to scales section in network. Not implemented.
'9':Totals of packaging in local.
':':Total of direct packaging to scales section in network. Not implemented.
<PES>::=Weight. Total in item sales by weight. Not used.
<DTO>::= Total discount for matching items.
<OPS>::=Operations.
Scales: Number of items lines of this section.
Labelling train : Number of items parcels of this section.
<PTS>::=Total amount due.
<TQT>::=Number of receipts.

Remark: the information from the direct sales to section is only given if the access is by PLU number.

**5.14.3 <art>::='4',52d,34h. Totals item sales**

Consulting of sales totals between the days of the item consulting.
In the current scales system, there is only the daily item total (only the ones of one day are saved).

Q:<tot><art><DTE1(8)><DTE2(8)><MOD(1)><IDA(6)><TYP(1)><LNS(1)>
A:<tot><LNS(1)><IDA(6)><PES(9)><UND(9)><PTS(9)><OPS(8)>
A:<RSP><ERR>
<DTE1>::=Initial date. If it is in blank, it is the current date. Not used.
<DTE2>::=Final date. If it is in blank, it is the current date. Not used
<MOD>::=Selection of access key.
'0'- <IDA> is interpreted as <SEC(2)><PLU(4)>. Not implemented.
'1'- <IDA> is interpreted as <COD(6)>
<TYP>::=Type of total requested:
'0':Sales totals carried out in the network scales.
. '1':Sales totals carried out in the scales being in local.
'2':Not used.
'3':Not used.
'4':Totals in labelling train. Not implemented.
'5':Totals of packaging carried out in network.
'6':Totals of packaging carried out in local.
<OPS>::=Operations.
Scales: number of sales receipt lines of this item.
Labelling train: Number of pracels of this item.
<PES> :: = Kilos sent.
<UND>::= Sales unit.
<PTS>::=Total amount due sold.
<LNS>::=Fixed value '1'.

**5.14.4 <vnd>::='5',53d,35h. Total vendors.**

Consulting of the totals carried out by a vendor between the requested dates.
The current scale system has only totals of the current day.

Q:<tot><vnd><DTE1(8)><DTE2(8)><IDV(4)><TYP(1)><LNS(1)>
A:<tot><LNS(1)><IDV(4)><PES(9)><UND(9)><PTS(9)><OPS(8)><TQT(8)><DESM(8)><REEM(8)>
A:<RSP><ERR>
<DTE1>::=Inital date. If it is blank,it is the current date. Not used.
<DTE2>::=Final date. If it is blank, it is the current date. Not used.
<IDV>::=Vendor identification.
If the scales programming is automatic [SSVV] (Sectión*1000+Key) <IDV>::=<SS><VV>
<TYP>::=Type of total:
'0':S Network sales totals. ':':Totals of scale in local.
'1':Totals of negative sales in network. ';':. Totals of negative sales in local.
'2':Totals of sale cancelled in network. '<': Totals of sale cancelled in local.
'3':Not used. '=': Not used
'4': Not used '>': Not used
'5': Not used '?': Not used
'6': Not used '@':Not used.
'7':Totals of packaging in network. 'A: Totals of packaging in local.
<LNS>::=Fixed value '1'.
<PES>::=Weight. Total in sales of items by weight. Not used.
<UND>::=Units. Total in sales of unitary item. Not used.
<OPS>::=Operations.
<PTS>::=Amount due.
<TQT>::=Receipts, to zero in negative and cancelled totals.
<DESM>::=Expenditure. Not used.
<REEM>::=Refund. Not used.

**5.14.5 <hor>::='6',54d,36h. Total timetable.**

Total time spent on sales in network and by section within the period specified.
The current scale system has only daily totals (of one day).
They are not kept separately by the counter (section).

Q:<tot><hor><DTE1(8)><DTE2(8)><SEC(2)><TME(8)><LNS(1)>
A:<tot><SEC(2)><LNS(1)><TME(8)><PES(9)><UND(9)><PTS(9)><OPS(8)<TQS(8)>
A:<RSP><ERR>
<DTE1>::=Inital date. If it is left in blank, it today's date. Not used.
<DTE2>::=Final date. If it is left in blank, it is today's date. Not used.
<SEC>::=Section. '00' indicates that the totals are global.
<LNS>::=Fixed value '1'.
<TME>::=Inital time HHMM, justified to the left.
<PES>::=Total weight. Sales of items by weight. Not used.
<UND>::=Total units. Sales of unitary items. Not used.
<OPS>::=Operations.
<TQS>::=Receipts carried out.
<PTS>::=Amount due.

**5.14.6 <dia>::='8',56d,38h. Global total of the equipment's system.**

Request the daily totals of each type.
Has 10 days stored separately and also an accumulated from the last clearing of accumulated total.

Q:<tot><dia><DTE1(8)><DTE2(8)><TYP(1)><LNS(1)>
A:<tot><LNS(1)><DTE(8)> <PES(9)><DTO(9)><PTS(9)><OPS(8)><TQT(8)>
<DTE1>::=Inital date. If it is in blank, it is today's date. Not used
If it is accumulated, it is '999999'
<DTE2>::=Final date. If it is in blank, it is today's date. Not used
<LNS>::=Fixed value '1'.
<TYP>::=Type of total requested:
'0': Total of scales sales in network.
'1': Total of scales sales in local.
'2': Total of direct sales to scales section in network. Not implemented.
'3': Total of direct sale to scales section in local. Not implemented.
'4':Not used.
'5':Not used.
'6':Totals in labelling train. Not implemented.
'7':Packaging totals in network.
'8':Total of direct packaging to scales section in network. Not implemented.
'9':Packaging totals in local.
':':Total of direct packaging to scales section in local. Not implemented.
<PES>::=Weight. Not used.
<DTO>::=Total discount for matching items.
<OPS>::=Operations
Scales: Number of item lines.
Labelling train: Number of items parcels.
<PTS>::=Total amount due.
<TQT>::=Number of receipts.
<DTE>::= If accumulated, date of the last setting zero. Not implemented.

Remark: The direct sales to section are only informed if the access is by codes.

**5.14.7 <tiva>::=';',59d,3bh. VAT total**

Requesting the daily totals of each type of VAT summed up network and local in sale.
Q:<tot><iva><DTE(8)><DTE2(8)>
A:<tot><DTE(8)><IMP0(9)><IVA0(9)><IMP1(9)><IVA1(9)><IMP2(9)><IVA2(9)>
<IMP3(9)><IVA3(9)><IMP4(9)><IVA4(9)><IMP5(9)><IVA5(9)>

<DTE1>::= Initial date. If it is left in blank, it is the current date.
If it is accumulated, '999999'
<DTE2>::=Final date. If it is left in blank, it is the current date.
<IMPx>::= Taxable of each VAT type. (IMP5 in zero).
<IVAx>::=Total of each type of VAT of the programmed ones. (IVA5 to zero)

#### 5.14.8 <art>::='?',63d,3Fh. Total item range sales

Consulting sales totals between the two days of the consulting of a family.
In the current scales system there is only the daily totals of item range (keeps only of one day).

Q:<tot><art><DTE1(8)><DTE2(8)><FAM(4)><QUE(1)><TYP(1)><LNS(1)>
A:<tot><LNS(1)><FAM(4)><PES(9)><UND(9)><PTS(9)><OPS(8)>
A:<RSP><ERR>
<DTE1>::=Inital date. If it is left in blank, it is the current date. Not used.
<DTE2>::=Final date. If it is left in blank, it is the current date. Not used.
<FAM>::=Family to be consulted.
<QUE>::=Indicates that data to be searched.
'0':Search the same.
'1':Search more or the same.
'2':Search more.
'3':Searche less or the same.
'4':Search less.
<TYP>::=Type of total requested:
'0':Sales totals carried out in the network scales.
'1': Sales totals carried out in the local scales.
'2':Not used.
'3':Not used.
'4':Totals in labelling train. Not implemented.
'5':Packaging totals carried out in network.
'6':Packaging totals carried out in local.
<OPS>::=Operations.
Scales: number of receipt lines of the items designated to the family.
Labelling train: Number of parcels.
<PES> :: = Kilos sold.
<UND>::= Units sold.
<PTS>::=Total amount due sold.
<LNS>::=Fixed value '1'.

#### 5.14.9 <zart>::='@',64d,40h. Total different sales of items zero.

Consulting an item's sales different to zero, between the two days specified.

In the current scales system there is only a daily total of items (only the ones of one day are saved).
**IMPORTANT**: The scale sends code 0 last. The item search with <IDA>=0 as biggest value (<QUE>='2') works the same as the bigger or same search (<QUE>='1'):

- if there are items programmed with totals, the smallest code will be sent.
- If there are no items with totals, the total of code 0 will be sent (if it has totals).
- If there are no totals on any item, 'end of consultation' will be sent.

Therefore, when code 0 is received, no further consultations should be made.

Q:<tot><art><DTE1(8)><DTE2(8)><QUE(1)><NUM(2)><MOD(1)><IDA(6)><TYP(1)><LNS(1)>
A:<tot><LNS(1)><IDA(6)><PES(9)><UND(9)><PTS(9)><OPS(8)>
A:<RSP><ERR>
<DTE1>::= Inital date. If it is left in blank, it is the current date. Not used
<DTE2>::= Final date. If it is left in blank, it is the current date. Not used
<QUE>::=Indicates that data to be searched.
'0':Search the same.Not implemented.
'1':Search more or the same.
'2':Search more.
'3':Searche less or the same.Not implemented.

'4':Search less. Not implemented.
<NUM>::= Scale number in which to search the total. Not implemented.
<MOD>::=Selection of access key.
'0'- <IDA> is interpreted as <SEC(2)><PLU(4)>. Not implemented.
'1'- <IDA> is interpreted as <COD(6)>.
<TYP>::=Type of total requested:
'0':Totals of sale carried out in the network scales.
'1': Totals of sale carried out in the local scales
'2':Not used.
'3':Not used.
'4':Totals in labelling train. Not implemented.
'5':Totals of packaging carried out in network.
'6':Totals of packaging carried out in local.
<OPS>::=Operations.
Scales: Number of sales receipt lines of the item assigned to the family.
Labelling train: Number of parcels of this item.
<PES> :: = Kilos sold.
<UND>::= Units sold.
<PTS>::= Total amount due sold.
<LNS>::=Fixed value '1'.

### 5.15 # <tqt>::=';',59d,3bh – RECEIPTS CONSULTING AND PARCELS SERIES

Messages relative to the consulting and acquisition of receipts and receipt lines and parcels series of labelling train.

#### 5.15.1 <stm>::='0',48d,30h.State of receipts memory (Not PC) (Not 386)

Q:<tqt><stm>
A:<tqt><TQL(5)><HOR(8)><TQN(8)>
<TQL> ::=Memory space for free receipts.
<HOR> ::= Time when receipts overflow took place.
<TQN> ::=Number of non-stored receipts due to the lack of capacity.

#### 5.15.2 <cab>::='1',49d,31h. Requesting receipts headings (Not 8031) (Not 386)

Q:<tqt><cab><QUE(1)><MOD(1)><TQT(5)>
A:<tqt><TQT(5)><SEC(2)><IDV(4)><BAL(2)><NLI(2)><NC(2)><PTS(7)><HOR(8)><STS(1)><PED(1)><CLI(6)>
<QUE> ::=Indicates the data to be searched.
'0': Search the same.
'1': Search more or the same.
'2': Search more.
'3': Search less or the same.
'4': Serach less.
<MOD> ::=Type of query:
'0'(30h): Absolute receipts. The receipt of <TQT> number is requested.
'1'(31h): Non-requested receipts. First receipt non-requested before.
'2'(32h): Receipts non requested and closed.
<TQT> ::=Receipt number.
<SEC> ::=Section.
<IDV> ::=Vendor identifier.
<CLI> ::Customer code for a *their turn* configuration.
<BAL> ::=Equipment where the sale was closed.
<NLI> ::=Number of lines that the receipt has.
<NC> ::=Number of cancelled lines.
<PTS> ::=Receipt amount due.
<HOR> ::= Time of receipt's closing. If it is not closed, in blank.
<STS> ::=State of receipt:
'0': Closed, already finished.
'1': Opened, lines are beig inserted.
'2': In total, the total has been requested, but not the clearing of the total.

'3': In reload. The receipt has been closed but has been opened again.
'4': By order and closed
'5': By order and reopened.

<PED> ::=State of receipt request.
'0': Never requested before.
'1': Order requested before.

**5.15.3 2',50d,32h.Sequential request for receipt lines (Not PC) (Not 386)**

Q:<tqt><sln><MOD(1)>
A:<tqt><HOR(8)><SEC(2)><IDV(3)><STK(1)><TQT(5)><LNT(2)><PLU(4)><STW(1)><PES(9)><PTS(9)>

<MOD> ::=Mode of sequential request
'0'(30h): 1st line of receipts
'1'(31h): Next line of receipts
'2'(32h): Repetition of last line sent.
'3'(33h): Sending of receipts memory.
'4'(34h): Clearing read receipts.
'5'(35h): Clearing of all the receipts.
'6'(36h): Sending lines to printer.

<HOR> ::=Time of receipt closing.
<SEC> ::=Receipt section.
<IDV> ::=Identifier of vendor.
<STK> ::=Status continuation.
'0': No continuation.
'1': Continuation.
'2': Closing from host or master scale.
<TQT> ::=Receipt number.
<LNT> ::= Number of receipt line. It is zero if it is the last line.
<PLU> ::= Item sold in the line.
<STW> ::= Sale status.
'W' : Weight.
'U': Units.
<PES> ::= Quantity sold in the line.
<PTS> ::= Amount due sold in the line.

**5.15.4 <lin>::='3',51d, 33h.Requesting receipt line (Not 8031) (Not 386)**

Q:<tqt><lin><TQT(5)><NLIN(2)><LIN(1)>
A:<tqt><TQT(8)><LIN(1)>
<NLIN(2)><IDA(6)><PRC(7)><PES(9)><STV(1)><STP(1)><STC(1)>..
<NLIN(2)><IDA(6)><PRC(7)><PES(9)><STV(1)><STP(1)><STC(1)>..

Requesting receipt lines.
<IDA> ::=In <SSPPPP> format.
<TQT> ::=Receipt number.
<NLIN> ::=Number of line as from the one requested.
<LIN> ::=Number of requested receipts.
<PRC> ::=Sales price.
<PES> ::=Quantity sold.
<STV> ::=Sale status.
'0': Add.
'1': Subtracts.
<STP> ::=Weight status.
'0': Weighed.
'1': Not weighed.
<STC> ::=Cancellation status. If cancelled, it doesn`t add to, nor subtract from the total.
'0': Not cancelled.
'1': Line cancelled, (it doesn`t add to, nor substract from the total).

**5.15.5 <etq>::='4',52d, 34h.Requesting label (Not 8031) (Not 386)**

Q:<tqt><etq><TIP(1)><QUE(1)><MOD(1)><ETQ(5)>
A:<tqt><ETQ(5)><SEC(2)><IDV(4)><BAL(2)><PTS(7)><HOR(8)><PED(1)><CLI(6><IDA(6)><PRC(7)><PES(9)><STV(1)><STP(1)>

Request of labels from the labeller or self-service.

<TIP> := Type of label to be searched.
'0': Labeller label.
'1': Self-service label.

<QUE> ::=Indicates the data to be searched.
'0': Search the same.
'1': Search more or the same.
'2': Search more.
'3': Search less or the same.
'4': Search less.

<MOD> ::=Type of request:
'0'(30h): Absolute labels. The label of number <ETQ> is requested.
'1'(31h): Non-requested labels. First label that has never been requested.

<ETQ> ::=Label number.
<SEC> ::=Section.
<IDV> ::=Identifier of vendor.
<BAL> ::=Scale where the sale was closed.
<PTS> ::=Label amount due.
<HOR> ::=Time of label closing in case it is. If not, left in blank.
<PED> ::=State of label request.
'0': Never requested before.
'1': Order requested before.
<CLI> ::Customer's code, for a turn-number configuration.
<IDA> ::= In <SSPPPP> format.
<PRC> ::=Sale price.
<PES> ::=Quantity sold.
<STV> ::=Sales status.
'0': Total.
'1': Subtraction.
<STP> ::=Weight status.
'0': Weighed.
'1': Not weighed.

**5.15.6 <lot>::='5',53d, 35h.Requesting of UBI labelling train series (Not 8031) (Not 386)**

Q:<tqt><lot><QUE(1)><MOD(1)><EQU(2)><LOT(5)>
A:<tqt><EQU(2)><LOT(5)><IDA(6)><CLI(8)><DTE(8)><HOR(8)><PRC(7)><PTS(9)><PES(9)><OPS(8)><STP(1)><PED(1)>

Labelling train series request.

<QUE> ::=Indicates the data to be searched.
'0': Search the same.
'1': Search more or the same.
'2': Search more.
'3': Search less or the same.
'4': Search less

<MOD> ::=Type of request:
'0'(30h): Absolute series. The serie of <LOT> number is requested.
'1'(31h): Non requested series. First serie that has never been requested.

<LOT> ::=Serial number.
<EQU> ::=Equipment .
<PTS> ::=Serial amount due.
<DTE> ::=Date of end of series.
<HOR> ::=Time of end or series.
<PED> ::=State of request of series.
'0': Never requested before.
'1': Order requested before.
<CLI> ::Customer's code.

<IDA> ::=Internal item code.
<PRC> ::=Item's price in the series.
<PES> ::=Quantity of item in the series.
<OPS> ::=Number of operations of the series.
<STP> ::=Weight status.
'0':Weighed.
'1':Non weighed.

**5.15.7 <ca1>::='6',54d,36h. Requesting receipt heading per sec/ven/bal/fecha(Not 8031) (Not 386)**

Q:<tqt><ca1><QUE(1)><MOD(1)><TQT(5)><ITM(8)>
A:<tqt><TQT(5)><ITM(8)><SEC(2)><IDV(4)><BAL(2)><NLI(2)><NC(2)><PTS(7)><HOR(8)><STS(1)><PED(1)><CLI(6)><DTE(8)>
<QUE> ::=Indicates the data to be searched.
'0': Search the same.
'1': Search more or the same.
'2': Search more.
'3': Search less or the same.
'4': Search less
If <MOD>=='7'
'0' : First.
'1','2' : Next.
'3','4' : Previous.
<MOD> ::=Type of request
'0'(30h): Consulting according to numbering by default of the concentrator.
'1'(31h): Non-requested receipts. First receipt never requested before.
'2'(32h): Non-requested and closed receipts.
'3'(33h): Receipts by section
'4'(34h): Receipts by scale.
'5'(35h): Receipts by date.
'6'(36h): Receipts by vendor.
'7'(37h): Sequential consulting, without considering neither ITM nor type of scale, in the following order: RECEIPTS of receipt-roll scale, LABELS of labelling scale, LABELS of Self-service scale, and LABELS in local by scales. For receipt in local if it is used in the ITM field (number or scale).
'8'(38h): Programmable.
'9'(39h): Non-requested and closed receipts (don't mark).
<TQT> ::=Receipt number, label, or self-service according to requested numbering, in case of mode '7' returns to the sequential number in the concentrator. From this number onwards it is possible to consult the alternative according to the concentrator.

<ITM> ::=Date to be consulted (section, vendor, etc.)
<SEC> ::=Section.
<IDV> ::=Identifier of vendor.
<CLI> ::=Customer code, for a *turn-number* configuration.
<BAL> ::=Scale where the sale was closed.
<NLI> ::=Number of lines that the receipt has.
<NC> ::=Number of cancelled lines.
<PTS> ::=Amount due of receipt.
<HOR> ::=Time of closing of the receipt, if closed, if not left in blank.
<STS> ::=State of receipt.
'0':Closed , it has been concluded.
'1':Open, lines are being inserted.
'2':In total, the total has been requested, but not the total clearing.
'3':In reload. The receipt has been closed but reopened.
<PED> ::=State of receipt request.
'0': Never requested before.
'1': Order requested before.
<DTE> ::=Date of receipt.

**5.15.8 <li1>::='7',55d,37h. Requesting receipt lines as per sec/ven/bal/dat (Not 8031) (Not 386)**

Q:<tqt><li1><TQT(5)><NLIN(2)><LIN(1)><MOD(1)><ITM(8)>
A:<tqt><TQT(5)><LIN(1)><ITM(8)>
<NLIN(2)><IDA(6)><PRC(7)><PES(9)><STV(1)><STP(1)><STC(1)>..
<NLIN(2)><IDA(6)><PRC(7)><PES(9)/><STV(1)><STP(1)><STC(1)>..

Request of receipt lines.

<IDA> ::= In <SSPPPP>format.
<TQT> ::=Receipt number.
<MOD> ::=Type of request:
'0'(30h): Consulting according to numbering by default of the concentrator.
'3'(33h): Receipts by section.
'4'(34h): Receipts by scale.
'5'(35h): Receipts by date.
'6'(36h): Receipts by vendor.
'7'(37h): Sequential consulting, without considering neither ITM nor type of scale, in the following order: RECEIPTS of receipt-roll scale, LABELS of labelling scale, RECEIPTS of Self-service scale, and RECEIPTS in local by scales. For receipt in local if it is used in the ITM field (number of scales).
'8'(38h): Programmable receipts format
'9'(39h): Non requested and closed receipt (are not marked)
<ITM> ::=Data to be consulted (section, vendor, etc.)
<NLIN> ::=Line number following the one requested.
<LIN> ::=Number of requested lines.
<PRC> ::=Sales price.
<PES> ::=Sold quantity.
<STV> ::=Sales status.
'0': Add.
'1': Subtract.
<STP> ::=Weight status.
'0': Weighed.
'1': Not weighed.
<STC> ::=Status of cancelling; if cancelled it neither adds to, nor subtracts from the total.
'0': Not cancelled.
'1': Line cancelled, (neither adds to, nor subtracts from the total).

**5.15.9 <et1>::='8',56d,38h. Requesting label as per sec/ven/bal/dat (Not 8031) (Not 386)**

Q:<tqt><etq><TIP(1)><QUE(1)><MOD(1)><ETQ(5)><ITM(8)>
A:<tqt><ETQ(5)><ITM(8)><SEC(2)><IDV(4)><BAL(2)><PTS(7)><HOR(8)><PED(1)>
<CLI(6)><IDA(6)><PRC(7)><PES(9)><STV(1)><STP(1)><DTE(8)>

Request for Label from the labelling machine or through self-service

<TIP> := Type of label to extract
'0': Label from labelling machine
'1': Label through self-service
<QUE> ::=Indicates the data to be searched.
'0':Search the same.
'1':Search more or the same.
'2':Search more.
'3':Search less or the same.
'4':Search less.
<MOD> ::=Type of request:
'0'(30h): Absolute labels. The label with the number <ETQ> requested.
'1'(31h): Non-requested labels. First label that's never been requested.
'3'(33h): Labels by sections
'4'(34h): Labels as per scales
'5'(35h): Labels by dates
'6'(36h): Labels by vendors
<ETQ> ::=Label number.
<ITM> ::=Data to be consulted (section, vendor, etc.)
<SEC> ::=Section.

<IDV> ::=Identifier of vendor
<BAL> ::=Scale where the sale was performed.
<PTS> ::=Receipt amount due.
<HOR> ::=Time of receipt closure, when existing, if not left in blank.
<PED> ::=Receipt request status.
‘0’: Never requested.
‘1’: Order has been requested before.
<CLI> ::=Customer code, for *turn-number* configuration.
<IDA> ::=In <SSPPPP> format
<PRC> ::=Sales price.
<PES> ::=Sold quantity.
<STV> ::=Sales status.
‘0’: Add.
‘1’: Subtract.
<STP> ::=Weight status.
‘0’: Weighed.
‘1’: Not weighed.
<DTE> ::=Label date.

#### 5.15.10 <cag>::=‘9’,57d,39h. Requesting heading of global receipt (Not 8031)

Unified request of receipts, labels and receipts in local.
For 386, it can be marked as order, but it must be confirmed so that it is not consulted again as not order.
(<MOD>=’2’).
Missing discount data (386).

**CONCENTRATOR**: Q:<tqt><cag><QUE(1)><MOD(1)><TIP(1)><TQT(5)><ITM(8)><MAR(1)>
A:<tqt><TIP(1)><TQT(5)><ITM(8)><SEC(2)><IDV(4)><BAL(2)><NLI(2)>
<NC(2)><PTS(7)><HOR(8)><STS(1)><PED(1)><CLI(6)><DTE(8)>

**386:** Q:<tqt><cag><QUE(1)><MOD(1)><TIP(1)><TQT(5)><ITM(8)><MAR(1)>
A:<tqt><TIP(1)><TQT(5)><ITM(8)><SEC(2)><IDV(4)><BAL(2)><NLI(2)>
<NC(2)><PTS(7)><HOR(8)><STS(1)><PED(1)><CLI(6)><DTE(8)>
<BAR(13)><INT(5)><SRV(2)>

<QUE> ::=Indicates the data to be searched.
‘0’:Search the same.
‘1’:Search more or the same.
‘2’:Search more.
‘3’:Search less or the same.
‘4’:Search less.
<MOD> ::=Type of request:
‘0’(30h): Request according to numbering by default of concentrator.
‘1’(31h): Non-requested receipts. First receipt that has never been requested. (Not 386).
‘2’(32h): Non-requested and closed receipts.
‘3’(33h): Receipts by section (Not 386)
‘4’(34h): Receipts by scale (Not 386)
‘5’(35h): Receipts by date (Not 386)
‘6’(36h): Receipts by vendor (Not 386)
‘7’(36h): Sequential receipts (Not 386)
‘8’(36h): Receipts programmed by scales (Not 386)
‘9’(36h): Non –requested and closed receipts. (Not marked). (Not 386)
<TIP> ::=Type of receipts.

| Concentrator: | 386: |
|---|---|
| ‘0’: roll scales receipt | ‘0’: network scales receipt |
| ‘1’: labeller scale label | ‘1’: network super mode scale label |
| ‘2’: self-service scale label | ‘2’: network packaging mode scale label. |
| ‘3’: local receipt | ‘3’: scale in local receipt |
| ‘4’: packaging labels. | ‘4’: network self-service mode scale label |
| | ‘5’: local super mode scale label |
| | ‘6’: local packaging mode scale receipt. |
| | ‘7’: local self-service mode scale label. |

<TQT> ::=Receipt number.
<ITM> ::= Date to be consulted according to numbering of concentrator (section, vendor, date, scale)
If the request is about receipt in local, it is the scale.
<MAR> ::=Marck or not as order.
'0': Mark
'1': Don't mark
<SEC> ::=Section.
<IDV> ::=Identifier of vendor (Zero in self-service scales and local receipts.)
<CLI> ::=Customer's code, for a *their turn* configuration.
<BAL> ::=Equipment where the sale was closed.
<NLI> ::= Number of lines that the receipt has.
<NC> ::= Number of lines cancelled.
<PTS> ::=Amount due of receipt.
<HOR> ::=Time of closing of receipt, if closed, if not, left it in blank.(In zero in local receipts).
<STS> ::=State of receipt.
'0':Closed, finished.
'1': Open, lines are being inserted.
'2':In total, the total has been requested, but not the clearing of the total.
'3':In reload.The receipt has been closed and reopened.
'4':Closed by order. Order receipt closed.
'5': In reload by order. Order receipt reopened. (In 386, receipt cancelled).
<PED> ::= State of receipt request.
'0': Never requested before.
'1': Order requested before.
<DTE> ::=Date of receipt. (Date of reception in local receipts).
<BAR>:: =Bar codes of receipt. (Not concentrator).
<INT>:: =Internal numbering of receipt in the scales that has been carried out to mark it. (Not concentrator).
<SRV>:: =Scale that serves the receipt (scale that keeps it).(Not concentrator).

#### 5.15.11 <lig>::=':',58d,3ah. Requesting global receipt lines (Not 8031)

Unified request of receipt lines, labels, receipts in local and packaging labels.
At the moment only one line is sent each time in the 386 scales system.
Missing discount data

Q:<tqt><lig><TIP(1)><TQT(5)><NLIN(2)><LIN(1)><MOD(1)><ITM(8)><MOA(1)>
A:<tqt><TIP(1)><TQT(5)><LIN(1)><ITM(8)><MOA(1)>
<NLIN(2)><IDA(6)><PRC(7)><PES(9)><STV(1)><STP(1)><STC(1)><PTS(7)>.. (No 386)
<NLIN(2)><IDA(6)><PRC(7)><PES(9)><STV(1)><STP(1)><STC(1)><PTS(7)>

Request of the receipts lines.
<TIP> ::=Type of receipts.

| Concentrator: | 386: |
|---|---|
| '0': roll scales receipt | '0': network scales receipt |
| '1': labeller scale label | '1': network super mode scale label |
| '2': self-service scale label | '2': network packaging mode scale label. |
| '3': local receipt | '3': scale in local receipt |
| '4': packaging labels. | '4': network self-service mode scale label |
| | '5': local super mode scale label |
| | '6': local packaging mode scale receipt. |
| | '7': local self-service mode scale label. |

<TQT> ::=Receipt number.
<NLIN> ::= Line number as from the one requested.
<LIN> ::=Number of lines requested. In 386 must be '1'.
<MOD> ::=Type of request:
'0'(30h):Consulting according to numbering by default of concentrator.
'3'(33h): Receipts by section (Not 386)
'4'(34h): Receipts by scale (No 386)
'5'(35h): Receipts by date (Not 386)
'6'(36h): Receipts by vendor (Not 386)
<ITM> ::=Date to be consulted (section, vendor, etc.)
If the request is of receipts in local, it is the scale number.
<MOA> ::= Way of consulting of the item identifier <IDA>.

‘0’: PLU section <SSPPPP> (not implemented in 386)
‘1’: Internal code.

<IDA> ::= Item identifier.
<PRC> ::=Sales price.
<PES> ::=Quantity sold.
<STV> ::= Sales status.
‘0’: Add.
‘1’: Subtract.
<STP> ::= Weight status.
‘0’:Weighed.
‘1’: Not weighed.
<STC> ::= Cancellation status, if cancelled it neither adds nor substracts in the total. (Not used in 386).
‘0’: Not cancelled.
‘1’: Cancelled line (doesn’t add nor subtacts an the total).

#### 5.15.12 <cag>::=‘;’,59d,3bh. Requesting for clearing global receipt (Not 8031) (Not 386)

Unified request of clearing receipts and labels.
Q:<tqt><cag><QUE(1)><MOD(1)><TIP(1)><TQT(5)><ITM(8)>
A:<RPS><ERR>
<QUE> ::=Indicates the data to be searched:
‘0’:Search the same
‘1’:Search more or the same.
‘2’:Search more.
‘3’:Search less or the same.
‘4’:Search less.
<MOD> ::=Type or request:
‘0’(30h): Consulting according to numbering by default of concentrator.
‘1’(31h): Non-requested receipts. First receipt never requested before.
‘2’(32h): Non-requested and closed receipts.
‘3’(33h): Receipts by section.
‘4’(34h): Receipts by scale.
‘5’(35h): Receipts by date.
‘6’(36h): Receipts by vendor.
<TIP> ::=Type of receipt:
‘0’: Receipt-roll scales receipt.
‘1’: Labeller scale label.
‘2’: Self-service scale label.
<TQT> ::=Receipt number.
<ITM> ::=Date to be consulted (section, vendor, etc.)

#### 5.15.13 <imt>::=‘<’,60d,3ch. Requesting printing receipts (Not 8031) (Not 386)

Sending order of printing the receipts carried out up till now, listed as the one done when finishing the execution (switching of) the concentrator.
Q:<tqt><imt>
A:<RPS><ERR>

#### 5.15.14 <nta>::=‘=’,61d,3dh. Requesting alternative numbering (Not 8031) (Not 386)

Asking the number of alternative receipts according to logical numbering of the concentrator (configuration) as from the sequential receipt number.
Q:<tqt><nta><TYP(1)><TQT(5)><ITM(8)>
A:<RPS><ERR>
<TYP> ::=Type of receipt.
‘0’: Roll scale.
‘1’: Labeller scale.
‘2’: Self-service scale.
‘3’: Receipt in local.
‘4’: Packaging label.
<TQT> ::=Number of sequential receipt of each type.
<ITM> ::=Scale number, only necessary for the consulting of receipts in local.

**5.15.15 <ent>::='>',62d,3eh Requesting heading of input/output indicator (Not 8031) (Not 386)**

Consulting the receipt heading carried out in an equipment of the type, goods input/output indicator.

An absolute numbering for all the equipments of this type is followed in the concentrator. It is only initiated in the zero setting of the concentrator with the receipt clearing. Apart from this numbering, the numbering by equipment and date and time is stored, since it is possible to initiate the numbering of the equipments without initiating the concentrator.

Q:<tqt><ent><QUE(1)><MOD(1)><TQT(5)><EQU(2)><DTE(8)><HOR(8)>
<MAR(1)>
A:<tqt><TQT(5)><NLI(2)><EQU(2)><IDV(4)><MAR(1)><PRO(6)><ALB(8)>
<DTE(8)><HOR(8)>

<TQT> ::=Number of receipt requested in the question, real in the reply.
<QUE> ::=Indicates the data to be searched.
'0': Search the same.
'1': Search more or the same.
'2': Search more.
'3': Search less or the same.
'4': Search less.
<MOD> ::=Type of request:
'0'(30h):Consulting according to absolute numbering of the concentrator.
'1'(31h):Number of receipt and equipment with date and time.
'2'(32h):Non-requested and closed receipts.
<NLI> ::=Number of receipt lines.
<EQU> ::= Equipment number.
<DTE> ::= Date on which it is carried out.
<HOR> ::= Time in which it is carried out.
<MAR> ::=
In the question, Mark as order.
'0' : Mark as order.
'1' : Don't mark.
In the reply. Current state.
'0' : Marked.
'1' : Not marked.
<IDV> ::= Vendor number.
<PRO> ::= Supplier number.
<ALB> ::= Delivery note number.

**5.15.16 <lent>::='?',63d,3fh. Requesting lines of input/output indicator (Not 8031) (Not 386)**

Q:<tqt><lent><MOD(1)><TQT(5)><EQU(2)><LIN(2)><DTE(8)><HOR(8)>
A:<tqt><TQT(5)><LIN(2)><SEC(2)><PLU(4)><COD(6)><STP(1)><CNT(9)><PTS(7)>
<OES(1)>

<TQT> ::=Number of receipt requested.
<MOD> ::=Type of request:
'0'(30h):Query according to absolute numbering of the concentrator
'1'(31h): Receipt and scale unit number with date and time.
<EQU> ::= Scale unit number.
<LIN> ::= Line number.
<DTE> ::= Date of operation.
<HOR> ::= Time of operation.
<SEC> ::= Number of item section.
<PLU> ::= Item PLU number.
<COD> ::= Code number.
<STP> ::=
'0': Weighing.
'1': Units.
<CNT> ::= Quantity
<PTS> ::= Line amount due.
<OES> ::= Type of operation.
'0': Goods input.
'1': Goods output.

**5.15.17 <mar>::='?',63,3fh. Requesting receipt marking (Only 386)**

Confirming a receipt as an order, so that the next consulting of not ordered doesn't give it again.

Q:<tqt><mar><QUE(1)><MOD(1)><TIP(1)><TQT(5)><ITM(8)>
A:<RPS><ERR>
<QUE> ::=Indicates the data to be searched.
'0': Search the same.
'1': Search more or the same.
'2': Search more.
'3': Search less or the same.
'4': Search less.

<MOD>::=Type of request:
'0'(30h)- Consulting according to numbering by default of the concentrator.
<TIP>::= Type of receipt:
'0': Receipt network scale.
'1': Super-mode network scale label.
'2': Packaging-mode network scale label.
'3': Network scale receipt.
'4': Self-service mode scale network label.
'5':Super-mode network scale label.
'6':Packaging-mode network scale label.
'7':Self-service mode network scale label.
<TQT>::=Scale receipt number.
<ITM>::=Date to be consulted (scale number).

**5.16 #<tur>::='<',60d,3ch-SPECIAL MESSAGES FOR TURN-NUMBERS.(Not 8031) (Not 386)**

The concentrator can work with two types of turn-number systems, one in which the customer number is assigned before initiating the receipt, the other where it is assigned at the end. In both cases, a customer number is assigned to each receipt and all the receipts of a customer can be individually consulted.The customer numbers can be repeated and are distinguished within them by the time in which have been requested.
They can also be distinguished with a continuous internal numbering of the concentrator (customer code).

**5.16.1 <ean>::='0',48d, 30h.Requesting EAN codes of a customer code**

NOT IMPLEMENTED.

Request the EAN code numbers of the scale receipt of a customer. They are returned by order of realization in the scales.

Q: <tur><ean><CLI(6)><NUM(2)><DTE(8)><HOR(8)><NTQT(2)>.
A: <tur><DTE(8)><HOR(8)><NTQ(2)><EAN1>..<EANn>.
<CLI> ::=Customer turn-number.
<NUM> ::=From which receipt is requested (Order begun by 1).
<DTE> ::= Date of the first time that the customer has been requested. If it is zero, the first that has never been requested is sent.
<HOR> ::=Time in which the customer has been requested.
<NTQ> ::= Number of EANs of scale sent.
<EANx>::= EAN codes of the scale receipts (Maximum 10).

**5.16.2 <tqt>::='1',49d,31h.Request the receipt numbers of a customer.**

Request the receipt number of a customer. Mark the customer and the receipts as orders.

Q: <tur><tqt><CLI(6)><NUM(2)><DTE(8)><HOR(8)><NTQT(2)>.
A: <tur><DTE(8)><HOR(8)><NTQT(2)><SEC1(2)><TQT1(5)><PTA1(8)>..
<SEC1(2)><TQTn(5)><PTAn(8).
<CLI> ::= Customer turn-number
<NUM> ::= As from the receipt that is requested (order is not the absolute receipt number)
<NTQT>::= Number of scale receipts sent.

<DTE> ::= Date on which the customer has been requested for the first time. If zero, the first one that has never been requested is sent.
<HOR> ::= Time on which the customer has been requested for the first time.
<TQTx>::= Numbers of the scale receipts of the customer (Maximum 10).
<SECX>::= Section of each receipt.
<PTAx>::= Amount due of each receipt.

#### 5.16.3 <dat>::='2',50d,32h. Consulting customer data

Consulting the data of a customer. Can be done using the customer number (ordered by the codes assigned by the concentrator), or by using the code directly.
Q: <tur><dat><COD(8)><CLI(6)><QUE(1)><MOD(1)>
A: <tur><COD(8)><CLI(6)><DTE(8)><HOR(8)><NTQS(2)> <PTS(8)>.
<CLI> ::=Customer turn-number.
<COD> ::=Customer turn-number code (internal in the concentrator)
<QUE> ::=Indicates the data to be searched.
'0': Search the same.
'1': Search more or the same.
'2': Search more.
'3': Search less or the same.
'4': Search less.
<QUE> ::=Indicates the data to be searched.
'0': Search the same.
'1': Search more or the same.
'2': Search more.
'3': Search less or the same.
'4': Search less.
<MOD> ::=Way of asking the customer.
'0': Asking by order <CLI><COD>
'1': Searching by order <COD>
<NTQS>::=Number of scale receipts sent.
<DTE> ::=Date on which the customer was asked. If it is zero, it means that it has not yet been requested.
<HOR> ::=Time of the first time that the customer has been requested.
<PTS> ::=Total amount due of the customer.

#### 5.16.4 <tqc>::='3',51d,33h. Requesting customer code number receipts

Requesting a customer's scale receipt numbers useing the code
(assigned by the concentrator, not of the number). Doesn't mark as orders.

Q: <tur><tqc><COD(8)><NUM(2)>< NTQT(1)>.
A:<tur><NTQT(1)><TQT1(5)><SEC1(2)><IDV1(4)><BAL1(2)><DTE1(8)><HOR1(8)>
<STS1(7)> <NLI1(2)><STS1(1)>..
<TQTn(5)><SECn(2)><IDVn(4)><BALn(2)><DTEn(8)><HORn(8)><PTSn(7)>
<NLIn(2)><STSn(1)>

<COD> ::= Customer turn-number code (internal of the concentrator)
<NUM> ::=As from the receipt requested (Order, not absolute number of receipt).
<NTQT>::=Number of scale receipts sent.
<TQTx>::=Numbers of the customer's scale receipts. (Maximum 4).
<SECx>::=Receipt section.
<IDVx>::=Identifier of vendor.
<BALx>::=Scale unit in which the sale was closed.
<NLIx>::=Number of lines the receipt has.
<PTSx>::=Receipt amount due.
<HORx>::=Time of receipt closing, if closed it not left in blank.
<STSx>::=Receipt state.
'0': Closed, it has been finished.
'1': Open, some lines are being inserted.
'2': In total, the total has been requested, but not the total clearing.
'3': In reload, the receipt has been reopened.

<DTEx>::=Date of receipt

**5.17 #<fmt>::='=',61d,3dh-LABEL AND RECEIPT FORMAT**

**5.17.1 <ett>::='0',48d, 30h.Label/receipt programming (Not 386)**

Q:<fmt><ett><PRG><TYP><JST><SIZ><TLT><DIR><XCR><YCR><FLD><DTA>

<TYP> ::=Type

'0' - label 0
'1' - label 1
'2' - label 2
'3' - label 3
'4' - label 4
'5' - label 5
'6' - label 6
'7' - label 7
'8' - label 8
'9' - label 9
'A' – Receipt heading,
'B' – Line of receipt continuation.
'C' – Line of receipt repetition.
'D' – Line of cancellations.
'1' – weight receipt line
' ' - return weight receipt
' ' - cancellation weight receipt
'2' – line of constant receipt
' ' - return
' ' - cancellation
'3' –line of item receipt
' ' - return
' ' - cancellation
' ' – sale section line to section
' ' – discount line
' ' – payment line
' ' - payment cancellation
' ' – offer line
'4' – line of receipt subtotal
' ' – line of discount in subtotal
'4' – line of receipt total
' ' – line of discount to total
'5' – End of receipt.

<PRG> ::= Modification of field.

'1' – There are more data to modify the field (programming)
'0' – Using, No programming

<JST> ::= Alignment

'0' – Right
'1' – Left
'2' – Centred

<TLT> ::= Type of letter

<XCR>::= X co-ordinate (Character in receipt)
<YCR>::= Y co-ordinate Y (Line in receipt)
<FLD>::= Identifier of field.
'00' – NETWORK WEIGHT
' ' – GROSS WEIGHT
' ' – TARE
'0x' - TOTAL WEIGHT
'00' – UNITARY WEIGHT
'02' – UNITS (PIECES)
'03' – UNIT PRICE
'0x' - AMOUNT DUE
'0x' - TOTAL AMOUNT DUE
'0x' – RECEIPT NUMBER
'00' – NUMBER OF ITEMS ON RECEIPT

'0x' – PACKAGING DATE (DATE)
- TIME
'00' – ITEM CODE
'00' – PLU CODE
'00' – SECTION CODE
'00' – SCALE UNIT CODE
'00' - HEADING
'00' - LEGEND
'00' – BAR CODE-(6)
'0x' – BEST BEFORE DATE
'00' – ORDER NUM.
'00' – LOT NUM.
'00' – ITEM TEXT
'00' - REGISTRATION
'00' - CUSTOMER
'00' - VENDOR
'00' – VENDOR CODE
'00' – VENDOR NUM.
'00' - SUPPLIER
' ' – SUPPLIER CODE
' ' – SUPPLIER NUMBER
'00' - VAT(6)
'00' – VAT TYPE (6)
'00' – PAYMENT MODE (6)
'00' - STOCK
' ' – FIXED FIELD (some)
' ' –Line left in blank.
' ' –Separation line.

**5.17.2 <etq>::='1',49d,31h. Label programming (amplified) (Not 386)**

Q:<fmt><etq><MOD><TYP><XCR><YCR><DIR><FNT><SIZE>
<MOD> ::=
<MOD>::= Identifier of operation
'0' = Clearing label
'1' = Text
XCR = X co-ordinate x
YCR = Y co-ordinate y
TYP = Inverse mode (1) normal (0)
DIR = Printing direction (0,1,2,3)
FNT = Type of character
SIZE= Number of characters
DATA= characters
'2' = Box
XCR = X co-ordinate
YCR = Y co-ordinate
TYP = Line width
DIR = Width
FNT = Hight
'3' = Bar code
XCR = Co-ordinated x
YCR = Co-ordinated y
TYP = Type of code
DIR = Density
FNT = Hight
SIZE= data bytes
DATA= code
(etc..)

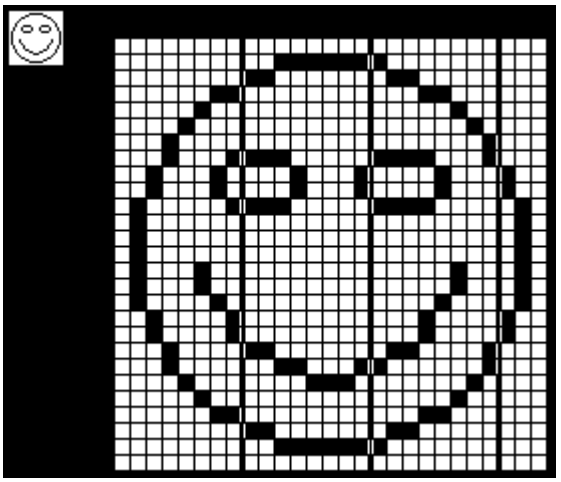
**5.17.3 <bmp>::='0',48d, 30h. Logotype programming. (Only 386)**

Send logotype to the scale units. The maximum size of the logo is 384 pixels width and 128 pixels length.

Q:<fmt><bmp><MOD(1)><DIR(2)><IDL(1)><SCA(3)><BYW(3)><BIW(3)><YPS(3)><DTA(BYW*2)>
A:<RPS><ERR>
IMPORTANT: Some fields should be sent without 'null' <NUL> at the end.

Remark: In preliminary versions the message was:
Q:<fmt><bmp><SCA(3)><BYW(3)><BIW(3)><YPS(3)><DTA(BYW*2)>
<MOD>::=Indicates if the programming is for the scale or for section.
'0':Section.
'1':Scale
'2':System.
<DIR>::=Number of scale or section/counter.
<IDL>::= Identifier of Logo.Must be one.
<SCA>::= ScanLines (logo hight in pixels ) IMPORTANT: Sent without 'null' at the end.
<BYW>::= ByteWidth (logo width in bytes) IMPORTANT: Sent without 'null' at the end.
<BIW>::= BitWidth (logo width in pixels ) IMPORTANT: Sent without 'null' at the end.
<YPS>:: Line number IMPORTANT: Sent without 'null' at the end.
<DTA>::Maps of bits of all the line in hexadecimal. The bits = 1 will be printed as black and the bits =zero as white. Each byte sent is interpreted as 8 pixels. IMPORTANT: Sent without 'null' at the end.

Example:
The face to the left of the figure is formed by the map of bits shown on the right.



For the first line send this:
(SCA)"027", (BYW)"004", (BIW)"027", (YPS)"000",(DTA)'00000000..'
For the second :
(SCA)"027", (BYW)"004", (BIW)"027", (YPS)"001", (DTA)'003F8000..'
For the third:
(SCA)"027", (BYW)"004", (BIW)"027", (YPS)"001", (DTA)'000C6000..'
For the fourth:
(SCA)"027", (BYW)"004", (BIW)"027", (YPS)"001", (DTA)'03001800..'
etc.
And for the last:
(SCA)"027", (BYW)"004", (BIW)"027", (YPS)"026", (DTA)'00000000..'

### 5.18 #<env>: :='?',63d,3fh-PACKAGING DATA (ONLY 386)

#### 5.18.1 <ing>::='0',48d, 30h. Ingredients programming.

Q:<env><ing><COD(4)><TXT(25)>
A:<RPS><ERR><COD>::=Ingredient code.
<TXT>::=Ingredient text.

#### 5.18.2 <?ing>::='1',49d, 31h. Ingredients consulting.

Q:<env><?ing><QUE(1)><COD(4)>
A:<env><COD(4)><TXT(25)><ELI(1)>
A:<RPS><ERR>

<QUE>::= Indicates the data to be searched.
'0':Search more.
'1': Search more or the same.
'2': Search more.
'3':Search for menor igual.
'4':Search less.
<COD>::=Ingredient code.
<TXT>::=Ingredient text.
<ELI>::= Not used.

#### 5.18.3 <bing>::='2',50d, 32h. Erasing ingredients.

Q:<env><bing><COD(4)><STE(1)>
A:<RPS><ERR>
<COD>::=Ingredient code.
<STE>::=Not used.

### 5.18.4 <inenv>::='3',51d, 33h. Initiating packaging (train).

A labelling sequence is programmed, at the end of which the train will automatically stop and wait for a new order.
Q:<env><inenv><ART(6)><VEL1(1)><VEL2(1)><VEL3(1)><VEL4(1)><TIPO(1)>
<VALOR(6)><FMT(2)>
A:<RPS><ERR>
<ART>::=Item code to be labelled, should be programmed.
<VEL1>::= Speed of the first labelling train belt (inbelt).
<VEL2>::= Speed of the second labelling train belt (separation).
<VEL3>::= Speed of the third labelling train belt (weighting).
<VEL4>::= Speed of the fourth labelling train belt (labelling).
<TIPO>::=Indicates the type of pre-selection to work with:
'0' :without pre-selection. The train can only be stopped manually.
'1' :pre-selection by number of parcels. The train will automatically stop when it performs the labels indicated in "VALOR".
'2' :pre-selection by weight. The train will automatically stop when it equals or surpasses (the accumulated weight), the datum indicated in "VALOR".
'3' :pre-selection by amount due. The train will automatically stop when it equals or surpasses (the accumulated weight), the datum indicated in "VALOR".
<VALOR>:=Pre-selection to which the train is programmed to stop automatically. This can be by number of labels, weight or amount due, according to what is indicated in "TIPO".
<FMT>:=Format of label to use, should be programmed on the train (the number is the order in which they shall be programmed), otherwise there will be no format to use and therefore no label. The correct values are:
"0" to "N": programmed labels.
"98": train default label.
"99": no label.
NOTE: Permitted speed values range from 0 to 9, (slower to faster), with'*' indicating the train uses the speed that was last programmed.

### 5.18.5 <finenv>::='4',52d, 34h. Finalising packaging (train).

Allows partial totals to be requested (whether cleared or not) and finalises the labelling sequence.

Q:<env><finenv><ST_CINTAS(1)><ST_TOTAL(1)>
R::<env><VND(4)><ART(6)><UND(8)><PTS(9)><PES(9)><ST_CINTAS(1)>
<ST_CINTAS>::=Indicates whether to stop or continue.
'0':Continue.
'1':Stop.
<ST_TOTAL>::= Format of the sequence's total data request (since last clearing).
'0':Request for total without clearing.
'1':Request for and clearing of total.
'2':Request for total by printing total label and not clearing total.
'3':Request for total by printing total label and clearing total.
<VND(5)>::=Password for current operator.
<ART(7)>::=Packaged item code.

<UND(9)>::=Number of labels performed.
<PTS(10)>::=Total amount due for packaged item.
<PES10)>::=Total weight (in grams) of packaged item.

### 5.19 #<TRN>::='>',62d,3eh-LABELLING TRAIN PROGRAMMING (NOT 386)

Programming items to the labelling train.
When an item's programming takes place in the concentrator, it is not sent to the labelling train. It must be programmed apart with these messages.It is possible to know if the concentrator has sent the programming order to the train and the order can also be cancelled.

#### 5.19.1 <prg>::='0',48d,30h. Item train programming

Order of sending the item to the labelling train as it is in the concentrator.
Q: <trn><prg><EQU(2)><TYP(1)><IDA(6)>
<EQU> ::=Scale unit to where the order has been sent. If it is zero, send to all.
<TYP> ::= Type of order to be sent to the train.
'1': Programming item as it is in the concentrator
If it doesn't exist or has been erased, it will be cleared in the train.
'2': Change price of item.
'3': Programming item descriptives
The descriptives of type 2 are sent to the labelling train.
<IDA> ::= Internal code of item. If it is zero, programme all.

#### 5.19.2 <prg?>::='1',49d,31h.Sending to train control

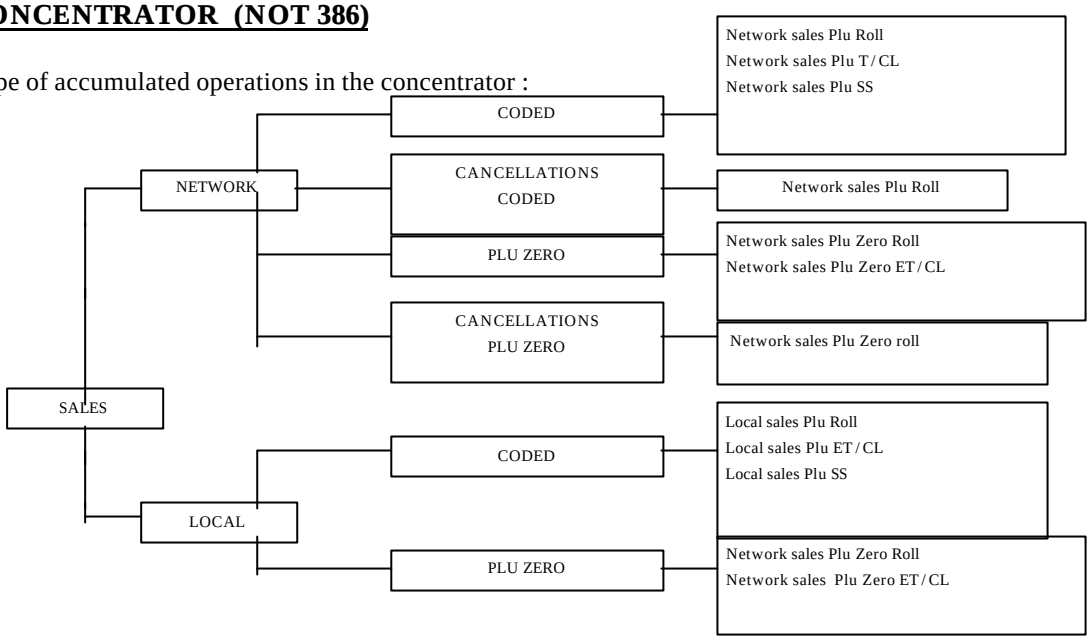
It is possible to know if the concentrator has sent the programming order to the labelling train to know if it is being programmed. The order can also be cancelled.
Q: <trn><prg?><EQU(2)><ORD(1)><QUE(1)>
A: <trn><PRO(1)>
<EQU> ::=Equipment to know if it is programmed.
<ORD> ::= Type of order:
'0': Programming item
'1': Clearing series
<QUE> ::= Operation to be carried out:
'0': Consulting order
'1': Clearing order
<PRO> ::= State of programming:
'0': Not programmed
'1': Programmed

#### 5.19.3 <blot>::='2',50d,32h.Clearing series

Orders the equipment to clear the Series.
Q: <trn><blot><EQU(2)>
<EQU> ::= Equipment number to which the order is sent. If it is zero, send to all.

## 6. USE OF THE MESSAGES FOR THE CALCULATION OF TOTALS OF THE CONCENTRATOR (NOT 386)

Type of accumulated operations in the concentrator :




In local, the operations cancelled are not expressed separately; it is as if they didn't exist.

SALES : It is for the operations in positive, addition or subtraction of weight, and addition, subtraction or multiplication of units.

CANCELLATIONS : It is the annulment of any sale operation of the ones carried out.

PLU ZERO : Includes any operation of addition, subtraction or multiplication of weight and units with direct import.

The amount dues are calculated from the:

Operations of weight : Addition.

Operations of subtraction : Subtraction.

Cancellations one weight : Subtraction.

Cancellations one subtraction : Addition.

The operations are calculated:

Always adds 1 for each operation received in the concentrator. Sales and cancellations always add.

### 6.1 AMOUNT DUE, NUMBER OF OPERATIONS AND RECEIPTS IN TOTAL (NOT 386)

## Total amount due of sales in the concentrator=

Network Sales Amount due PLU Roll -
Network Cancellations Amount due PLU Roll +
Network Sales Amount due PLU ET +
Network Sales Amount due PLU SS +
Network Sales Amount due PLU Zero Roll -
Network Cancellations Amount due PLU Roll +
Network Sales Amount due PLU Zero ET+
Local Sales Amount due PLU Roll +
Local Sales Amount due PLU ET +
Local Sales Amount due PLU SS +
Local sales amount due PLU Zero Roll +
Local sales amount due PLU Zero ET

## Number of operations in total of concentrator =

Number of lines of receipt Network Sales PLU Roll+
Number of lines of receipt of Cancellations Network PLU Roll +
Number of Labels Sales Network PLU ET/CL +
Number of Sales Network PLU SS +
Number of lines of receipt with Sales Network PLU Zero Roll+
Number of receipt lines with Cancellations Network PLU Zero Roll+

Number of Labels Sales Network PLU Zero ET+
Number of PLU weighings by weight in Local Sales PLU Roll+
Number of units of unitary PLU in Local Sales PLU Roll+
Number of weighings of PLU by weight Local PLU ET +
Number of units of PLU unitary Local PLU ET+
Number of weighings of PLU by weight Local PLU SS +
Number of units of PLU unitary local PLU SS+
Number of sales by weight of Zero PLU Local Roll +
Number of units of PLU Zero Local PLU Zero Roll+
Number of sales by weight of PLU Zero Local PLU ET/CL +
Number of PLU units Zero Local PLU Zero ET/CL

## Number of receipts of the concentrator =

Number of receipts in network (Network PLU Roll and Network PLU Zero Roll) +
Number of Labels in network of ET (Network PLU ET + Network PLU Zero ET) +
Number of labels in network SS (Network PLU SS) +
Number of receipts in local (Local PLU Roll and Local PLU Zero Roll) +
Number of Labels in local of ET (Local PLU ET + Local PLU Zero ET) +
Number of Labels in local SS (Local PLU SS)

### 6.2 CALCULATION OF AMOUNT DUE AND NUMBER OF OPERATIONS (NOT 386)

For calculating, it is necessary to add the amount dues and operations of the protocol queries detailed as follows:

**Scales sales total by sections, subsections and families (Group '8', Query '3')**

| Protocol consulting | Amount due Consulted |
|---|---|
| Pts and Ops of type '0' (section, subsection and family) | Network sales PLU Roll -<br>Cancellations Network PLU Roll |
| Pts and Tqt of type '5' (section, subsection and family) | Network sales PLU ET |
| Pts and Tqt of type '4' (section, subsection and family) | Network sales PLU SS |
| Pts and Ops of type '2' (section) | Network sales PLU Zero Roll -<br>Cancellations Network PLU Zero Roll +<br>Network sales PLU Zero ET |
| Pts and Ops of type '1' (section, subsection and family) | Local sales PLU Roll +<br>Local sales PLU ET +<br>Local sales PLU SS |
| Pts and Ops of type '3' (section) | Local sales PLU zero Roll +<br>Local sales PLU zero ET |

**Total of items sales in scale units (Group '8', Query '4')**

| Protocol query | Importe Consultado |
|---|---|
| Pts and Ops of type '0' | Network sales PLU Roll -<br>Network cancellations PLU Roll |
| Pts and Ops of type '2' | Network sales PLU ET |
| Pts and Ops of type '3' | Network sales PLU SS |
| Can not be consulted, the PLU zero is not coded. | Network sales PLU Zero Roll +<br>Network cancellations PLU Zero Roll<br>Network sales PLU zero ET |
| Pts and Ops of type '1' | Local Sales PLU Roll +<br>Local sales PLU ET<br>Local sales PLU SS |
| Can not be consulted, the PLU zero is not coded. | Local sales PLU zero Roll +<br>Local sales PLU zero ET |

**Total vendors of scales (Group '8', Query '5')**

| Protocol consulting | Amount due Consulted |
|---|---|
| Pts and Ops of type '0' | Network sales PLU Roll |
| Pts and Ops of type '2' | Network sales PLU Roll +<br>Network cancellations PLU zero Roll |
| Pts and Ops of type '4' | Network sales PLU ET |
| Don't have vendors | Network sales PLU SS |
| Pts and Ops of type '5' | Network sales PLU Zero Roll |

| | |
|---|---|
| Pts and Ops of type'6' | Network sales PLU Zero ET |
| The local vendors are not of network<br>(different codification). | Local Sales PLU Roll<br>Local Sales PLU ET<br>Local Sales PLU SS<br>Local Sales PLU zero Roll<br>Local Sales LU zero ET |

**General total of the concentrator (Group '8', Query '8')**

| Protocol Query | Amount due Consulted |
|---|---|
| Pts and Ops of type '0' | Sales Red PLU Roll -<br>Cancellations Network PLU Roll |
| Pts and Tqt of type '5' | Sales Network PLU ET |
| Pts and Tqt of type '4' | Sales Network PLU SS |
| Pts and Ops of type '2' | Sales Network PLU Zero Roll -<br>Cancellations Red PLU Zero Roll +<br>Sales Red PLU Zero ET |
| Pts and Ops of type '1' | Local Sales PLU Roll +<br>Local Sales PLU ET +<br>Local Sales PLU SS |
| Pts and Ops of type '3' | Local Sales PLU zero Roll +<br>Local Sales PLU zero ET |

**Total general by scale units (Group '8',query '7')**

| Protocol query | Amount due Consulted |
|---|---|
| Pts and Ops of type '0' | Network Sales PLU Roll<br>Cancellations Network PLU Roll<br>Network Sales PLU ET<br>Network Sales PLU SS |
| Ptd and Opd of type '0' | Network Sales PLU Zero Roll<br>Cancellations Red PLU Zero Roll<br>Network Sales PLU Zero ET |
| Pts and Ops of type '1' | Local Sales PLU Roll<br>Local Sales PLU ET<br>Local Sales PLU SS |
| Ptd and Opd of type '1' | Local Sales PLU zero Roll<br>Local Sales PLU zero ET |

**6.3 QUERIES FOR THE CALCULATION OF THE NUMBER OF RECEIPTS**

**(NOT 386)**

**Total sales of scales by sections, subsections and families (Group '8', Query '3')**

| Protocol consulting | Numero de receipts consultado |
|---|---|
| Tqt of type '0' | Number or receipts in network |
| Tqt of type'4' | Number of labels in network of ET |
| Tqt of type '5' | Number of labels in network SS |
| Tqt of type'1' | Number of receipts in local<br>Number of Labels in local of ET<br>Number of Labels in local SS |

**Total of sales of items in scale units (Group '8', Query '4')**

| Protocol consulting | Receipts number consulted |
|---|---|
| The number of receipts cannot be consulted. | |

**Total vendors of scales (Group '8', Consulted '5')**

| Protocol query | Number of receipts consulted |
|---|---|
| Tqt of type '0' | Number of receipts in network |
| Tqt of type '4' | Number of labels in network of ET |

| | |
|---|---|
| Don't have vendors | Number of labels in network SS |
| There isn't receipt in local for the vendors. | Number of receipts in local<br>Number of labels in local de ET<br>Number of labels in local SS |

**General total of the concentrator (Group '8', Query '8')**

| Protocol query | Number of receipts consulted |
|---|---|
| Tqt of type '0' | Number of receipts in network |
| Tqt of type '4' | Number of labels in network of ET |
| Tqt of type '5' | Number of labels in network of SS |
| Tqt of type '1' | Number of receipts in local<br>Number of Labels in local de ET<br>Number of Labels in local SS |

**General total by scale units (Group '8',query '7')**

| Protocol query | Number of receipts consulted |
|---|---|
| Tqt of type '0' | Number of receipts in network.<br>Number of labels in network of ET<br>Number of labels in network SS |
| Tqt of type '1' | Number of receipts in local<br>Number of Labels in local of ET<br>Number of Labels in local SS |

## 7. FORMAT SOCKETS COMUNICATION FORMAT

The notation used for the specification of this protocol is:

| | |
|---|---|
| <field> | Field of a message |
| <field (x)> | x= Definition of the field width in bytes. |
| <field>::=description | Description of the field. |
| *XX*d | Number expressed in decimal base. |
| *XX*h | Number expressed in hexadecimal base. |
| 'x' | Character. |
| Q:... | Request to the concentrator |
| A:... | Reply to the concentrator |
| [option] | Indicates an optional field. |
| \| (option 1 \| option 2) | Indicates two or more possibilities. |
| <field>...<field> | The field is repeated several times. |

The same information requested to the concentrator by serie or by local network can be requested by sockets.

### 7.1 Physical format sockets

The concentrator programme works as customer of a system of sockets. It is possible to configurate three sockets ports:
- warning (DGRAM) : the remote computer informs that it has orders to send.
· work (STREAM) : orders, results, receipts, lines of receipts

## 7.2 Work mode with sockets

The information sent to the concentrator will have a serie of ASCII channels that will correspond to the different orders and to the results of these orders, or to the receipts.
The work mode will be the following:

**Orders:**

1. The message that there are orders pending to be sent is awaited through the orders warning port.
   Q: <ORDEN>

2. If it is received, the work socket is opened as SOCK_STREAM .
3. The concentrator sends an order request message. Then, the remote computer replies the order.
If there are problems, the socket is closed and turns to point 1.
4. The work socket is closed.

5. The work socket is opened.
6. The concentrator sends the reply to the remote computer which replies if it is correct or not.
7. The work socket is closed.
Both the orders data and the results will have an order number of 4 bytes to identify the replies with the orders.

**Receipts:**

1.If there are receipts to send to the remote computer, connect it to the work socket.
If it is not possible, try to do it again every certain configurable time (10 seconds by default).
If there is no receipt, it consults every certain configurable time (10 seconds by default).
2.Connect it to the work socket.
3.Sends the receipt and waits for an reply from the remote computer. If there are problems, the port is closed and after the configurable time to point 1.
4. Closes the work socket.
5. Opens the work socket.
6. Sends a receipt line and awaits the reply.
7. Closes the work socket.
If there are more than 5 lines, goes to point 5, if not to 1.

This work mode makes it impossible to carry out the following operations from the scales (once sent):
- Receipt reload.
- Cancellation of receipt lines.

## 7.3 Socket messages format:

<MESAGES>::=<IDN><IDR><BUF>
<IDN>::=<BYTE><BYTE><NUL> Data identifier for configurable socket. By default:
Orders: "11"
Receipts "10"
<IDR>::=<BYTE><NUL> Reply of result
0:Correct .The scales system has replyed.
1:Error, problem in scales system. There is no information in <BUF>
<NUL>::=<BYTE> 0d,0h. Character nulled.
<BUF>::=<MEN><RES> Message, with a total of 1024 bytes.
<MEN>::=<IDO><IDG>[<IDM>]<DAT0>...<DATn><NUL> Message to be sent to the scales system.
<IDO>::=<BYTE><BYTE><BYTE><BYTE> Identifier of order number.
<IDG>::=<BYTE> Identifier of group of messsages.
<IDM>::=<BYTE> Identifier of message.
<DATx>::=<BYTE>..<BYTE><BAR>Messages data of variable length. Ended by < |>
<BAR>::=<BYTE> |, 124d, 7Ch.
<RES>:: <BYTE>.<BYTE> rest of bytes (doesn't matter that there are) until arriving at1024 the <BUF>
The maximum length of any complete message<MEN> is 255 bytes.

### 7.4 Request for data from the scale units

<REQUEST>::=<MESSAGE>

IDG and IDM define the request.
DATx are parameters associated to such data request:
The system also replies with a message with the associated data headed by IDG.
A: <IDG><DAT0>.<DATn> (IDN,IDR,IDO,IDG,DAT0,.,DATn,NUL)
or with a message with the following format:
A:<RSP(1)><ERR(2)> ( IDN,IDR,IDO,RSP,ERR1,ERR2,NUL)
<RSP>::=<OK> | <KO>
<OK>::=<BYTE>,35d, 23h= Message treated correctly.
<KO>::=<BYTE>,36d, 24h= Error when carrying out the requested operation.
<ERR>::= Has information that specifies the error made (specific of message).
<ERR>::=<BYTE>,<BYTE>= Code or state error: see list in serie communication section.
<ERR> is not sent if <RSP> is ACK.

## 7.5 System programming:

<PROGRAMMING>::=<MESSAGE>

IDG and IDM define the programming to be carried out, while DATx are parameters associated to this programming.

The system replies at the same time with the message:
A:<RSP(1)><ERR(2)> ( IDN,IDR,IDO,RSP,ERR1,ERR2,NUL)

<RSP>::=<OK>|<KO>
<OK>::=<BYTE>,35d, 23h= Message treated correctly.
<KO>::=<BYTE>,36d, 24h= Error when carrying out the requested operation.
<ERR>::=Has information that specifies the error committed.
<ERR>::=<BYTE>,<BYTE>=Error or state code : see list in serie communication section, the communications protocol for concentrator.

## 8.6 Modification of messages for working with sockets:

The following messages have been modified for the sockets communication.

<tqt>::=';',59d,3bh – RECEIPTS CONSULTING
<cag>::='9',57,39h. Global receipt heading query.

Unified query of receipts, labels and receipts in local.

Q:<tqt><cag><QUE(1)><MOD(1)><TIP(1)><TQT(5)><ITM(8)><MAR(1)>
A:<tqt><TIP(1)><TQT(5)><ITM(8)><SEC(2)><IDV(4)><BAL(2)><NLI(2)>
<NC(2)><PTS(7)><HOR(8)><STS(1)><PED(1)><CLI(6)><DTE(8)>
<BAR(13)><INT(5)>
The following fields are added:
<BAR>::=Bar codes of the receipt.
<INT>::=Internal numbering of receipt in the scale that has been carried out to mark it.

# 8. BINARY PROTOCOL

## 8.1 DESCRIPTION OF A BINARY PROTOCOL MESSAGE

A binary protocol message is made up of 3 parts: the heading of the message, the body of the message (data), and the end of the message.

### 8.1.1 Heading

| Field | Length | Description | | |
|---|---|---|---|---|
| STX | 1 byte | 2d, 02h | | |
| IDG | 1 byte | Binary protocol identifier<br>126d, 7Eh → Data without compression<br>127d, 7Fh → Data with LZW compression ***(not implemented)*** | | |
| DST: Flags | 4 bits | Flag | Description | |
| | | 0x0<br>0x1<br>0x2 | By default<br>No backup *(returns an error if the destination scale is a backup)*<br>Next *(searches the next scale connected to the network)* | |
| DST: Type | 4 bits | Id | Description | Dst Contents |
| | | 0<br>1<br>2<br>3<br>4<br>5 | Equipment connected<br>Counter<br>Scale<br>System<br>Master system<br>Backup | NULL<br>Counter number<br>Scale number<br>NULL<br>NULL<br>NULL |
| DST: Dest | 1 byte | Destination (0.1-63) *(binary)* *(0 by default)* | | |
| XCHG CR | 1 byte | CR replacement *(binary)* | | |
| LON | 2 bytes | Data length *(binary)* | | |
| IDF | 3 bytes | Function identifier *(binary)* | | |

**Destination**: If this field shows 0, communication is carried out on the scale connected to the series channel.

If the field shows a number other than 0, communication is carried out on the scale indicated, through the scale connected to the series channel. That is, the connected scale resends the order to the scale indicated in this field.

**XCHG CR Field**

This field shows the bytes used to replace the control character 0x0D *(EOM, end of message).* Control character EOM should only appear at the end of the message, and should therefore be searched in the fields between STX and CRC (both included). The value of this field cannot be either 0 *(NULL)* or 0x0D *(EOM)*. The replacement byte will be applied between the fields **LON** and **DATA** (both included).

### 8.1.2 Body

The body of the message depends on the message's function identifier, i.e. either a message about database access or other scale services.
The body of the message is described in detail in following chapters.

### 8.1.3 End of message

| Field | Length | Description |
|---|---|---|
| CRC | 2 bytes | CRC for the whole message *(binary)* |
| EOM | 1 byte | 13d, 0Dh *(when end of message occurs)* |

### 8.1.4 CRC Field

The CRC is calculated from the first byte **(STX)** until the end of the **DATA** field (message length less than 3 bytes). It should be carried out on the original message, without replacing the 0x0D (EOM) characters.

The following points should be taken into account when calculating the CRC:

- The XCHG CR field should be 0 (NULL).
- The CRC should not include either the CRC or EOM fields.
- If either or both of the results bytes correspond to the EOM character, they shall be replaced by the 0x0E character.

*Routine in C to calculate the CRC of 16 bits:*

```
ushort Calc_CRC (uchar *Data, short Size)
{
	ushort r=0, a, b;
	short t;

	for (t=0; t<Size; t++)
	{
		r+=(t*Data[t])+t;
	}

	a=r>>8;   if (a==0x0D) a=0x0E;
	b=r&0xFF; if (b==0x0D) b=0x0E;

	a<<=8;

	return a|b;
}
```

### 8.1.5 Routine of adjusting a message to the binary protocol

This routine calculates the CRC, searches for a character that does not appear on the message and puts in the XCHCR field. It then runs the message and substitutes all the 0x0D=CR values for the value of the XCHCR field.

```
void AdjBinary(char *Data, short Size)
{
	char  DataKey[256];
	short  t;
	ushort Crc, v;
	uchar  Xchg_CR=0;
```

```
    Data[1]=0x7E;  // IDG: Standard binary protocol identifier
    Data[4]=NULL;  // XCHG CR: Replacement of 0x0D character

    Crc=Calc_CRC ((uchar*)Data, (short)(Size-3));  // Calculate CRC

    v=(uchar)(Crc>>8);  Data[Size-2]=v;  // Assemble high CRC part on the message
    v=(uchar)(Crc&0xFF); Data[Size-3]=v;  // Assemble low CRC part on the message

    memset (DataKey, 0, sizeof(DataKey));  // table to localise XCHG CR field

    for (t=0; t<Size-1; t++)
    {
        DataKey[Data[t]]=1;  // Mark the bytes used in the message
    }

    for (t=0; t<256; t++)   // Search the XCHG CR character
    {
        if (!DataKey[t] && t!=0x0D)
        {
            Xchg_CR=(uchar)t;   // Load found character on XCHG CR field
            Data[4]=Xchg_CR;    // Store field on the data array

            for (t=5; t<Size-3; t++)
            {
                // Substitute EOM character for  XCHG CR field in the whole message.
                if (Data[t]==0x0D) Data[t]=Xchg_CR;
            }

            return;
        }
    }

    // Error of message coherence!
}
```

# 9. ACCESS TO THE DATABASE ENGINE (IDF:000001)

## 9.1 STRUCTURE OF AN ACCESS MESSAGE TO THE ENGINE

| Field | Length | Description |
|---|---|---|
| FUN – RSP | 1 byte | Function or reply to function |
| MET | 2 bytes | Method |
| FUNMASK | 1 byte | Always 0 |
| DBID | 1 byte | Database identifier |
| KEY | 1 byte | Main-access key |
| RKEY | 1 byte | Extended-access key |

***RKEY!=0xFF***

| Field | Length | Description |
|---|---|---|
| NDTA | 1 byte | No. of data blocks |
| DTA | ..... | Data blocks |

***RKEY==0xFF***

| Field | Length | Description |
|---|---|---|
| NDTA | 1 byte | No. of data blocks |
| NITEMS | 1 byte | No. of data items |
| ITEMS | NITEMS | Id's to data items |
| DTA | ..... | Data blocks |

**Data blocks *(DTA field)***

| Field | Length | Description |
|---|---|---|
| Index | 2 bytes | Direct index to database contents<br>*Only if the database is self-indexed.* |

| | | |
|---|---|---|
| Data | ..... | Data stored consecutively, according to the structure described in the section ***"field data"***.<br>Length will depend on the key used and is variable if this is 0xFF. |

### 9.1.1 Function - Reply field

This field is used for different purposes, depending on whether the message is a function message or a reply message. Also, in the case of an error reply, the error code is indicated in this field.
The various functions and replies are shown below.

**Field as Function:**

| Function | Value | Description |
|---|---|---|
| CONSULT | 0x00 | Carry out a consultation |
| PROGRAMME | 0x03 | Inset or modify an entry |
| CLEAR | 0x06 | Mark an entry as cleared |

**Field as Reply:**

| Function | Value | Description |
|---|---|---|
| CONSULT_OK | 0x01 | OK reply to CONSULT function |
| CONSULT_KO | 0x02 | KO reply to CONSULT function |
| PROGRAMME_OK | 0x04 | OK reply to PROGRAMME function |
| PROGRAMME_KO | 0x05 | KO reply to PROGRAMME function |
| CLEAR_OK | 0x07 | OK reply to CLEAR function |
| CLEAR_KO | 0x08 | KO reply to CLEAR function |
| CLEAR ALL_OK | 0x0A | OK reply to CLEAR_ALL function |
| CLEAR ALL_KO | 0x0B | KO reply to CLEAR_ALL function |
| ACCUMULATE_OK | 0x0D | OK reply to ACCUMULATE function |
| ACCUMULATE_KO | 0x0E | KO reply to ACCUMULATE function |
| MARK_OK | 0x10 | OK reply to MARK function |
| MARK_KO | 0x11 | KO reply to MARK function |
| UNMARK_OK | 0x13 | OK reply to UNMARK function |
| UNMARK_KO | 0x14 | KO reply to UNMARK function |
| ADD_OK | 0x16 | OK reply to ADD function |
| ADD_KO | 0x17 | KO reply to ADD function |
| SUBTRACT_OK | 0x19 | OK reply to SUBTRACT function |
| SUBTRACT_KO | 0x1A | KO reply to SUBTRACT function |

**Field as Error Code:**

| Error codes | Value | Description |
|---|---|---|
| ERR_DST | 0x64 | Incorrect destination |
| ERR_ERROR | 0x65 | Non-identified error |
| ERR_MSG_KO | 0x66 | Function not recognised |
| ERR_DATA | 0x67 | Erroneous data |
| ERR_IDF | 0x68 | Unknown function identifier |
| ERR_TIMEOUT | 0x69 | Timeout |

### 9.1.2 Method field

We can use the MET field to determine our position within the database on the register in which we want to work. We should therefore combine the MET field (the condition which should be complied with) with the KEY field (the items that should comply with the condition).
This field can take the following values:

| Method | Value | Description |
|---|---|---|
| NULL | 0x0000 | Null method |
| First-level search methods *(can be combined with second-level search methods)* | | |
| SAME | 0x0001 | Search same as |
| MORE | 0x0002 | Search more than |

| | | |
|---|---|---|
| MORESAME | 0x0003 | Search more than or same as |
| LESS | 0x0004 | Search less than |
| LESSSAME | 0x0005 | Search less than or same as |
| FIRST | 0x0006 | Search by first or smallest |
| LAST | 0x0007 | Search by last or biggest |
| SIMILAR | 0x0008 | Search by most similar |
| RKEY_ZERO | 0x0009 | Search entries with extended key with null fields |
| RKEY_NOTZERO | 0x000A | Search entries with extended key with non-null fields |
| Second-level search methods *(can e combined with first-level search methods)* | | |
| L2_ZERO | 0x0010 | Search entries with extended key with null fields |
| L2_NOTZERO | 0x0020 | Search entries with extended key with non-null fields |
| L2_SAME | 0x0030 | Search by same as according to extended key |
| L2_NOTSAME | 0x0040 | Search by different to extended key |
| Flags added to search methods | | |
| GROUP | 0x0200 | Process entries until end of condition |
| MARKED | 0x0400 | Process marked entries |
| NOT_MARKED | 0x0800 | Process non-marked entries |

### 9.1.3 Funmask field

This is envisioned for future uses, in this protocol version it should always be 0.

| Method | Value | Description |
|---|---|---|
| Direct access | 0x80 | Carries out the desired operation directly on the database engine, avoiding the server and therefore losing the network properties of the system. |
| Reserved | 0x40 | Should be 0 |
| Reserved | 0x20 | Should be 0 |
| Reserved | 0x10 | Should be 0 |
| Reserved | 0x08 | Should be 0 |
| Reserved | 0x04 | Should be 0 |
| Reserved | 0x02 | Should be 0 |
| Reserved | 0x01 | Should be 0 |

### 9.1.4 Key field – main-access key

Main access-key identifier. This key usually defines the fields in the main condition. The different keys defined in each database are described in detail in chapter 2.3.

### 9.1.5 RKey field – extended-access or return key

Extended access-key identifier. The identifiers are shown beginning in the section **"Databases"** in the **Id** field. This key can be used to complement the main search, partially modify database entries, consult protected fields, etc. The most common use is programming. This field contains the extended-access key: the key with the database items we want to modify. In consultations, it contains the return key: the key with items we want to consult.

### 9.1.6 Addition of keys

The keys are combined to form a single key, eliminating repeated fields from the results. The adding up is performed by copying the fields of the first key and then adding the fields of the second, taking into account that they are not repeated, as shown on the table below:

**Example:**
Adding together the *ID_ART_SEARCH_PLUMOS* and *ID_ART_UPDATE_STDPLU* keys from the item database:

| Main key | Extended key | Adding-up key |
|---|---|---|
| 2 → Counter code | 0 → Code | 2 → Counter code |
| 1 → PLU number | 1 → PLU number | 1 → PLU number |
| | 2 → Counter code | 0 → Code |
| | 3 → Family | 3 → Family |
| | 6 → Form of sale | 6 → Form of sale |
| | 4 → Price | 4 → Price |
| | 5 → Descriptive | 5 → Descriptive |
| | 9 → Tare | 9 → Tare |

### 9.1.7 NDATA field

The value of the NDATA field is always set at '1' as we can only access a single position on the database at the moment.

### 9.1.8 Data field

Data are stored sequentially, following the pattern obtained from the access keys, which is determined by the value of the extended key or the sum of its contents. The structure of each field is determined by the type of element with which it is associated in the database framework, defined in detail in chapter 2.3.

**Data size according to data type:**

| Type | Length | Description |
|---|---|---|
| BOOL | 1 byte | Boolean field |
| BIT | (1–8) 1 byte<br>(9-16) 2 bytes<br>(17-24) 3 bytes<br>(25-32) 4 bytes | Bits field with signal |
| UBIT | (1–8) 1 byte<br>(9-16) 2 bytes<br>(17-24) 3 bytes<br>(25-32) 4 bytes | Bits field without signal |
| CHAR or BYTE | 1 byte | 1-byte field with signal |
| UCHAR or UBYTE | 1 byte | 1-byte field without signal |
| SHORT or WORD | 2 bytes | 2-byte field with signal |
| USHORT or UWORD | 2 bytes | 2-byte field without signal |
| TRIBYTE | 3 bytes | 3-byte field with signal |
| UTRIBYTE | 3 bytes | 3-byte field without signal |
| LONG or DWORD | 4 bytes | 4-byte field with signal |
| ULONG or UDWORD | 4 bytes | 4-byte field without signal |
| STRING | 2 bytes + StrSize | Memo or String field<br>The first two bytes indicate the string length, followed by two data blocks. The whole string is sent just as it appears on the database, with nulls following to fill up the string size. |
| POINTER | That of the pointer field | Pointer field to another field via a LINK. |

## 9.2 Examples

### 9.2.1 Programming database elements:

The first thing we should do to programme a position on the database is determine the position we want. We can use the KEY field or main-access key to help us as, combined with the MET field, they will give us the position we want to programme. Most databases have a main key for a single item, which is usually a code (customer code, item code, vendor code, etc.) so that, if we want to programme a position, we should enter the key that only contains the item code in the KEY field, and the SAME AS method (0x01) in the MET field.
If we want to insert a new element in the database, we should always use the extended key RKEY=0xFF=KEYALL, indicating that all the items should be programmed, while to modify an already existing position we can use a key that only contains the items we want to modify as RKEY.

### 9.2.1.1 Programming all items:

To programme all the items of a database position, we should enter the value 0xFF (KEY ALL) in the RKEY or extended-key field. This will be necessary whenever we programme a new element in a database.
We then put the number of database items in the NITEMS field.

**Table showing databases, their identifiers and the number of items in each.**

| DATABASE | IBD | NITEMS |
|---|---|---|
| Customers | 1 | 2 |
| Countries | 2 | 2 |
| Price changes | 3 | 5 |
| Ingredients | 4 | 2 |
| Advertising | 5 | 1 |
| Coded tares | 6 | 2 |
| Goods | 7 | 4 |
| VATs | 8 | 1 |
| Items + Totals | 9 | 37 * |
| Descriptives | 10 | 1 |
| Texts | 11 | 2 |
| Bar codes | 12 | 1 |
| Vendors + Totals | 13 | 24 * |
| Vendor keys | 14 | 3 |
| Orders | 16 | 4 |
| Bovine | 20 | 10 |

**In the case of Items, we do not programme them all, only those that refer to the item (15 or 25).*
*Items : 15 referring to Item + 10 Descriptives + 12 referring to Totals (protected).*
**The case of Vendors is similar to that of Items. We have 3 items referring to the vendor + 21 referring to Totals (protected).*
We then put the item identifiers we want to program. In most databases, we will put them all.

**EXAMPLE:**

**An order database with 4 items:**

| Field | Id | Mode | Type | Size |
|---|---|---|---|---|
| Code | 0 | Normal | ULONG | 4 bytes |
| Order | 1 | Normal | LONG | 4 bytes |
| Form of sale | 2 | Normal | UBIT | 1 bit |
| Descriptive | 3 | Normal | STRING | 20 bytes |

(See chapter 2.3.15)
Item = 1 that corresponds to the Item Code will be the Main Access Code and will be Key 1 (ID_ORDER_INSERT) of this database, which only contains the Item Code.
As we are programming all the elements, the extended key will be 0xFF.

NOTE: We would usually consult an item order before programming it to know the form of sale and descriptive of the item so they are the same in the order database and the item database.

**The message we would send to programme the order would be as follows:**

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, this will be a byte that does not appear in the message |
| LON | ??? | Calculated at the end, this will be the total length of the message |
| IDF | 0x010000 | Id for access to database engine protocol |
| | | |

| | | |
|---|---|---|
| FUN – RSP | 0x03 | PROGRAMME function |
| MET | 0x0100 | SAME AS method |
| FUNMASK | 0x00 | |
| DBID | 0x10 | Identifier of ORDER database = 16 |
| KEY | 0x01 | Main-access key ID_ORDER_INSERT |
| RKEY | 0xFF | Extended-access key |
| NDATA | 0x1 | We should programme a single position on the database |
| NITEMS | 0x04 | 4 Items |
| ITEM | 0x00 | ITEM Item code |
| ITEM | 0x01 | ITEM Order quantity |
| ITEM | 0x02 | ITEM Form of sale |
| ITEM | 0x03 | ITEM Descriptive |
| DATO | 0x01000000 | Datum corresponding to item code (4Bytes) |
| DATO | 0xD0070000 | Datum corresponding to order =2000 (4Bytes) |
| DATO | 0x00 | Datum corresponding to form of sale (1Byte) |
| DATO | 0x1400+"Potatoes….." | Datum corresponding to descriptive item (2+20Bytes) |
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

If we wanted to programme the item database, we could put the RKEY 0xFF and, depending on whether we wanted to programme the item or some type of total, we would put the set of items we wanted. We could also do it by putting a key that contains the items we wanted to programme in the RKEY, as described below. The same thing applies with vendors.

### 9.2.1.2 Programming a set of items:

Each database has a set of keys, in which each key contains a different set of items from the database. Most databases have few items (as a maximum, the 5 of the BD of Price Changes), which makes it is easy to programme them simultaneously, although we can also use keys to programme a single set of items if we want. However, in the case of items or vendors, the keys that only contain the items we want to programme will be very useful.
When we put the extended key in the RKEY with the items we want to programme, it is no longer necessary to indicate the number of items nor the items that we are going to introduce later, as this information is implicit in extended-key form, so that we simply enter data after the RKEY. For example:

**EXAMPLE**
Imagine we want to programme an item and we do not need to programme the item totals. We can use the key ID_ART_UPDATE_ART defined with the items corresponding to the item *(such as Code, PLU, Counter, Family, Form of Sale, VAT, Price, Descriptive, Tare, Expiry date, Best-before date, Label, Pre-selection no., Bar code, Bar-code format)*. (See chapter 2.3.9)

**Imagine we want to programme the next item:**

| Field | Contents |
|---|---|
| Code | 1234 |
| PLU number | 1234 |
| Counter code | 1 |
| Family | 20 |
| Price | 100 |
| Descriptive | "Potatoes" |
| Form of sale | 0 NOT WEIGHTED |
| Expiry date | 0 |
| Best-before date | 0 |
| Tare | 0 |
| Label number | 0xFF BY DEFAULT |
| VAT type | 0 |
| Bar-code format | 0 |
| Bar code | "" |
| Pre-selected parcel no. | 0 |

The order of the data should be the same as that described in the Key that we use as the extended key, which is not the same as the item-definition order.
We should use the key ID_ART_INSERT_COD, which is key 1 (see chapter 2.3.9), as the main-access key.

### The message to programme the item would be as follows:

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, this will be a byte that does not appear in the message |
| LON | ??? | Calculated at the end, this will be the total length of the message |
| IDF | 0x010000 | Id for access to database engine protocol |
| | | |
| FUN – RSP | 0x03 | PROGRAMME function |
| MET | 0x0100 | SAME AS method |
| FUNMASK | 0x00 | |
| DBID | 0x09 | ITEM database identifier |
| KEY | 0x01 | Main-access key ID_ART_INSERT_COD (1) |
| RKEY | 0x15 | Extended-access key ID_ART_UPDATE_ART (21) |
| NDATA | 0x1 | Programming a single position on the database |
| DATO | 0xD2040000 | Datum corresponding to the item Item code (4Bytes) |
| DATO | 0D204 | Datum corresponding to the item PLU (2Bytes) |
| DATO | 0x01 | Datum corresponding to the item Counter(1Byte) |
| DATO | 0x04 | Datum corresponding to the item Family (1Byte) |
| DATO | 0x00 | Datum corresponding to the item Form of sale (1Byte) |
| DATO | 0x00 | Datum corresponding to the item VAT type (1Byte) |
| DATO | 0x640000 | Datum corresponding to the item Price (3Bytes) |
| DATO | 0x1400+"Potatoes….." | Datum corresponding to the item Descriptive (2+20Bytes) |
| DATO | 0x0000 | Datum corresponding to the item Tare (2Bytes) |
| DATO | 0x00 | Datum corresponding to the item Expiry date (1Byte) |
| DATO | 0x00 | Datum corresponding to the item Best-before date (1Byte) |
| DATO | 0xFF | Datum corresponding to the item Label=By default (1Byte) |
| DATO | 0x00 | Datum corresponding to the item Pre-selection (1Bytes) |
| DATO | 0x0D00+"…" | Datum corresponding to the item Bar Code (2+13Bytes) |
| DATO | 0x00 | Datum corresponding to the item Bar-code format (1Byte) |
| | | |
| CRC | ??? | CRC calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

#### 9.2.1.3 Programming an indexed database:

An indexed database is programmed in exactly the same way as in the first example described, with the peculiarity that these databases do not have access codes, but simply the defined code 0x01. For this reason, 0x01 always appears in the main-access code, and 0xFF always appears in the extended-access code, so that (as in the case of the first example) we should specify the number of items and the item identifiers we are going to programme.
Immediately after the item identifiers, and before introducing data, we put two bytes that correspond to the database index, which will point us directly to the element number to be programmed. .
Let's take the example of programming bar codes for packaging mode in Network 1

**EXAMPLE:**
On the bar-code database, we have a single item: the string with the bar-code number. Imagine we want to programme the bar code for the packaging mode in Network 1. This corresponds to index 6 of the database (see chapter 2.3.12).

Imagine that the bar code we want to programme has 4 digits for the weight 'W', 4 digits for the price 'Z', and 4 digits for the amount due 'I'. The resulting bar-code string would be: 'WWWWZZZZIIII' (See appendix 1).

**The message we send to programme this bar code would be as follows:**

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, this will be a byte that does not appear in the message |
| LON | ??? | Calculated at the end, this will be the total length of the message |
| IDF | 0x010000 | Id for access to database engine protocol |
| | | |
| FUN – RSP | 0x03 | PROGRAMME function |
| MET | 0x0100 | SAME AS method |
| FUNMASK | 0x00 | |
| DBID | 0x0C | BAR CODE database identifier = 12 |
| KEY | 0x01 | SELF-INDEXED main-access key |
| RKEY | 0xFF | Extended-access key |
| NDATA | 0x1 | Programming a single position on the database |
| NITEMS | 0x01 | 1 Item |
| ITEM | 0x00 | ITEM Bar code |
| INDICE | 0x06 | INDEX = 6 (seventh position on the database) |
| DATO | 0x0D00+"WWWWZZZZIIII" | Datum corresponding to bar code (2+13Bytes) |
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

### 9.2.1.4 Example of programming a complete item:

***Item data:***

| Field | Contents |
|---|---|
| Code | 1234 |
| PLU number | 1234 |
| Counter code | 1 |
| Family | 20 |
| Price | 100 |
| Descriptive | "Potatoes" |
| Form of sale | 0 |
| Expiry date | 0 |
| Best-before date | 0 |
| Tare | 0 |
| Label number | 0xFF |
| VAT type | 0 |
| Bar-code format | 0 |
| Bar code | "" |
| Pre-selected parcel number | 0 |
| Descriptive 1 | "5-kg bag" |
| Descriptive 2 | "White" |
| Descriptive 3 | "" |
| Descriptive 4 | "" |
| Descriptive 5 | "" |
| Descriptive 6 | "" |
| Descriptive 7 | "" |
| Descriptive 8 | "" |
| Descriptive 9 | "" |
| Descriptive 10 | "" |

## Item programming message:

| | Field | Datum | Description |
|---|---|---|---|
| | STX | 0x02 | STX |
| | IDG | 0x7E | Binary protocol identifier |
| | DST | 0x0000 | Destination *(to the connected equipment)* |
| | XCHG CR | ??? | CR replacement |
| | LON | ??? | Data length |
| | IDF | 0x010000 | Id for access to database engine protocol |
| | | | |
| | FUN | 0x03 | PROGRAMME function |
| | MET | 0x0100 | SAME AS method |
| | FUNMASK | 0x00 | Special functions |
| | DBID | 0x09 | Item database |
| | KEY | 0x01 | Insertion key *(contains the code)* |
| | RKEY | 0xFF | Extended key – access to all fields |
| | NDTA | 0x01 | 1 data block |
| **?** | NITEMS | 0x19 ó 0x09 | Number of items |
| | ID ITEMS | 0x00 | “Code” field identifier |
| | ID ITEMS | 0x01 | “PLU number” field identifier |
| | ID ITEMS | 0x02 | “Counter number” field identifier |
| | ID ITEMS | 0x03 | “Family Number” field identifier |
| | ID ITEMS | 0x04 | “Price” field identifier |
| | ID ITEMS | 0x05 | “Descriptive” field identifier |

## Item programming message (continued):

| | | | |
|---|---|---|---|
| | ID ITEMS | 0x06 | “Form of sale” field identifier |
| | ID ITEMS | 0x07 | “Expiry date” field identifier |
| | ID ITEMS | 0x08 | “Best-before date” field identifier |
| | ID ITEMS | 0x09 | “Tare” field identifier “ |
| | ID ITEMS | 0x0A | “Label number” field identifier |
| | ID ITEMS | 0x0B | “VAT type” field identifier |
| | ID ITEMS | 0x0C | “Bar-code format” field identifier |
| | ID ITEMS | 0x0D | “Bar code” field identifier |
| | ID ITEMS | 0x0E | “Pre-selected parcel number” field identifier |
| | ID ITEMS | 0x25 | “Descriptive 1” field identifier |
| | ID ITEMS | 0x26 | Descriptive 2” field identifier “ |
| | ID ITEMS | 0x27 | “Descriptive 3” field identifier |
| | ID ITEMS | 0x28 | “Descriptive 4” field identifier |
| | ID ITEMS | 0x29 | “Descriptive 5” field identifier |
| | ID ITEMS | 0x2A | “Descriptive 6” field identifier |
| | ID ITEMS | 0x2B | “Descriptive 7” field identifier |
| | ID ITEMS | 0x2C | Descriptive 8” field identifier “ |
| | ID ITEMS | 0x2D | “Descriptive 9” field identifier |
| | ID ITEMS | 0x2E | “Descriptive 10” field identifier |
| | ITEMS | 0xD2040000 | Code |
| | ITEMS | 0xD204 | PLU number |
| | ITEMS | 0x01 | Counter code |
| | ITEMS | 0x14 | Family |
| | ITEMS | 0x640000 | Price |
| **X** | ITEMS | 0x1400 + “Potatoes...” | Descriptive |
| | ITEMS | 0x00 | Form of sale |
| | ITEMS | 0x0000 | Expiry date |
| | ITEMS | 0x0000 | Best-before date |
| | ITEMS | 0x0000 | Tare |
| | ITEMS | 0xFF | Label number |
| | ITEMS | 0x00 | VAT type |
| | ITEMS | 0x00 | Bar-code format |

| | | | |
|---|---|---|---|
| **X** | ITEMS | 0x0D00 + "..." | Bar code |
| | ITEMS | 0x00 | Pre-selected parcel number |
| **X** | ITEMS | 0x1400 + "5-kg bag..." | Descriptive 1 |
| **X** | ITEMS | 0x1400 + "White..." | Descriptive 2 |
| **X** | ITEMS | 0x1400 + "..." | Descriptive 3 |
| **X** | ITEMS | 0x1400 + "..." | Descriptive 4 |
| **X** | ITEMS | 0x1400 + "..." | Descriptive 5 |
| **X** | ITEMS | 0x1400 + "..." | Descriptive 6 |
| **X** | ITEMS | 0x1400 + "..." | Descriptive 7 |
| **X** | ITEMS | 0x1400 + "..." | Descriptive 8 |
| **X** | ITEMS | 0x1400 + "..." | Descriptive 9 |
| **X** | ITEMS | 0x1400 + "..." | Descriptive 10 |
| | | | |
| | CRC | ??? | CRC |
| | EOM | 0x0D | EOM |

***Legend of the previous table:***

| | |
|---|---|
| | Elements that don't need to be programmed |
| **?** | The contents of this field depend on the message elements |
| **X** | String-type fields<br>· The first 2 bytes indicate the length of the previous chain<br>· Dots indicate the chain should be completed to the length indicated |

## 9.2.2 Reply message to a programming operation:

The reply message tells us the result of the programming operation and appears in the RESP field. Replies are usually 0x04 *(Programming OK)*, or 0x05 *(Programming NOK)*, unless the programming message was structured incorrectly and we got some other type of error in reply.
The MET and FUNMASK fields have no significance whatsoever and are usually set at 0.
The DBIBD has the database identifier we have programmed.
If the programming was not correct, the values of the other fields would have no significance. If the programming was correct, the insertion key we have used responds in the KEY field, and should be the same value as that we put in the programming message. The same thing occurs with the RKEY field.
The NDATA field will contain 0x00, indicating that there are no data.

### The reply message to the above example would be as follows:

| Field | Datum, | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x07E | Binary protocol identifier |
| DST | 0x0000 | Destination |
| XCHG CR | ??? | CR replacement |
| LON | ??? | Data length |
| IDF | 0x010000 | Id for access to database engine protocol |
| | | |
| RESP | 0x04 | Programming OK |
| MET | 0x0000 | NULL |
| FUNMASK | 0x00 | NULL |
| DBID | 0x09 | Item database |
| KEY | 0x01 | Insertion key |
| RKEY | 0xFF | Extended key – access to all fields |
| NDTA | 0x00 | 0 data blocks |
| | | |
| CRC | ??? | CRC |
| EOM | 0x0D | EOM |

### 9.2.3 Consulting database elements:

Consulting database elements is performed in exactly the same way as programming. That is, with the KEY and MET fields we can choose the database position we want to consult. With the RKEY Return Key we can select the database items we want to consult.

#### 9.2.3.1 Consulting all items RKEY=KEYALL:

As with programming, if we want to consult all the items of a database position, we put the value 0xFF in the RKEY field, indicating that we want to consult all the items of a selected position *(0xFF=KEYALL=All)*.
We then put the number of items and their identifiers that belong to the main-access key, followed by data (it will only be necessary to put the data that correspond to items that belong to the main-access key).

**NOTE:** A consultation with KEY_ALL gives us all the elements that are physically found on the database in reply. Elements that are links to other databases, such as item descriptives, should be consulted with an RKEY that contains these elements (e.g., ID_ART_UPDATE_DESCR).
Elements stored as "PROTECTED" on the database cannot be consulted with KEYALL.

Let's take an example:

**Example:**

Image we want to consult an order for item 2.The order database contains 4 items: item code, order, form of sale and descriptive *(see chapter 2.3.15)*.
We will use ID_ORDER_INSERT (0x01) as the main-access code, which only contains the item code.

**The message we send would be as follows:**

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, will be a byte that does not appear on the message |
| LON | ??? | Calculated at the end, will be the total length of the message |
| IDF | 0x010000 | Id for the database engine protocol |
| | | |
| FUN – RSP | 0x00 | CONSULTATION function |
| MET | 0x0100 | SAME AS method |
| FUNMASK | 0x00 | |
| DBID | 0x10 | ORDER database identifier = 16 |
| KEY | 0x01 | Main-access key ID_ORDER_INSER |
| RKEY | 0xFF | Extended-access key KEYALL |
| NDATA | 0x1 | Consult a single position on the database |
| NITEMS | 0x01 | 1 Item |
| ITEM | 0x00 | ITEM Item code |
| DATO | 0x02000000 | Datum corresponding to consulted item code (4bytes) |
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

**Reply:**
When carrying out a consultation with the Return key KEYALL, our result would be a message with all the database items that have a value other than NULL. That is, if we consult an item for example, it will only give us the items it has programmed with values other than null. To know which items these are, we have the field NITEMS, and the list of items with their identifiers that respond, so that by knowing these identifiers we can know the item-reply order and the size of each item for gathering the data. Items do not have to follow any particular order, and for that reason gathering data is not insignificant. If the consultation is carried out with a return key other than KEYALL, we will not get the number of items nor the list of item identifiers, but we will get all the data that correspond to the items belonging to that key (even if they are null) and the order specified by the key, which helps us greatly in gathering results.
.

**A possible reply to the previous consultation would be as follows:**

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, will be a byte that does not appear on the message |
| LON | ??? | Calculated at the end, will be the total length of the message |
| IDF | 0x010000 | Id for the database engine protocol |
| | | |
| FUN – RSP | 0x01 | Reply CONSULTATION_OK |
| MET | 0x0100 | SAME AS method |
| FUNMASK | 0x00 | |
| DBID | 0x10 | ORDER database identifier = 16 |
| KEY | 0x01 | Main-access key ID_ORDER_INSER |
| NDATA | 0x1 | Gives us a single position on the database |
| RKEY | 0xFF | Extended-access key KEYALL |
| NITEMS | 0x04 | 4 Items |
| ITEM | 0x00 | ITEM Item code |
| ITEM | 0x01 | ITEM Order |
| ITEM | 0x02 | ITEM Form of sale |
| ITEM | 0x03 | ITEM Item descriptive |
| DATO | 0x02000000 | Datum corresponding to consulted item code (4Bytes) |
| DATO | 0x0A000000 | Datum corresponding to order (4Bytes) |
| DATO | 0x00 | Datum corresponding to Form of sale (weighted) (1Byte) |
| DATO | 0x1400+"Potatoes…" | Datum corresponding to item descriptive (2+20Bytes) |
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

Interpretation of the result:

Item code: 2
Order: 10
Form of sale: Weighted
Descriptive: Potatoes

### 9.2.3.2 Consulting a set of items:

As with programming, if we want to consult a set of items of a position on the database, we should put the value that corresponds to a key containing only the items we want to consult in the RKEY field.
We then put the number of items and their identifiers that belong to the main-access key (KEY field), followed by the data that correspond to these items.

Let's take an example:

**Example:**

Imagine we only want to consult the vendor number with code 2, which is found on the vendor database. We should use the key ID_INSERT_VND (0x01), which only contains the vendor code, as the main-access key (KEY); .the key ID_UPDATE_VENDOR (0x0B), which contains the vendor code (item 1) as the return key (RKEY), and the vendor name (Item 0) in this order. *(See chapter 2.3.13)*

**The message we send will be as follows:**

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, will be a byte that does not appear on the message |
| LON | ??? | Calculated at the end, will be the total length of the message |
| IDF | 0x010000 | Id for the database engine protocol |

| | | |
|---|---|---|
| | | |
| FUN – RSP | 0x00 | CONSULTATION function |
| MET | 0x0100 | SAME AS method |
| FUNMASK | 0x00 | |
| DBID | 0x13 | VENDORS database identifier = 13 |
| KEY | 0x01 | Main-access key ID_ INSERT_VND |
| RKEY | 0x0B | Extended-access key ID_UPDATE_VENDOR |
| NDATA | 0x1 | Consult a single position on the database |
| NITEMS | 0x01 | 1 Item |
| ITEM | 0x00 | ITEM Vendor code |
| DATO | 0x02000000 | Datum corresponding to consulted vendor code (4 Bytes) |
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

**Reply:**

### A possible reply to the previous consultation would be as follows:

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, will be a byte that does not appear on the message |
| LON | ??? | Calculated at the end, will be the total length of the message |
| IDF | 0x010000 | Id for the database engine protocol |
| | | |
| FUN – RSP | 0x01 | Reply CONSULTATATION_OK |
| MET | 0x0100 | SAME AS method |
| FUNMASK | 0x00 | |
| DBID | 0x13 | VENDORS database identifier = 13 |
| KEY | 0x01 | Main-access key ID_ INSERT_VND |
| RKEY | 0x0B | Extended-access key ID_UPDATE_VENDOR |
| NDATA | 0x1 | Returns a single position on the database |
| DATO | 0x02000000 | Datum corresponding to consulted vendor code (4 Bytes) |
| DATO | 0x02+"Pedro Díaz" | Datum corresponding to vendor name (2+20Bytes) |
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

In this case, extracting results is easier because we do not have the items number or the list of item identifiers, but have direct access to the data corresponding to the items that belong to the Return Key.

Interpretation of the result:

Vendor: 2
Name: Pedro Díaz

#### 9.2.3.3 Consulting item totals.

We will endeavour to consult the following item with NON-null totals. The first item is obtained by requesting it, beginning from code 0.
All the totals are requested: totals in the network (sale and packaging) should be requested from the serving scale, while local totals should be requested from each scale.

### The message we send will be as follows:

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |

| | | |
|---|---|---|
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, will be a byte that does not appear on the message |
| LON | ??? | Calculated at the end, will be the total length of the message |
| IDF | 0x010000 | Id for the database engine protocol |
| | | |
| FUN – RSP | 0x00 | CONSULTATION function |
| MET | 0x0022 | MORE THAN + L2_NOTZERO method |
| FUNMASK | 0x00 | |
| DBID | 0x09 | ITEM database identifier |
| KEY | 0x01 | Main-access code ID_ART_INSER_COD |
| RKEY | 0x19 | Extended-access code ID_ ART_CLEAR_TOTAL (Totals) |
| NDTA | 0x01 | 1 Item |
| DATOS | 0x00 | Datum corresponding to consulted item code (4 Bytes) |
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

In subsequent requests, the last value obtained from the scale should be used in the DATA field (item code).

**Reply:**

### A possible reply to the previous consultation would be as follows:

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, will be a byte that does not appear on the message |
| LON | ??? | Calculated at the end, will be the total length of the message |
| IDF | 0x010000 | Id for the database engine protocol |
| | | |
| FUN – RSP | 0x00 | CONSULTATION function |
| MET | 0x0022 | MORE THAN + L2_NOTZERO method |
| FUNMASK | 0x00 | |
| DBID | 0x09 | ITEM database identifier |
| KEY | 0x01 | Main-access code ID_ART_INSER_COD |
| RKEY | 0x19 | Extended-access code ID_ ART_CLEAR_TOTAL (Totals) |
| NDTA | 0x01 | 1 Item |
| DATOS | 0x00 | Datum corresponding to consulted item code (4 Bytes) |
| ITEM | 0x01 | ITEM Order |
| ITEM | 0x02 | ITEM Form of sale |
| ITEM | 0x03 | ITEM Item descriptive |
| DATO | 0x16 | Datum corresponding to item code (4 Bytes) |
| DATO | 0x04 | No. operations in sale mode (2 Bytes). |
| DATO | 0x00 | No. units in sale mode (4 Bytes) |
| DATO | 0x638 | Amount due in sale mode (4 Bytes). |
| DATO | 0x992 | Weight in sale mode (4 Bytes) |
| DATO | 0x00 | No. operations in local (2 Bytes). |
| DATO | 0x00 | No. units in local (4 Bytes) |
| DATO | 0x00 | Amount due in local (4 Bytes). |
| DATO | 0x00 | Weight in local (4 Bytes) |
| DATO | 0x00 | No. operations in packaging (2 Bytes). |
| DATO | 0x00 | No. units in packaging (4 Bytes) |

| | | |
|---|---|---|
| DATO | 0x00 | Amount due in packaging (4 Bytes). |
| DATO | 0x00 | Weight in packaging (4 Bytes) |
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

Interpretation of the result:
Item code: 38
Only if there are totals in sale: 4 operations, 2450gr weight and 1592 pesetas.

The type of totals in local is determined by the work mode of the scale (see section STATE OF THE SYSTEM).
The labelling train stores the totals as "sales".

#### 9.2.3.4 Consulting a self-indexed database

Consulting elements of a self-indexed database is similar to programming. We can consult all the items of a RKEY=KEYALL position, or a set of RKEY!=KEYALL items in the same way as shown in previous examples. The only difference is that because it is self-indexed, the main-access key will always be 1 (KEY=0x01), and we should specify the index of the database element we want to consult. We would usually consult all elements, thereby using RKEY=KEYALL, and, as we do not need to add any data, we would put 0 in the NITEMS field and then put the Index.
Let's take an example:

**Example:**

Imagine we want to consult the second heading line, which is Index 2 in the text database. This database is self-indexed and has 2 items: item=0, which is the text, and item=1, which is the type of letter (*see chapter 2.3.11)*.

### The consultation message would be as follows:

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, will be a byte that does not appear on the message |
| LON | ??? | Calculated at the end, will be the total length of the message |
| IDF | 0x010000 | Id for the database engine protocol |
| | | |
| FUN – RSP | 0x00 | CONSULTATION function |
| MET | 0x0100 | SAME AS method |
| FUNMASK | 0x00 | |
| DBID | 0x11 | TEXT database identifier = 11 |
| KEY | 0x01 | Main-access key SELF-INDEXED |
| RKEY | 0xFF | Extended-access key KEYALL |
| NDATA | 0x1 | Consult a single position on the database |
| NITEMS | 0x00 | 0 Items |
| INDICE | 0x02 | Database index (2nd position) |
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

**Reply:**

### A possible reply to the previous consultation would be as follows::

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |

| | | |
|---|---|---|
| XCHG CR | ??? | Calculated at the end, will be a byte that does not appear on the message |
| LON | ??? | Calculated at the end, will be the total length of the message |
| IDF | 0x010000 | Id for the database engine protocol |
| | | |
| FUN – RSP | 0x01 | Reply CONSULTATATION_OK |
| MET | 0x0100 | SAME AS method |
| FUNMASK | 0x00 | |
| DBID | 0x11 | TEXT database identifier = 11 |
| KEY | 0x01 | Main-access key SELF-INDEXED |
| RKEY | 0xFF | Extended-access key KEYALL |
| NDATA | 0x1 | Returns a single position on the database |
| NITEMS | 0x02 | 2 Items |
| ITEM | 0 | Text |
| ITEM | 1 | Type of letter |
| DATO | 0x1900+"Heading2" | Datum corresponding to text (2+20Bytes) |
| DATO | 0x0A | Datum corresponding to type of letter (1Byte) |
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

Interpretation of the result:

Position 2 of the BD of Text:
Text: Heading2
Type of letter: 10

### 9.2.3.5 Consulting next element

Imagine we want to make a list of a database. We will need to consult the elements one by one and make the list. For this purpose, we first consult the first element with the method field MORESAME, and from then on, the method field MORE.

**Example:**

Imagine we want to consult the next order after item 2. The order database has four items: item code, order, form of sale and descriptive *(see chapter 2.3.15)*.
We will use the key ID_ORDER_INSERT(0x01) as the main-access key, which only contains the item code.

**The message we send will be as follows:**

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, will be a byte that does not appear on the message |
| LON | ??? | Calculated at the end, will be the total length of the message |
| IDF | 0x010000 | Id for the database engine protocol |
| | | |
| FUN – RSP | 0x00 | CONSULTATION function |
| MET | 0x0200 | MORE method |
| FUNMASK | 0x00 | |
| DBID | 0x10 | ORDER database identifier = 16 |
| KEY | 0x01 | Main-access key ID_ORDER_INSER |
| RKEY | 0xFF | Extended-access key KEYALL |
| NDATA | 0x1 | Consult a single position on the database |
| NITEMS | 0x01 | 1 Item |
| ITEM | 0x00 | ITEM Item code |
| DATO | 0x02000000 | Datum corresponding to the consulted item code (4 Bytes) |

| | | |
|---|---|---|
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

**Reply:**

### A possible reply to the previous message would be as follows:

| Field | Contents | Description |
|---|---|---|
| STX | 0x02 | STX |
| IDG | 0x7E | Binary protocol identifier |
| DST | 0x0000 | To the connected equipment |
| XCHG CR | ??? | Calculated at the end, will be a byte that does not appear on the message |
| LON | ??? | Calculated at the end, will be the total length of the message |
| IDF | 0x010000 | Id for the database engine protocol |
| | | |
| FUN – RSP | 0x01 | Reply CONSULTATATION_OK |
| MET | 0x0200 | MORE method |
| FUNMASK | 0x00 | |
| DBID | 0x10 | ORDER database identifier = 16 |
| KEY | 0x01 | Main-access key ID_ORDER_INSER |
| NDATA | 0x1 | Returns a single position on the database |
| RKEY | 0xFF | Extended-access key KEYALL |
| NITEMS | 0x04 | 4 Items |
| ITEM | 0x00 | ITEM Item code |
| ITEM | 0x01 | ITEM Order |
| ITEM | 0x02 | ITEM Form of sale |
| ITEM | 0x03 | ITEM Item descriptive |
| DATO | 0x03000000 | Datum corresponding to item code (4 Bytes) |
| DATO | 0x0F000000 | Datum corresponding to order (4 Bytes) |
| DATO | 0x00 | Datum corresponding to form of sale (Weighted) (1 Byte) |
| DATO | 0x1400+”Pistachios.” | Datum corresponding to item descriptive (2+20Bytes) |
| | | |
| CRC | ??? | CRC Calculated according to the description in chapter 1.2.2 |
| EOM | 0x0D | EOM |

Interpretation of the result:

Item code: 3
Order: 15
Form of sale: Weighted
Descriptive: Pistachios

If we want to implement a correct orders list, we should consult the next order and check it is not zero, as it would therefore not be necessary to print it on the list.

## 9.3 Databases

| Database | Id | Mode | Type of communication |
|---|---|---|---|
| Customers | 1 | Normal | System copy |
| Countries | 2 | Normal | System copy |
| Price changes | 3 | Normal | System copy |
| Ingredients | 4 | Normal | System copy |
| Coded tares | 6 | Normal | System copy |
| Goods | 7 | Self-indexed | System master |
| VATs | 8 | Self-indexed | System copy |
| Items + Totals | 9 | Normal | System copy |
| Descriptives | 10 | Chained | System copy |
| Texts headings & legends | 11 | Self-indexed | Local |

| | | | |
|---|---|---|---|
| Bar codes | 12 | Self-indexed | Local |
| Vendors + Totals | 13 | Normal | System copy |
| Vendor keys | 14 | Normal | System copy |
| Orders | 16 | Normal | System copy |

### 9.3.1 Customers

Maximum number of elements by default: 100
Number of elements pre-defined by the system: 0
Work mode: System copy
Can be updated and self-synchronised
Access mode: Normal

**Basic framework structure:**

| Field | Id | Mode | Type | Accessible range | Size |
|---|---|---|---|---|---|
| Code | 0 | Normal | U3BYTE | (0x0 - 0x1869F) | 3 bytes |
| Name | 1 | Normal | STRING | | 16 bytes |

**Access key:**

| Key | Id | Id field | Fields |
|---|---|---|---|
| IDC_INSERT | 1 | 0 | Code |
| IDC_SEARCH | 2 | 0 | Code |
| IDC_CONSULT | 3 | 0<br>1 | Code<br>Name |

### 9.3.2 Countries:

Maximum number of elements by default: 25
Number of elements pre-defined by the system: 1
Work mode: System copy
Can be updated
Access mode: Normal

**Basic framework structure:**

| Field | Id | Mode | Type | Accessible range | Size |
|---|---|---|---|---|---|
| Code | 0 | Normal | ULONG | (0x0 - 0x63) | 4 bytes |
| Name | 1 | Normal | STRING | | 21 bytes |

**Access key:**

| Key | Id | Id field | Fields |
|---|---|---|---|
| IDPA_CODE | 1 | 0 | Code |
| IDPA_NAME | 2 | 1<br>0 | Name<br>Code |

### 9.3.3 Price changes

Maximum number of elements by default: 100
Number of elements pre-defined by the system: 0
Work mode: System copy
Can be updated and self-synchronised
Access mode: Normal

**Basic framework structure:**

| Field | Id | Mode | Type | Accessible range | Size |
|---|---|---|---|---|---|
| Code | 0 | Normal | ULONG | (0x0 - 0x0F423F) | 4 bytes |
| Price | 2 | Normal | ULONG | (0x0 - 0x0F423F) | 4 bytes |
| Type of change | 1 | Normal | CHAR | (0x0 - 0x3) | 1 byte |
| Initial date | 3 | Normal | ULONG | | 4 bytes |
| Memo | 4 | Normal | MEMO | | 8 bytes |

***Values of the 'type of change' field:***

0x0: Price change
0x1: Price changed today
0x2: Seasonal offer
0x3: Scaled price

***Values of the 'memo' field (depend on the values of the "type of change" field):***

| Type of change | Structure | Type of datum | Size | Accessible range |
|---|---|---|---|---|
| 0x01 | Final date | ULONG | 4 bytes | |
| 0x02 | Weight 1 | SHORT | 2 bytes | Depends weight config. |
| | Price2 | ULONG | 4 bytes | (0x0 - 0x0F423F) |
| | Weight2 | SHORT | 2 bytes | Depends weight config. |

**Access key:**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_PRC_TYPE | 1 | 2<br>0 | Type of change<br>Code |
| ID_PRC_DATAINI | 2 | 2<br>3<br>0 | Type of change<br>Initial date<br>Code |
| ID_PRC_CODE | 3 | 0 | Code |

## 9.3.4 Ingredients:

Number of elements by default: 250
Number of elements pre-defined by the system: 0
Work mode: System copy
Can be updated
Access mode: Normal

**Basic framework structure:**

| Field | Id | Mode | Type | Accessible range | Size |
|---|---|---|---|---|---|
| Code | 0 | Normal | ULONG | (0x0 - 0x0F423F) | 4 bytes |
| Text | 1 | Normal | STRING | | 16 bytes |

**Access key:**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_ING_INSERT | 1 | 0 | Code |
| ID_ING_SEARCH | 2 | 0 | Code |
| ID_ING_CONSULT | 3 | 0<br>1 | Code<br>Text |

### 9.3.5 Coded tares:

Maximum number of elements: 10
Number of elements pre-defined by the system: 0
Work mode: System copy
Can be updated
Access mode: Normal

**Basic framework structure:**

| Field | Id | Mode | Type | Size |
|---|---|---|---|---|
| Code | 0 | Normal | CHAR | 1 byte |
| Tare | 1 | Normal | USHORT | 2 bytes |

**Access key**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_TN_INSERT | 1 | 0 | Code |
| ID_TN_SEARCH | 2 | 1 | Tare |

### 9.3.6 Goods (*Self-indexed)*

Maximum number of elements by default: 30
Number of elements pre-defined by the system: 0
Work mode: System master
Can be updated
Access mode: Self-indexed

**Basic framework structure:**

| Field | Id | Mode | Type | Size |
|---|---|---|---|---|
| Item | 0 | Normal | LONG | 4 bytes |
| Weight | 1 | Normal | LONG | 4 bytes |
| Certified | 2 | Normal | SRTING | 12 bytes |
| Date | 3 | Normal | ULONG | 4 bytes |

### 9.3.7 VATs *(Self-indexed):*

Maximum number of elements by default: 4
Number of elements pre-defined by the system: 1
Work mode: System copy
Can be updated
Access mode: Self-indexed

**Basic framework structure:**

| Field | Id | Mode | Type | Size |
|---|---|---|---|---|
| VAT | 0 | Normal | SHORT | 2 bytes |

### 9.3.8 Items + Totals

Maximum number of elements by default: 5000
Number of elements pre-defined by the system: 1
Work mode: System copy
Can be updated and self-synchronised
Access mode: Normal

**Basic framework structure:**

| Field | Id | Mode | Type | Accessible range | Size |
|---|---|---|---|---|---|
| Code | 0 | Normal | ULONG | (0x0 - 0xF423F) | 4 bytes |
| PLU number | 1 | Normal | USHORT | (0x0 - 0x273F) | 2 bytes |
| Counter code | 2 | Normal | UCHAR | (0x0 - 0x63) | 1 byte |
| Family | 3 | Normal | UCHAR | (0x0 - 0xFF) | 1 byte |
| Price | 4 | Normal | UTRIBYTE | (0x0 - 0xF423F) | 3 bytes |
| Descriptive | 5 | Normal | STRING | | 25(20*) bytes |
| Form of sale | 6 | Normal | UBIT | (0x0=U, 0x1=W) | 1 bit |
| Expiry date | 7 | Normal | UBIT | (0x0 - 0x03E7) | 10 bits |
| Best-before date | 8 | Normal | UBIT | (0x0 - 0x03E7) | 10 bits |
| Tare | 9 | Normal | USHORT | (0x0 - 0x260F) | 2 bytes |
| Label number | 10 | Normal | BIT | (0x0 - 0xA)<br>(0xB➔ No Etq)<br>(0xFF➔Defecto) | 6 bits |
| VAT type | 11 | Normal | UBIT | (0x0 - 0x4) | 3 bits |
| Bar-code format | 12 | Normal | UBIT | (0x0 ➔ EAN13) | 2 bits |
| Bar code | 13 | Normal | STRING | See appendix 1 | 12 bytes |
| Pre-selected parcel number | 14 | Normal | UCHAR | (0x0 - 0x63) | 1 byte |
| Stock | 47 | Protected | ULONG | (0x0 - 0xF423F) | 4 bytes |
| Totals in sale mode | | | | | |
| Number of operations | 15 | Protected | USHORT | | 2 bytes |
| Number of units | 16 | Protected | ULONG | | 4 bytes |
| Amount due | 17 | Protected | LONG | | 4 bytes |

| | | | | | |
|---|---|---|---|---|---|
| Weight | 18 | Protected | ULONG | | 4 bytes |
| Totals in local | | | | | |
| Number of operations | 19 | Protected | USHORT | | 2 bytes |
| Number of units | 20 | Protected | ULONG | | 4 bytes |
| Amount due | 21 | Protected | LONG | | 4 bytes |
| Weight | 22 | Protected | ULONG | | 4 bytes |
| Totals in packaging mode | | | | | |
| Number of operations | 23 | Protected | USHORT | | 2 bytes |
| Number of units | 24 | Protected | ULONG | | 4 bytes |
| Amount due | 25 | Protected | LONG | | 4 bytes |
| Weight | 26 | Protected | ULONG | | 4 bytes |

**Basic framework structure (continued):**

| Field | Id | Mode | Type | Accessible range | Size |
|---|---|---|---|---|---|
| Descriptives | | | | | |
| Descriptive 1 | 37 | Normal | POINTER | | *According to pointer* |
| Descriptive 2 | 38 | Normal | POINTER | | *According to pointer* |
| Descriptive 3 | 39 | Normal | POINTER | | *According to pointer* |
| Descriptive 4 | 40 | Normal | POINTER | | *According to pointer* |
| Descriptive 5 | 41 | Normal | POINTER | | *According to pointer* |
| Descriptive 6 | 42 | Normal | POINTER | | *According to pointer* |
| Descriptive 7 | 43 | Normal | POINTER | | *According to pointer* |
| Descriptive 8 | 44 | Normal | POINTER | | *According to pointer* |
| Descriptive 9 | 45 | Normal | POINTER | | *According to pointer* |
| Descriptive 10 | 46 | Normal | POINTER | | *According to pointer* |
| Descriptive 11 | 58 | Normal | POINTER | | *According to pointer* |
| Descriptive 12 | 59 | Normal | POINTER | | *According to pointer* |
| Descriptive 13 | 60 | Normal | POINTER | | *According to pointer* |
| Descriptive 14 | 61 | Normal | POINTER | | *According to pointer* |
| Descriptive 15 | 62 | Normal | POINTER | | *According to pointer* |
| Descriptive 16 | 63 | Normal | POINTER | | *According to pointer* |
| Descriptive 17 | 64 | Normal | POINTER | | *According to pointer* |
| Descriptive 18 | 65 | Normal | POINTER | | *According to pointer* |
| Descriptive 19 | 66 | Normal | POINTER | | *According to pointer* |
| Descriptive 20 | 67 | Normal | POINTER | | *According to pointer* |

***Note: Only beginning from train version 8.52 and scale version 9.00:**

- **Id 58 to 67**
- **Id 5 with 25 characters (previously 20)**

**Access keys :**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_ART_INSERT_COD | 1 | 0 | Code |
| ID_ART_SEARCH_PLUMOS | 2 | 2<br>1 | Counter code<br>PLU number |
| ID_ART_SEARCH_CODFAM | 3 | 3<br>0 | Family<br>Code |
| ID_ART_UPDATE_STDPLU | 4 | 0<br>1<br>2<br>3<br>6<br>4<br>5<br>9 | Code<br>PLU number<br>Counter code<br>Family<br>Form of sale<br>Price<br>Descriptive<br>Tare |

| | | | |
|---|---|---|---|
| ID_ART_UPDATE_ENVPLU | 5 | 7<br>8<br>10<br>14<br>0 | Expiry date<br>Best-before date<br>Label number<br>Pre-selected parcel number<br>Code |
| ID_ART_UPDATE_CODBAR | 6 | 13<br>12<br>0 | Bar code<br>Bar-code format<br>Code |
| ID_ART_UPDATE_TIVA | 7 | 11<br>0 | VAT type<br>Code |
| ID_ART_UPDATE_PRC | 8 | 4<br>0 | Price<br>Code |
| ID_ART_UPDATE_TOTVNT | 9 | 15<br>16<br>17<br>18 | No. operations in sale mode<br>No. units in sale mode<br>Amount due in sale mode<br>Weight in sale mode |
| ID_ART_UPDATE_TOTLOC | 10 | 19<br>20<br>21<br>22 | No. operations in local<br>No. units in local<br>Amount due in local<br>Weight in local |
| ID_ART_UPDATE_TOTENV | 11 | 23<br>24<br>25<br>26 | No. operations in packaging mode<br>No. units in packaging mode<br>Amount due in packaging mode<br>Weight in packaging mode |
| ID_ART_LISTAR_TOTVNT | 12 | 15<br>16<br>17<br>18<br>5<br>3<br>0 | No. operations in sale mode<br>No. units in sale mode<br>Amount due in sale mode<br>Weight in sale mode<br>Descriptive<br>Family<br>Code |
| ID_ART LISTAR__TOTLOC | 13 | 19<br>20<br>21<br>22<br>5<br>3<br>0 | No. operations in local<br>No. units in local<br>Amount due in local<br>Weight in local<br>Descriptive<br>Family<br>Code |

**Access keys (continued):**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_ART_LISTAR _TOTENV | 14 | 23<br>24<br>25<br>26<br>5<br>3<br>0 | No. operations in packaging mode<br>No. units in packaging mode<br>Amount due in packaging mode<br>Weight in packaging mode<br>Descriptive<br>Family<br>Code |

| | | | |
|---|---|---|---|
| ID_ART_CONSUL_TOTAL | 15 | 15 | No. operations in sale mode |
| | | 16 | No. units in sale mode |
| | | 17 | Amount due in sale mode |
| | | 18 | Weight in sale mode |
| | | 19 | No. operations in local |
| | | 20 | No. units in local |
| | | 21 | Amount due in local |
| | | 22 | Weight in local |
| | | 23 | No. operations in packaging mode |
| | | 24 | No. units in packaging mode |
| | | 25 | Amount due in packaging mode |
| | | 26 | Weight in packaging mode |
| | | 0 | Code |
| ID_ART_UPDATE_DESCR0 | 16 | 37 | Descriptive 1 |
| | | 38 | Descriptive 2 |
| | | 39 | Descriptive 3 |
| | | 40 | Descriptive 4 |
| | | 41 | Descriptive 5 |
| | | 0 | Code |
| ID_ART_UPDATE_DESCR1 | 17 | 42 | Descriptive 6 |
| | | 43 | Descriptive 7 |
| | | 44 | Descriptive 8 |
| | | 45 | Descriptive 9 |
| | | 46 | Descriptive 10 |
| | | 0 | Code |
| ID_ART_UPDATE_DESCR | 18 | 37 | Descriptive 1 |
| | | 38 | Descriptive 2 |
| | | 39 | Descriptive 3 |
| | | 40 | Descriptive 4 |
| | | 41 | Descriptive 5 |
| | | 42 | Descriptive 6 |
| | | 43 | Descriptive 7 |
| | | 44 | Descriptive 8 |
| | | 45 | Descriptive 9 |
| | | 46 | Descriptive 10 |
| | | 0 | Code |

**Access keys (continued):**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_ART_BORRAR_TOTAL | 19 | 15 | No. operations in sale mode |
| | | 16 | No. units in sale mode |
| | | 17 | Amount due in sale mode |
| | | 18 | Weight in sale mode |
| | | 19 | No. operations in local |
| | | 20 | No. units in local |
| | | 21 | Amount due in local |
| | | 22 | Weight in local |
| | | 23 | No. operations in packaging mode |
| | | 24 | No. units in packaging mode |
| | | 25 | Amount due in packaging mode |
| | | 26 | Weight in packaging mode |
| ID_ART_CLEAR_PRC | 20 | 4 | Price |

| | | | |
|---|---|---|---|
| ID_ART_UPDATE_ART | 21 | 0 | Code |
| | | 1 | PLU number |
| | | 2 | Counter number |
| | | 3 | Family |
| | | 6 | Form of sale |
| | | 11 | VAT type |
| | | 4 | Price |
| | | 5 | Descriptive |
| | | 9 | Tare |
| | | 7 | Expiry date |
| | | 8 | Best-before date |
| | | 10 | Label number |
| | | 14 | Pre-selected parcel number |
| | | 13 | Bar code |
| | | 12 | Bar-code format |
| ID_ART_SEARCH_FAM | 22 | 3 | Family |
| ID_FAM_LISTAR_TOTVNT | 23 | 15 | No. operations in sale mode |
| | | 16 | No. units in sale mode |
| | | 17 | Amount due in sale mode |
| | | 18 | Weight in sale mode |
| ID_FAM LISTAR__TOTLOC | 24 | 19 | No. operations in local |
| | | 20 | No. units in local |
| | | 21 | Amount due in local |
| | | 22 | Weight in local |
| ID_FAM_LISTAR _TOTENV | 25 | 23 | No. operations in packaging mode |
| | | 24 | No. units in packaging mode |
| | | 25 | Amount due in packaging mode |
| | | 26 | Weight in packaging mode |
| ID_ART_SEARCH_CODBAR | 26 | 13 | Bar code |
| ID_ART_UPDATE_INVENT | 27 | 47 | Stock |
| ID_ART_CONSUL_INVENT | 28 | 0 | Code |
| | | 47 | Stock |
| | | 6 | Form of sale |
| | | 5 | Descriptive |

**Access keys (continued):**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_ART_CONSPROT | 29 | 0 | Code |
| | | 1 | PLU number |
| | | 2 | Counter code |
| | | 3 | Family |
| | | 4 | Price |
| | | 5 | Descriptive |
| | | 6 | Form of sale |
| | | 7 | Expiry date |
| | | 8 | Best-before date |
| | | 9 | Tare |
| | | 10 | Label number |
| | | 11 | VAT type |
| | | 12 | Bar-code format |
| | | 13 | Bar code |
| | | 14 | Pre-selected parcel number |
| | | 37 | Descriptive 1 |
| | | 38 | Descriptive 2 |
| | | 39 | Descriptive 3 |
| | | 40 | Descriptive 4 |
| | | 41 | Descriptive 5 |
| | | 42 | Descriptive 6 |
| | | 43 | Descriptive 7 |
| | | 44 | Descriptive 8 |
| | | 45 | Descriptive 9 |
| | | 46 | Descriptive 10 |
| ID_ART_SEARCH_MOSCOD | 30 | 2 | Counter code |
| | | 0 | Item code |

**Access keys (continued):**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_ART_UPDATE_DESCRL | 31 | 37 | Descriptive 1 |
| | | 38 | Descriptive 2 |
| | | 39 | Descriptive 3 |
| | | 40 | Descriptive 4 |
| | | 41 | Descriptive 5 |
| | | 42 | Descriptive 6 |
| | | 43 | Descriptive 7 |
| | | 44 | Descriptive 8 |
| | | 45 | Descriptive 9 |
| | | 46 | Descriptive 10 |
| | | 58 | Descriptive 11 |
| | | 59 | Descriptive 11 |
| | | 60 | Descriptive 13 |
| | | 61 | Descriptive 14 |
| | | 62 | Descriptive 15 |
| | | 63 | Descriptive 16 |
| | | 64 | Descriptive 17 |
| | | 65 | Descriptive 18 |
| | | 66 | Descriptive 19 |
| | | 67 | Descriptive 20 |
| | | 0 | Code |

**Access keys (continued):**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_ART_CONSPROTL | 32 | 0 | Code |
| | | 1 | PLU number |
| | | 2 | Counter code |
| | | 3 | Family |
| | | 4 | Price |
| | | 5 | Descriptive |
| | | 6 | Form of sale |
| | | 7 | Expiry date |
| | | 8 | Best-before date |
| | | 9 | Tare |
| | | 10 | Label number |
| | | 11 | VAT type |
| | | 12 | Bar-code format |
| | | 13 | Bar code |
| | | 14 | Pre-selected parcel number |
| | | 37 | Descriptive 1 |
| | | 38 | Descriptive 2 |
| | | 39 | Descriptive 3 |
| | | 40 | Descriptive 4 |
| | | 41 | Descriptive 5 |
| | | 42 | Descriptive 6 |
| | | 43 | Descriptive 7 |
| | | 44 | Descriptive 8 |
| | | 45 | Descriptive 9 |
| | | 46 | Descriptive 10 |
| | | 58 | Descriptive 11 |
| | | 59 | Descriptive 11 |
| | | 60 | Descriptive 13 |
| | | 61 | Descriptive 14 |
| | | 62 | Descriptive 15 |
| | | 63 | Descriptive 16 |
| | | 64 | Descriptive 17 |
| | | 65 | Descriptive 18 |
| | | 66 | Descriptive 19 |
| | | 67 | Descriptive 20 |

### 9.3.9 Descriptives *(Chained):*

Maximum number of elements by default: 20000
Number of elements pre-defined by the system: 0
Work mode: System copy
Cannot be updated *(updating depends on the database to which it is connected)*
Access mode: Chained

**This database can only be accessed through other databases using fields like LINK.**

**Basic framework structure:**

| Field | Id | Mode | Type | Size |
|---|---|---|---|---|
| Descriptive | 0 | Normal | STRING | 25 (20) bytes |

***Note: Only beginning from train version 8.52 and scale version 9.00:**

- **Id 0 with 25 characters (previously 20)**

## 9.3.10 Texts: headings, legends, free text and advertising *(Self-indexed):*

Maximum number of elements by default: 34
Number of elements pre-defined by the system: 0
Work mode: Local
Can be updated
Access mode: Self-indexed

Entries 0 to 4 → Headings
Entries 5 to 9 → Legends
Entries 10 to 19 → Free texts
Entries 20 to 29 → Advertising: specials
Entries 30 to 33 → Advertising: continuous

### Basic framework structure:

| Field | Id | Mode | Type | Size |
|---|---|---|---|---|
| Text | 0 | Normal | STRING | 26 bytes |
| Type of letter | 1 | Normal | UCHAR | 1 byte |

The scale has 80 different types of letter, from 1 to 80, and the value of the byte for each letter type stored on the database is 'Type of letter + 47'.

## 9.3.11 Bar code *(Self-indexed):*

Maximum number of elements by default: 14
Number of elements pre-defined by the system: 0
Work mode: Local
Can be updated
Access mode: Self-indexed

### Basic framework structure:

| Field | Id | Mode | Type | Size |
|---|---|---|---|---|
| Bar code | 0 | Normal | STRING | 13 bytes |

**Bar codes in each position:**

| Index | Bar code |
|---|---|
| 0 | Bar code Sales Mode in Network |
| 1 | Bar code Sales Mode in Local |
| 2 | Bar code Mixed Mode in Network |
| 3 | Bar code Mixed Mode in Local |
| 4 | Bar code Super Mode in Network |
| 5 | Bar code Super Mode in Local |
| 6 | Bar code Packaging Mode in Network 1 |
| 7 | Bar code Packaging Mode in Local 1 |
| 8 | Bar code Packaging Mode in Network 2 |
| 9 | Bar code Packaging Mode in Local 2 |
| 10 | Bar code Total Mode in Network 1 |
| 11 | Bar code Total Mode in Local 1 |
| 12 | Bar code Total Mode in Network 2 |
| 13 | Bar code Total Mode in Local 2 |

The significance of the bytes of the bar-code string is shown in APPENDIX 1.

## 9.3.12 Vendors + Totals:

Maximum number of elements by default: 100
Number of elements pre-defined by the system: 0
Work mode: System copy
Can be updated and self-synchronised
Access mode: Normal

**Basic framework structure:**

| Field | Id | Mode | Type | Size |
|---|---|---|---|---|
| Name | 0 | Normal | STRING | 31 bytes |
| Code | 1 | Normal | ULONG | 4 bytes |
| Drawer | 23 | Normal | LONG | 4 bytes |
| Sale totals | | | | |
| Number of receipts | 2 | Protected | ULONG | 4 bytes |
| Number of operations | 3 | Protected | USHORT | 2 bytes |
| Amount due | 4 | Protected | LONG | 4 bytes |
| Number of negative operations | 5 | Protected | USHORT | 2 bytes |
| Negative amount dues | 6 | Protected | LONG | 4 bytes |
| Number of cancelled operations | 7 | Protected | USHORT | 2 bytes |
| Cancelled amount dues | 8 | Protected | LONG | 4 bytes |
| Local totals | | | | |
| Number of receipts | 9 | Protected | ULONG | 4 bytes |
| Number of operations | 10 | Protected | USHORT | 2 bytes |
| Amount due | 11 | Protected | LONG | 4 bytes |
| Number of negative operations | 12 | Protected | USHORT | 2 bytes |
| Negative amount dues | 13 | Protected | LONG | 4 bytes |
| Number of cancelled operations | 14 | Protected | USHORT | 2 bytes |
| Cancelled amount dues | 15 | Protected | LONG | 4 bytes |
| Packaging totals | | | | |
| Number of receipts | 16 | Protected | ULONG | 4 bytes |
| Number of operations | 17 | Protected | USHORT | 2 bytes |
| Amount due | 18 | Protected | LONG | 4 bytes |
| Number of negative operations | 19 | Protected | USHORT | 2 bytes |
| Negative amount dues | 20 | Protected | LONG | 4 bytes |
| Number of cancelled operations | 21 | Protected | USHORT | 2 bytes |
| Cancelled amount dues | 22 | Protected | LONG | 4 bytes |

**Access keys:**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_INSERT_VND | 1 | 1 | Code |
| ID_BORRAR_TOTAL_VND | 2 | 2 | Receipts in sale mode |
| | | 3 | Operations in sale mode |
| | | 4 | Amount due in sale mode |
| | | 5 | Negative operations in sale mode |
| | | 6 | Negative amount dues in sale mode |
| | | 7 | Cancelled operations in sale mode |
| | | 8 | Cancelled amount dues in sale mode |
| | | 9 | Receipts in local |
| | | 10 | Operations in local |
| | | 11 | Amount due in local |
| | | 12 | Negative operations in local |
| | | 13 | Negative amount dues in local |
| | | 14 | Cancelled operations in local |
| | | 15 | Cancelled amount dues in local |
| | | 16 | Receipts in packaging mode |
| | | 17 | Operations in packaging mode |
| | | 18 | Amount due in packaging mode |
| | | 19 | Negative operations in packaging |
| | | 20 | Negative amount dues in packaging |
| | | 21 | Cancelled operations in packaging |
| | | 22 | Cancelled amount dues in packaging |

| | | | |
|---|---|---|---|
| ID_UPDATE_TOTVNT_VND | 3 | 2<br>3<br>4<br>5<br>6<br>7<br>8 | Receipts in sale mode<br>Operations in sale mode<br>Amount due in sale mode<br>Negative operations in sale mode<br>Negative amount dues in sale mode<br>Cancelled operations in sale mode<br>Cancelled amount dues in sale mode |
| ID_UPDATE_TOTLOC_VND | 4 | 9<br>10<br>11<br>12<br>13<br>14<br>15 | Receipts in local<br>Operations in local<br>Amount due in local<br>Negative operations in local<br>Negative amount dues in local<br>Cancelled operations in local<br>Cancelled amount dues in local |
| ID_UPDATE_TOTENV_VND | 5 | 16<br>17<br>18<br>19<br>20<br>21<br>22 | Receipts in packaging mode<br>Operations in packaging mode<br>Amount due in packaging mode<br>Negative operations in packaging<br>Negative amount dues in packaging<br>Cancelled operations in packaging<br>Cancelled amount dues in packaging |

**Access keys (continued):**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_LISTAR _TOTVNT_VND | 6 | 2<br>3<br>4<br>5<br>6<br>7<br>8<br>1 | Receipts in sale mode<br>Operations in sale mode<br>Amount due in sale mode<br>Negative operations in sale mode<br>Negative amount dues in sale mode<br>Cancelled operations in sale mode<br>Cancelled amount dues in sale mode<br>Code |
| ID_LISTAR _TOTLOC_VND | 7 | 9<br>10<br>11<br>12<br>13<br>14<br>15<br>1 | Receipts in local<br>Operations in local<br>Amount due in local<br>Negative operations in local<br>Negative amount dues in local<br>Cancelled operations in local<br>Cancelled amount dues in local<br>Code |
| ID_LISTAR _TOTENV_VND | 8 | 16<br>17<br>18<br>19<br>20<br>21<br>22<br>1 | Receipts in packaging mode<br>Operations in packaging mode<br>Amount due in packaging mode<br>Negative operations in packaging<br>Negative amount dues in packaging<br>Cancelled operations in packaging<br>Cancelled amount dues in packaging<br>Code |

| | | | |
|---|---|---|---|
| ID_CONSULTAR_TOTAL_VND | 9 | 2 | Receipts in sale mode |
| | | 3 | Operations in sale mode |
| | | 4 | Amount due in sale mode |
| | | 5 | Negative operations in sale mode |
| | | 6 | Negative amount dues in sale mode |
| | | 7 | Cancelled operations in sale mode |
| | | 8 | Cancelled amount dues in sale mode |
| | | 9 | Receipts in local |
| | | 10 | Operations in local |
| | | 11 | Amount due in local |
| | | 12 | Negative operations in local |
| | | 13 | Negative amount dues in local |
| | | 14 | Cancelled operations in local |
| | | 15 | Cancelled amount dues in local |
| | | 16 | Receipts in packaging mode |
| | | 17 | Operations in packaging mode |
| | | 18 | Amount due in packaging mode |
| | | 19 | Negative operations in packaging |
| | | 20 | Negative amount dues in packaging |
| | | 21 | Cancelled operations in packaging |
| | | 22 | Cancelled amount dues in packaging |
| | | 1 | Code |
| ID_UPDATE_CAJON | 10 | 1 | Code |
| | | 23 | Drawer |
| ID_UPDATE_VENDEDOR | 11 | 1 | Code |
| | | 0 | Name |

### 9.3.13 Vendor keys:

Maximum number of elements: 140
Number of elements pre-defined by the system: 0
Work mode: System copy
Can be updated.
Access mode: Normal

**Basic framework structure:**

| Field | Id | Mode | Type | Size |
|---|---|---|---|---|
| Counter code | 0 | Normal | CHAR | 1 byte |
| Vendor code | 1 | Normal | ULONG | 4 bytes |
| Current key | 2 | Normal | CHAR | 1 byte |

**Access key:**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_SEARCH_TECLAMOS | 1 | 0 | Counter code |
| | | 2 | Current key |
| ID_SEARCH_CODEVND | 2 | 1 | Vendor code |
| ID_INSERT_TECLASVND | 3 | 0 | Counter code |
| | | 2 | Current key |
| | | 1 | Vendor code |
| ID_SEARCH_CUSTOMEREPM | 4 | 2 | Current key |

### 9.3.14 Orders:

Maximum number of elements by default: 100
Number of elements pre-defined by the system: 0
Work mode: System copy
Can be updated and self-synchronised
Access mode: Normal

**Basic framework structure:**

| Field | Id | Mode | Type | Size |
|---|---|---|---|---|
| Code | 0 | Normal | ULONG | 4 bytes |
| Order | 1 | Normal | LONG | 4 bytes |
| Form of sale | 2 | Normal | UBIT | 1 byte |
| Descriptive | 3 | Normal | STRING | 20 bytes |

**Access key:**

| Key | Id | Id field | Fields |
|---|---|---|---|
| ID_PEDIDO_INSERT | 1 | 0 | Code |
| ID_UPDATE_PEDIDO | 2 | 0<br>1 | Code<br>Order |

## 9.3.15 Bovine *(self-indexed):*

Maximum number of elements by default: 5
Number of elements pre-defined by the system: 1
Work mode: System copy
Can be updated
Access mode: Self-indexed

**Basic framework structure:**

| Field | Id | Mode | Type | Size |
|---|---|---|---|---|
| Animal identifier | 1 | Normal | STRING | 10 bytes |
| Country-of-origin code | 2 | Normal | USHORT | 2 bytes |
| Country-of-breeding code | 3 | Normal | USHORT | 2 bytes |
| Country-of-slaughter code | 4 | Normal | USHORT | 2 bytes |
| Country-of-quartering code | 5 | Normal | USHORT | 2 bytes |
| Country-of-production code | 6 | Normal | USHORT | 2 bytes |
| RSI slaughterhouse | 7 | Normal | STRING | 10 bytes |
| RSI quartering facility | 8 | Normal | STRING | 10 bytes |
| Category | 9 | Normal | CHAR | 1 byte |
| Race | 10 | Normal | CHAR | 1 byte |
| Totals in sale mode | | | | |
| Number of operations | 11 | Protected | USHORT | 2 bytes |
| Number of units | 12 | Protected | ULONG | 4 bytes |
| Amount | 13 | Protected | LONG | 4 bytes |
| Weight | 14 | Protected | ULONG | 4 bytes |
| Totals in local mode | | | | |
| Number of operations | 15 | Protected | USHORT | 2 bytes |
| Number of units | 16 | Protected | ULONG | 4 bytes |
| Amount | 17 | Protected | LONG | 4 bytes |
| Weight | 18 | Protected | ULONG | 4 bytes |
| Card status | | | | |
| Card locked | 19 | Normal | UBIT | Has totals | 1 bit |
| Card total | 20 | Normal | UBIT | Completed. | 1 bit |

**Access codes :**

| Code | Id | Field Id | Fields |
|---|---|---|---|
| ID_VAC_LISTARTOTVNT | 2 | 11<br>12<br>13<br>14 | Number of operations<br>Number of units<br>Amount<br>Weight |
| ID_VAC_LISTARTOTLOC | 3 | 15<br>16<br>17<br>18 | Number of operations<br>Number of units<br>Amount<br>Weight |

| | | | |
|---|---|---|---|
| ID_VAC_TOT | 4 | 15<br>16<br>17<br>18<br>11<br>12<br>13<br>14 | Number of operations<br>Number of units<br>Amount<br>Weight<br>Number of operations<br>Number of units<br>Amount<br>Weight |
| ID_VAC_FINAL | 5 | 20 | Card completed |
| ID_VAC_LOCK | 6 | 19 | Card with totals |
| ID_VAC_BORRAR | 7 | 20<br>19<br>15<br>16<br>17<br>18<br>11<br>12<br>13<br>14 | Card completed<br>Card with totals<br>Number of operations<br>Number of units<br>Amount<br>Weight<br>Number of operations<br>Number of units<br>Amount<br>Weight |

# 10. RECEIPT REQUEST

If requested, the scale will send the greatest number of **complete** receipts that can contain a message.
If a receipt does not fit completely on a message, it will be sent on one, two or however many messages are necessary.

## 10.1 FORMAT REQUEST (IDF:000003)

### 10.1.1 Question format

With this message, you can request the format the scale will use to respond to the receipt request.
There are no question data, the reply is the ordered reception list of field identifiers with their lengths.

### 10.1.2 Reply format

Gives information about the structure of the message that responds to the receipt request. It consists of a sequence of identifiers (IDs) and data sizes (Ions). It has 3 parts that are separated by group identifiers according to the type of information.
None of the identifiers is sent later with the receipt data.

The message has the following format:
: <IDC1><ID1><lon1><ID2><lon2>….<IDn><lon><IDC2><IDn+1><lonn+1>…
'IDCi', 'Idi' and 'lon' have a length of 1 byte, with the following being:
<IDCi> the group identifier.
<IDi>: the datum identifier.
<lon> the length.

**Group identifiers:**

| FIELD | IDC | DESCRIPTION |
|---|---|---|
| Message heading | 50 | General information about the reply message: no. of receipts sent, initial receipt, etc.<br>1-accompanying ID is sent per message. |
| Receipt heading | 51 | The accompanying IDs form part of the receipt heading.<br>1 ID is sent for each receipt that contains the message. |
| Receipt line | 52 | The accompanying IDs form each receipt line.<br>1 ID is sent for each receipt line that contains the message. |

**Message format and LON content (scale version: 9.000):**

| FIELD | ID | LON | COMMENTS |
|---|---|---|---|
| Message heading | 50 | | |

| | | | |
|---|---|---|---|
| Number of receipts | 1 | 1 | Total of receipts that contain the message. |
| Initial line (1) | 2 | 1 | Number of the first line. |
| Line numbers (1) | 3 | 1 | Only valid if the message contains part of a receipt. |
| Receipt number | 4 | 4 | Number of the receipt sent, or number of the first receipt if various are sent. |
| Receipt heading | 51 | | |
| Receipt number | 150 | 4 | |
| Vendor | 151 | 4 | Vendor code. |
| Counter | 152 | 1 | |
| Date and time | 153 | 4 | In seconds from 1 January 1980. |
| Scale number | 154 | 1 | |
| Total amount due | 155 | 4 | Virtual decimal point. |
| Number of lines (1) | 156 | 1 | |
| Reserved 1(1) | 157 | 1 | |
| Cancelled lines (1) | 158 | 1 | Number of lines cancelled. |
| Receipt type | 159 | 1 | |
| Receipt state (1) | 160 | 1 | |
| Customer | 161 | 4 | |
| Receipt discount | 162 | 1 | In %. |
| Reserved 2(2) | 199 | 2 | |
| Internal tick. No. (2) | 164 | 2 | |
| Total gross amount due | 163 | 4 | Virtual decimal point. Gross total (without discount). |
| Receipt line | 52 | | |
| Line number(1) | 100 | 1 | |
| Item code | 101 | 4 | |
| Price | 102 | 4 | Sale price. Virtual decimal point. |
| Quantity | 103 | 4 | Weight in grams or units. |
| Reserved 3 | 149 | 1 | |
| Line type | 104 | 1 | |
| Line discount | 106 | 1 | In %. |
| Amount due | 105 | 4 | Virtual decimal point. |

1 Prior to scale version 9, the “LON” of these fields was 2.

2 Prior to scale version 9, these two fields were together:

| FIELD | ID | LON | COMMENTS |
|---|---|---|---|
| Reserved 2 | 199 | 6 | |

## 10.2 RECEIPT REQUEST (IDF:000002)

### 10.2.1 Question format

| FIELD | Bytes | VALUES/DESCRIPTION |
|---|---|---|
| What | | 1: Consult same as<br>2: More than<br>3: More than or same as<br>4: Less than<br>5: Less than or same as |
| Type of numbering | 1 | 0: External numbering according to system configuration<br>1: Internal numbering, without receipt type in order of carrying out<br>2: Indexed numbering without receipt type (network, local, roll, packaging, etc.) in order in the database |
| Receipt number | 4 | Number of the first receipt requested according to the value of the ‘what’ field |
| Receipt | 1 | Number of receipts requested. FF the ones that fit entirely on the reply. |

| | | |
|---|---|---|
| quantity | | A receipt that does not fit on a complete message will be sent alone. There is never an incomplete receipt at the end of a reply. |
| Initial line | 2 | Number of the initial line requested, the first is line 1, the field to use in the case of receipts that do not fit on a single reply |
| Order | 1 | 0 : Do not mark as an order<br>1 : Mark as an order<br>2 : Mark as confirmed |
| Receipt type | 1 | Type of receipt. Types 4 to 7 are only for orders by external numbering. Type 8 is only for indexed or internal numbering.<br>0 : Roll<br>1 : Super<br>2 : Packaging<br>3 : Self-service<br>4 : Local roll<br>5 : Local super<br>6 : Local packaging<br>7 : Local self-service<br>8 : Global without type |
| Item number of the consultation | 4 | Request by external numbering: scale number, counter, vendor, date, etc. according to the numbering type of the system<br>0 in all other cases |
| Consultation condition (1) | 1 | Look at or don't look at the condition of the state datum. Do not use the condition if the bit is at 0<br>bit 0 : Local<br>bit 1 : Loaded<br>bit 2 : Ordered<br>bit 3 : Ordered and confirmed<br>bit 4 : Closed<br>bit 5 : Reloaded<br>bit 6 : Cancelled |
| Condition datum (1) | 1 | State of the condition. The datum should comply with the condition if the bit is at 1. The bits are the same as in 'consultation condition' above |

**Question format (continued)**

| FIELD | Bytes | VALUES/DESCRIPTION |
|---|---|---|
| Extra consultation condition (2) | 1 | Another condition that should comply with the receipt consultation<br>0: No condition<br>1: Same as vendor<br>2: Same as counter<br>3: Same as date<br>4: Same as scale<br>5: Same as customer |
| Extra condition datum (2) | 4 | Datum (item) that the extra condition should comply with |

(1) and (2) : These fields are not necessary and the scale only uses them from version 9.001 on.
If the fields 'consultation condition' and 'condition datum' do not appear, it is considered a consultation without condition with 'receipt type' at 8, and 'type of numbering' as internal or indexed.
If the fields 'extra consultation condition' and 'extra condition datum' do not appear, it is considered a consultation without an extra condition.
For a request with external numbering according to the configuration of the system, it is necessary to put the type of receipt and the item number being consulted. Numbering is always carried out by the scale for local receipts.
For a request by internal numbering, only requests made by using 'same as' or 'more than' are valid.
For a request by index numbering, the first index is 0.

Receipts in the network should be requested from the serving scale.

The backup scale stores the receipts in local together with a copy of the receipts in the network (by order of arrival). To request the receipts in local from this scale, use the conditional request with the local bit activated.

### 10.2.2 Reply format.

Reply format with "y" receipts numbered from "x" on.
<Message heading ><X receipt heading><Line 1>…<Line n>…<X + y receipt heading ><Line 1>…<Line n>

**Reply message format (scale version 9.000):**

| FIELD | ID | LON | DESCRIPTION |
|---|---|---|---|
| Message heading | 50 | | |
| Number of receipts | 1 | 1 | Total number of reply receipts, if one it could be incomplete. |
| Initial line | 2 | 1 | Initial line number sent in the reply.<br>Lines are numbered from 1 on.<br>Only valid if the reply has a single receipt. |
| Number of lines | 3 | 1 | Number of lines of the reply.<br>Only valid if the reply has a single receipt. |
| Receipt number | 4 | 4 | Indexed numbering of the first receipt consulted. |
| Receipt heading | 51 | | |
| Receipt number | 150 | 4 | Receipt number loaded, according to system numbering. |
| Vendor | 151 | 4 | Code of the vendor who produced the receipt. |
| Counter | 152 | 1 | Number of the counter at which the receipt was produced. |
| Date and time | 153 | 4 | When the receipt was produced, by date and time.<br>In seconds from 1 January 1980. |
| Scale number | 154 | 1 | Number of the scale that produced the total. |
| Total amount due | 155 | 4 | With virtual decimal point. |
| Total no. of lines | 156 | 1 | Number of lines of the receipt. |
| Reserved 1 | 157 | 1 | |
| Cancelled lines | 158 | 1 | Number of lines cancelled. |
| Receipt type | 159 | 1 | 1: Roll receipt<br>2: Super receipt<br>3: Packaged receipt<br>4: Self-service receipt |
| State of the receipt | 160 | 1 | Each bit indicates the state of the receipt: 1 is correct, from left to right :<br>bit 0 : Local<br>bit 1 : Loaded<br>bit 2 : Ordered<br>bit 3 : Ordered and confirmed<br>bit 4 : Closed<br>bit 5 : Reloaded<br>bit 6 : Cancelled |
| Customer | 161 | 4 | Receipt customer (e.g., turn system). |
| Receipt discount | 162 | 1 | Percentage of discount applied to the receipt total.<br>bit 7: no discount if set at 0<br>bits 0 to 6: percentage discount |
| Reserved 2 | 199 | 2 | |
| Internal tick no. | 164 | 2 | Internal receipt numbering of the scale network. |
| Gross amount due | 163 | 4 | Amount due without applying discount (sum of the lines) |

**Reply message format (continued) (scale version: 9.000):**

| FIELD | ID | LON | DESCRIPTION |
|---|---|---|---|
| Receipt line | 52 | | |
| Line number | 100 | 1 | Number of the line. |
| Item code | 101 | 4 | Code of the item.<br>If 0, it is a sale line with a price entry. |
| Price | 102 | 4 | Sale price. With virtual decimal point. |

| | | | |
|---|---|---|---|
| Quantity | 103 | 4 | Weight can be in grams or units according to the field [line type] |
| Reserved 3 | 149 | 1 | |
| Line type | 104 | 1 | 4 high bits:<br>0000 – sale at weight<br>0001 – sale units<br>0010 – sale negative weight<br>0011 – sale negative units<br>4 low bits:<br>0000 – line not cancelled<br>0001 – line cancelled |
| Discount line | 106 | 1 | Percentage of discount applied to the receipt line<br>bit 7: no discount if set at 0<br>bits 0 to 6: percentage discount |
| Amount due | 105 | 4 | |

If the scale has no more receipts to send, the reply message returns IDF:0000100 (0x000064).

# 11. STATE OF THE SYSTEM (IDF:000004)

## 11.1 STATE OF THE SYSTEM REQUEST MESSAGE

| Field | Length | Description |
|---|---|---|
| FUN | 1 byte | Function. |

## 11.2 REPLY TO STATE OF THE SYSTEM REQUEST

| Field | Length | Description | |
|---|---|---|---|
| Equipment 0 | byte 1 | Bit | Description |
| | | 0 | State in the network |
| | | 1 | Master / Not master (1/0) |
| | | 2 | Backup / Not backup (1/0) |
| | | 3 – 5 | Form of sale |
| | | 6 | Totals |
| | | 7 | Receipts |
| | byte 2 | 0 | Updating items |
| | | 1 | Updating vendors |
| | | 2-7 | Reserved |
| | byte 3 | 0-7 | Counter |
| Equipment 2 to | | ... | |
| Equipment 64 | | ... | |

Information is always sent for 64 pieces of equipment (even if they do not physically exist).
Bits within a byte are numbered from 0 to 7, with 0 meaning the least significant (LSB) and 7 the most significant (MSB).
Totals and receipts:

0: No data. 1: Data.

The backup scale does not differentiate between local and network data. Even if there are no data in local, it will indicate that there are totals and receipts in the network.
Form of sale:

1: Sale (001 in bits).
2: Super (010).
3: Packaging (011).
4: Self-service (100).
5: Mixed (101).

# 12. LANGUAGE CONTROL MESSAGES (IDF:000005)

RESERVED FOR INHOUSE APPLICATIONS

# 13. APPENDICES

## 13.1 APPENDIX 1: BAR-CODE STRING

The following table shows the correspondence between the characters stored by the database in the fields corresponding to bar codes, and the letter introduced by using the keypad, as well as a brief description.

**Significance of bar-code string bytes:**

| DATUM | Significance | Character |
|---|---|---|
| 0x00 | Nothing | ‘ ‘ |
| 0x01 | Overflow | ‘S’ |
| 0x02 | Weight | ‘W’ |
| 0x03 | Price | ‘Z’ |
| 0x04 | Amount due | ‘I’ |
| 0x05 | Receipt | ‘T’ |
| 0x06 | PLU | ‘P’ |
| 0x07 | Code | ‘C’ |
| 0x08 | Vendor | ‘V’ |
| 0x09 | Line no. | ‘A’ |
| 0x0A | Batch no. | ‘L’ |
| 0x0B | Order no. | ‘0’ |
| 0x0C | Manufacturer | ‘F’ |
| 0x0D | Country | ‘M’ |
| 0x0F | 2-decimal weight | ‘D’ |
| 0x10 | Scale | ‘N’ |
| 0x11 | Counter | ‘Y’ |
| 0x12 | National amount due | ‘J’ |
| 0x13 | Intermediate check | ‘K’ |
| 0x14 | Digit 0 | ‘0’ |
| 0x15 | Digit 1 | ‘1’ |
| 0x16 | Digit 2 | ‘2’ |
| 0x17 | Digit 3 | ‘3’ |
| 0x18 | Digit 4 | ‘4’ |
| 0x19 | Digit 5 | ‘5’ |
| 0x1A | Digit 6 | ‘6’ |
| 0x1B | Digit 7 | ‘7’ |
| 0x1C | Digit 8 | ‘8’ |
| 0x1D | Digit 9 | ‘9’ |

## 13.2 APPENDIX 2: DATE-CONVERSION ROUTINES

*Auxiliary routines:*

```
#define EsBisiesto(a)     (!(a&3))
int Dias[12]={31,28,31,30,31,30,31,31,30,31,30,31};

long DiasAnyo (int i)         //Days of the year.
{
      return 365+EsBisiesto(i);
}

long DiasMes( int anyo, int mes )     // Days of the month
{
      long res = Dias[mes];
      if (mes==1) res += EsBisiesto(anyo);
      return res;
}
```

*Conversion from long to date in year, month, day:*

```
void Long2Fecha (long tptr, short *Dias, short *Mes, short *Anyo)
{
	long  Mdias, ADias;

	Dias = tptr / 86400L + 1;
	Anyo = 1980;
	while( Dias > ( ADias = DiasAnyo( Anyo) ) )
	{
		Dias -= ADias;
		Anyo++;
	}
	Mes = 0;
	while( Dias > ( MDias = DiasMes( Anyo, Mes ) ) )
	{
		Dias -= MDias;
		Mes++;
	}
	Mes++;
}
```

*Conversion from date in year, month, day to long:*

```
long Fecha2Long(short Dia, short Mes, short Anyo)
{
	long Fecha, anyo, mes, dia;
	long i;

	anyo = Anyo%100;
	anyo -= (1980%100);
	if (anyo <0) anyo=Anyo+100-(1980%100);
	Fecha = 0;
	for( i = 0; i < anyo; i++ )
		Fecha += DiasAnyo( 1980 + i );
	mes = Mes-1;
	for( i = 0; i < mes; i++ )
		Fecha += DiasMes( 1980 + anyo, i );
	Fecha += Dia - 1;
	return( Data * 86400L );
}
```

GRUPO EPELSA S.L.

ALBASANZ 6 y 8 28037 MADRID
TEL: (91) 754-30-14; FAX: (91) 754-48-26 e-mail:info@epelsa.com
CTRA. SANTA CRUZ DE CALAFELL , 35
08830-SANT BOI DE LLOBREGAT (BARCELONA-SPAIN)
TEL. (93) 654-62-12; FAX: (93) 654-54-53 e-mail:info@epel-ind.com
http / / www.epel-ind.com